INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Transformers for Time Series Forecasting

Vasco Manuel dos Santos Pedrosa Gonçalves de Jesus

MSc in Data Science

Supervisor:
Ph.D., Diana Aldea Mendes, Associate Professor,
ISCTE – University Institute of Lisbon

September, 2024

BUSINESS SCHOOL

TECNOLOGIAS E ARQUITETURA

Department of Quantitative Methods for Management and Economics

Department of Information Science and Technology

Transformers for Time Series Forecasting

Vasco Manuel dos Santos Pedrosa Gonçalves de Jesus

MSc in Data Science

Supervisor:
Ph.D., Diana Aldea Mendes, Associate Professor,
ISCTE – University Institute of Lisbon

September, 2024

Acknowledgements

I would like to express my gratitude to everyone who has supported me throughout the journey of completing this thesis.

First and foremost, I would like to thank my advisor, Professor Diana Aldea Mendes, for her invaluable guidance. I am profoundly grateful for the opportunity to work under her supervision and her expertise.

I am also grateful to my colleagues at Banco de Portugal, whose camaraderie and constant support made this experience both productive and enjoyable.

Lastly, I extend my heartfelt appreciation to my family for their love and encouragement. To my parents, Sílvia and Manuel, thank you for always believing in me and supporting my ambitions. You have always pushed me to aim higher and have provided me with the foundations to pursue my dreams. I am profoundly grateful to have you as my role models and I owe my successes to your sacrifices and unwavering belief in me. To my beloved grandparents, Mimi and Júlio, Josefina and José, your stories, unconditional love and hard work throughout your lives have been a constant source of inspiration. You have shown me the importance of resilience and the value of humility. Each of you has imparted lessons that have guided me, and I carry your lessons with me always. To my brothers, Bernardo and Francisco, your companionship, humor and unwavering support have been a source of strength and joy. You have always been my greatest allies, helping me in some way or another, thought the challenges I face. Thank you for being my confidants, and my best friends.

I dedicate this achievement to all of you as this thesis is as much a testament to your influence as it is to my efforts. This achievement is also yours.

Resumo

Este estudo tem como objetivo diminuir a lacuna existente entre a pesquisa (teórica) e a prática existente para a tarefa de previsão de séries temporais, ao introduzir e avaliar um novo modelo *transformer-based*. Este é baseado nas arquiteturas já existentes em modelos como o *Frequency Enhanced Decomposed Transformer* (Zhou et al., 2022) e o *Patch Time Series Transformer* (Nie et al., 2022), que se destacaram pela positiva em ambientes univariados e multivariados, respetivamente. O cerne desta tese é então o desenvolvimento de um modelo *transformer-based*, que combina elementos dos dois modelos acima mencionados. Através de diversos testes rigorosos recorrendo às métricas erro quadrático médio (*MSE*) e erro médio absoluto (*MAE*), o desempenho deste novo modelo foi comparado com os que o originaram. Em sede de conclusão as descobertas revelam que este novo modelo supera um dos modelos que o origina ainda que não supere o outro. Esta pesquisa contribui para a área da Ciência de Dados ao fornecer *insights* sobre a eficácia deste tipo de modelos e orientando possíveis avanços futuros para a tarefa de previsão de séries temporais.

Palavras-Chave: Previsão de Séries Temporais, *Transformers*, *Attention Mechanism*, Erro Quadratico Médio (*MSE*), Erro Médio Absoluto (*MAE*)

Abstract

This study aims to bridge the gap between theoretical research and practical application in time series forecasting by introducing and evaluating a novel transformer-based model. It builds on the foundations set by models such as Frequency Enhanced Decomposed Transformer (Zhou et al., 2022) and Patch Time Series Transformer (Nie et al., 2022), which have excelled in univariate and multivariate settings. The core of this thesis is the development of a transformer-based model, combining elements of the two models mentioned above. Through rigorous testing using Mean Squared Error (MSE) and Mean Absolute Error (MAE) as evaluative metrics, the new model's performance was benchmarked against its precursors. The findings reveal that while the latest model surpasses one of its predecessors in forecasting accuracy, it does not outperform the other. This research contributes to the field of Data Science by providing insights into the effectiveness of these models and guiding future advancements in time series forecasting.

Keywords: Time Series Forecasting, Transformers, Attention Mechanism, MSE, MAE

Index

Index of figures

Index of tables

Introduction

Throughout the years, the models used in time series forecasts have changed. From the traditional Auto-Regressive Integrated Moving Average (ARIMA) statistical linear models to the Recurrent Neural Networks (RNN) or the Long Short-Term Memory (LSTM) model as neural network implementations, we have seen an increase in the forecast performance by performing these algorithms. It was noticed, then, that machine (deep) learning models, such as RNN or LSTM, were the ones that performed better (Lezmi & Xu, 2023; Sreelekshmy Selvin et al., 2017).

That said, it is rational to think that more machine learning models had to come out to try and outperform old ones. And so it was, with the appearance of the Transformer (Vaswani et al., 2017), initially designed to perform natural language processing tasks, that time series forecasting took a turn. The excellent performances in other fields triggered a great interest in the time series community (Wen et al., 2022).

It was only a matter of time until researchers tried to use algorithms from the Transformers family in time series forecasting - and they did with great success (H. Wu et al., 2021; H. Zhou et al., 2020). The main idea relies on the understanding that just like a phrase/corpus in a natural language processing problem is a sequence of words that serve as input to the Transformer architecture to obtain an output; it is also fair to consider the values in a time series a sequence of numbers as valid input to get a certain output – the forecast. Since there is no perfect model, it is also relevant to mention that research has been done on the problems inherent to using transformers for time series forecasting. The detection of problems occurred quite early, and the main ones are, as mentioned by the authors (Li et al., 2019): (1) Memory bottleneck associated with space and time complexity. (2) The fact that this model is insensitive to local context comes as a problem when solving issues in time series forecasting.

This whole study, and more precisely the literature review section, will focus on exploring and exposing the more relevant methodologies to time series forecasting using the Transformer architecture or, as mentioned before, transformer-based solutions. We will investigate key components of the research topic based on studies that followed some of the methodologies in question. Following that, emphasis will be placed on comparing and discussing all relevant existing transformer-based solutions for this problem, also with simple linear models, so that it will be possible to better understand and progress with further studies. Finally, a novel implementation of a hybrid transformer-based model will be tested and paired against the original models to obtain better or confirm already discussed insights. To do so, we utilize well-known benchmark datasets. These include the Traffic dataset, Weather dataset, Illness dataset, and Exchange Rate dataset, each representing different domains such as traffic, meteorology, healthcare, and finance, respectively. These datasets were chosen to ensure the robustness of the proposed hybrid model across diverse application areas. The characteristics of these datasets, such as their time spans, frequencies, and variability, provide a comprehensive basis for evaluating the performance of transformer-based models in capturing both short-term and long-term dependencies That said, three main research questions arise:

Question 1: Are transformer-based models relevant in time series forecasting?

Question 2: Does transformer-based models' performance change for different windows/settings?

Question 3: How does the integration of PatchTST into FEDFormer influence the computational efficiency and predictive accuracy of transformer-based models for time series forecasting?

With that, and briefly, this study will be divided into four main chapters: the first one will regard the selection and revision of relevant literature for the study; chapter two will describe the corresponding methodology for the models, data used and respective evaluation metrics used; chapter three will mainly be composed of the results and discussion of the said results; and finally the fourth and final chapter will describe the obtained conclusions and possible future work that will be drawn from the previously obtained results.

CHAPTER 1

Literature Review

1.1 Literature selection

To better understand the subject of study, the systematic literature review (SLR) method was used, which is amply considered worldwide and aims to help identify and gather relevant literature better. This literature review resorted to one of the most significant research databases, Scopus.

In Scopus, one can gather relevant literature in many fields using specific queries to obtain more specific results. The systematic literature review started by using a more generalist query; that is, it was started by searching papers regarding the task of time series forecasting with the following query: "TITLE-ABS-KEY ("time series" AND "forecasting")" which means that it searched the Scopus database for papers that contain in the title, abstract or keywords, the terms "time series" and "forecasting". That specific query matched 49,025 documents found in the database, which is a lot due to the query being too simple and general. Yet some papers were extracted from this search because it was intended to understand better the task of time series forecasting in its essence. However, it was necessary to narrow the scope of the search by introducing a more complex query.

That said, the second and final query that was used was the following: TITLE-ABS ("forecasting" AND "time series" AND "transformers") AND PUBYEAR > 2016 AND PUBYEAR < 2024, which means that it searched the Scopus database for papers that contain in the title or abstract the terms "forecasting", "time series" and "transformers" where the year the paper was published is comprehended between 2017 and 2023. In this final query, the intention was to gather the most important papers for my study, that is, papers that matched in several aspects with my work – that's why this query already included the term "transformers" and set the interval of the publishing year starting in 2017, the year the original transformer paper (Vaswani et al., 2017) was released.

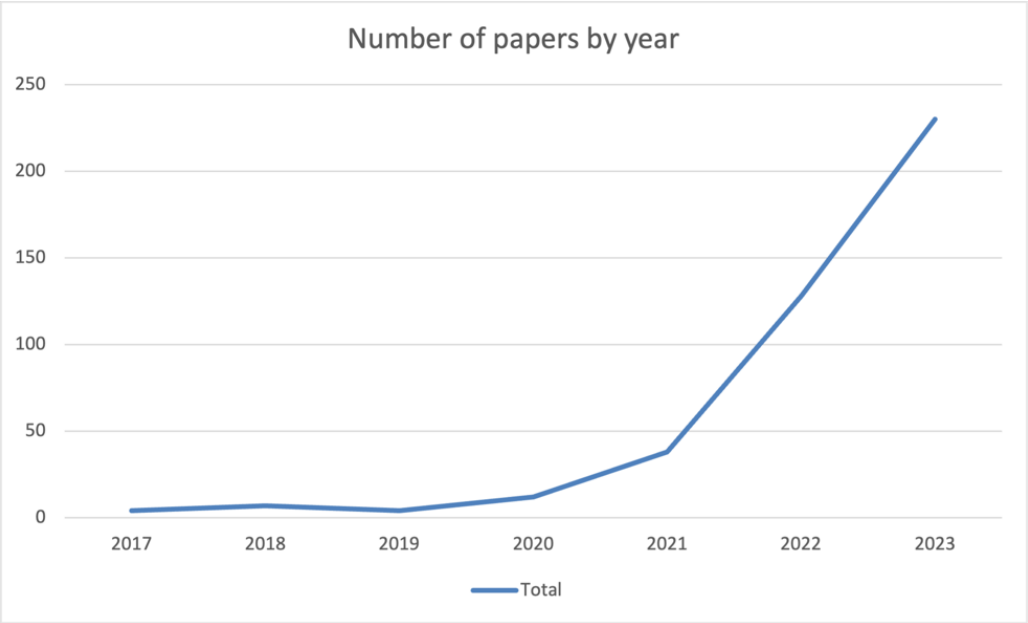

Figure 1 – Number of papers by year.

This final query resulted in 423 documents, which is still a large number, yet it started the research using the most relevant and cited ones. As can be observed in Figure 1, the exponential growth in interest in using transformers for the task of time series forecasting is evident as the number of papers dramatically increased.

As mentioned earlier, there was still a need to narrow the number of articles considered relevant to this study. It started by selecting the ones that proposed to create transformer-based solutions, such as the Informer (Zhou et al., 2020) or the Autoformer (Wu et al., 2021), as my study will, in some ways, do precisely that. It is also relevant to mention that Scopus was not the only source for gathering the research papers that were found helpful. Some of the collected documents were found on platforms like Google Scholar, since, after reading the Informer and Autoformer papers - that mentioned other transformer-based solutions, surveys, and general knowledge on the matter - and didn't appear on the Scopus database - as was the case of the Scaleformer (Amin Shabani et al., 2022), or the Crossformer (Zhang & Yan, 2023), among others.

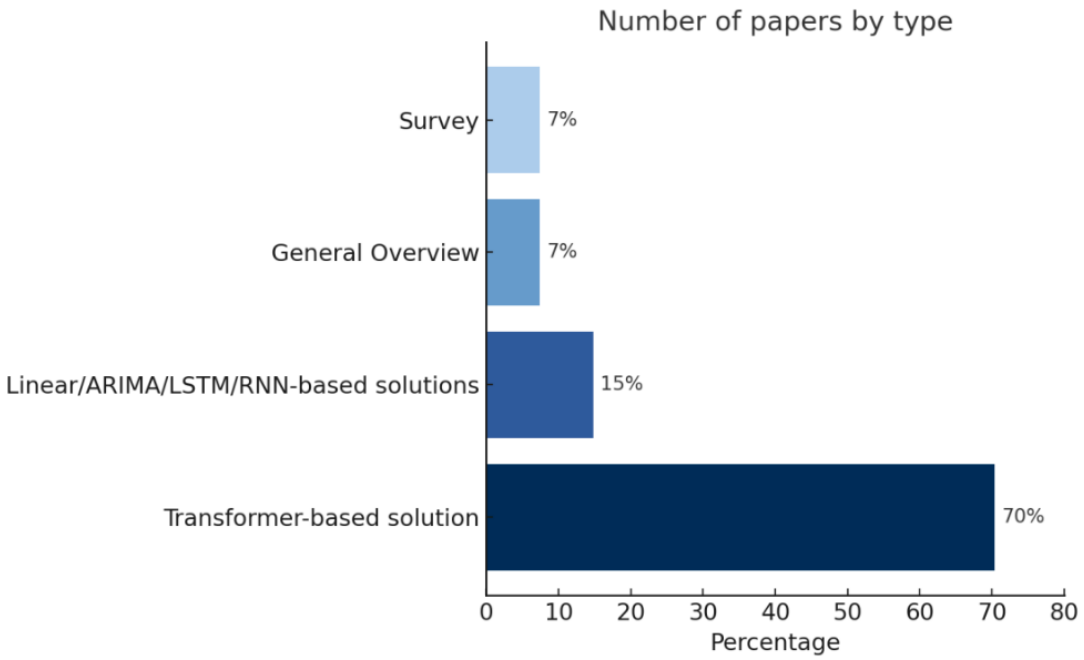

Figure 2 – Number of papers by type.

That said, 27 papers were gathered and divided into four relevant classes. It is possible to observe, based on Figure 2, that most of them regard transformer-based solutions that were already mentioned and that were found highly relevant, and others, due to the way the results were displayed, were not so relevant, such as the State Space Decomposition Neural Network (Lin et al., 2021), the Temporal Fusion Transformer (Lim et al., 2021), or the Adversial Sparse Transformer (Wu et al., 2020) that presented the results using, as the main evaluation metric, for example, the accuracy of the model and not *MSE* or *MAE* as most of the papers use. Still, these papers were pertinent in the research as, even though the results were not comparable with other methods, they still provided knowledge on their methodologies, yet, they will not be present in the results section of this study. Other publications, more precisely, four of them, regarded time series forecasting solutions using Linear, ARIMA, LSTM, or RNN models, which is the case (Chimmula & Zhang, 2020; Siami-Namini et al., 2018; Sreelekshmy Selvin et al., 2017; Zeng et al., 2023) and provided the fundamental historical context on the task using some "older" and classical models. Finally, two of them regard surveys done on the task of time series forecasting using transformers/deep learning models (Lim & Zohren, 2020; Wen et al., 2022), which gave an overview of the task itself, and the other two papers regarding a general overview of the topic of transformers and/or time series forecasting (Lara-Benítez et al., 2021; Lezmi & Xu, 2023) that gave the insights needed to combine the two subjects in question, Transformers, and Time Series Forecasting, in a more general way.

1.2 Literature Results

In this section, it is proposed to present and discuss the models and respective results found in the selected literature - to do so, the results gathered from the papers shown in Appendixes A.1 – A.5 were used.

To present these results, the same methodology was used as in most papers: separate the results into univariate and multivariate sections and go from there. It is also relevant to mention that the results in the Appendixes show only the best results for each model and each dataset for each prediction length – that is, if a model is composed of two or three different variants of the model, each with its particularity, the results shown in the tables only regard the best results among all the variants of the same model.

In what follows, we present the data used in most selected papers for the literature review, alongside with the respective models. Also, results will be extracted directly from the papers (regarding univariate and multivariate settings) and summarized in order to gather valuable insights and formulate conclusions to enrich future studies.

1.2.1 Data

It is important to mention that the models in the selected papers used the same datasets as, in this particular field of time series forecasting, they are well-known for their benchmark nature. That said, the considered datasets will be described in the following parts of this sub section.

Starting with the Electricity Transformer Temperature (ETT datasets - ETTh1, ETTh2, ETTm1, and ETTm2)[1] that consist of 2 years of electric power deployment data from two separate counties in China that are split into 4 different subsets where the only difference between them is the granularity - that is, ETTh1 and ETTh2 represent hourly data, and ETTm1 and ETTm2 represent 15-minute data. The Weather dataset is also largely used in this task, and it contains weather data with a 10-minute frequency for an entire year. The ECL dataset represents data on electricity consumption recorded every 15 minutes from 2011 to 2014. The electricity dataset is like the previous one, but only regarding hourly data. The exchange dataset, as it explicitly says, regards (daily) exchange rate data of eight different countries ranging from 1990 to 2016. The ILI dataset is the Illness dataset regarding weekly data on patients with influenza-like illness in the United States between 2002 and 2020, and finally, the Traffic

[1] https://github.com/zhouhaoyi/ETDataset

dataset considers hourly traffic data. All datasets can also be found on the Autoformer GitHub page[2].

Dataset	Time Period	Frequency	Number of Observations
ETTh1	2 years	Hourly	17420
ETTh2	2 years	Hourly	17420
ETTm1	2 years	15 minutes	69680
ETTm2	2 years	15 minutes	69680
Weather	1 year	10 minutes	52696
ECL	2011-2014	15 minutes	26304
Electricity	2011-2014	Hourly	26304
Exchange	1990-2010	Daily	7588
ILI	2002-2020	Weekly	966
Traffic	2016-2018	Hourly	17544

Table 1 – Description of the datasets.

As summarized in Table 1, it is represented, for each dataset, the corresponding time period, its frequency and the number of observations.

The models and the respective results presented in the relevant selected literature will be reviewed. The transformer-based models will be described, some advantages presented, and then the results will be compared.

1.2.2 Models

Firstly, and to give some context, a brief description will be made regarding the original transformer model proposed in 2017 (Vaswani et al., 2017) so that it is possible to better understand some of the particularities of other transformer-based solutions.

[2] https://github.com/thuml/Autoformer

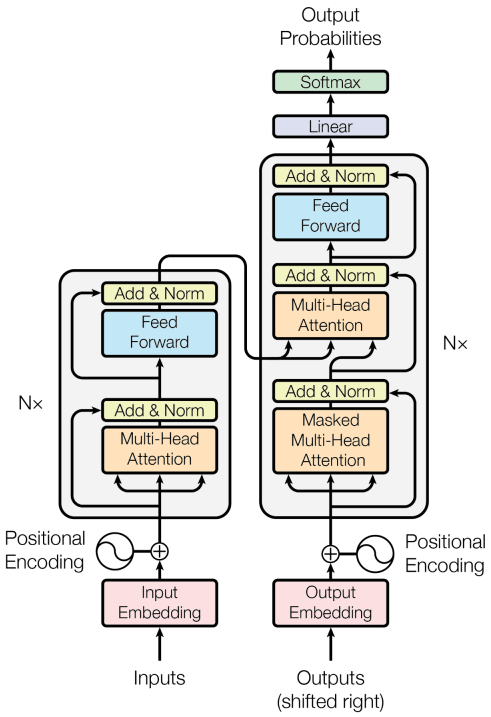

Figure 3 – Original Transformer model architecture.

In a simple way, the Transformer model consists of an encoder-decoder architecture where both are composed of a stack of identical layers – as can be observed in Figure 3. In the original architecture, the encoder is composed of six identical layers, and each layer has two main layers that consist of a multi-head self-attention layer, and a fully connected feed-forward network. The decoder, in the original architecture, is also composed of six identical layers. In addition to the two sub-layers in each encoder layer, the decoder also inserts a third sub-layer, which is a multi-head attention layer.

That said, the Transformer can be categorized as a type of neural network model designed to process sequential data, such as text or, in this case, time series. One of the main particularities of this model lies in the fact that it uses an attention mechanism (self-attention) to weigh the importance of different elements in the data, unlike earlier models. Some key concepts to retain from this architecture are as follows: (1) The self-attention mechanism is an attention mechanism that allows the model to focus on different parts of the input and weight it accordingly. (2) Positional encoding that straightforwardly helps the model to understand the order of the elements in the sequence.

Starting with the Informer (Zhou et al., 2020), the authors mainly identified the inefficiency of traditional self-attention mechanisms in handling long sequences and so, they presented two main characteristics for the model: a *ProbSparce* self-attention mechanism with its benefits and a generative style decoder. Building on this, the Autoformer (Wu et al., 2021), observed that while the *ProbSparce* self-attention mechanism improved efficiency, it still struggled to capture dependencies over time. To solve this, the authors proposed replacing self-attention with the Auto-Correlation mechanism. While the Autoformer enhanced dependency modeling, the FEDformer (Zhou et al., 2022) noted that previous models still had difficulty capturing both seasonal and trend components. That said, the authors proposed to combine the vanilla transformer with frequency analysis using a seasonal-trend decomposition method. The Pyraformer (S. Liu et al., 2022) introduces a pyramidal graph-structure attention mechanism to the transformer in order to address the challenge of capturing multi-scale dependencies. PatchTST (Nie et al., 2022), identified the need, in a general way, for better segmentation of time series data. The authors then proposed two main changes to the original transformer: the first one is the segmentation of the time series into subseries-level patches; the second one is the implementation of channel-independence (already used in other traditional models, just not on transformers). The Non-stationary Transformers (Y. Liu et al., 2022) focused on the problem of over-stationarization. To solve this the authors proposed to use a series stationarization and a de-stationary attention mechanism, that allowed the model to adapt to changes in the underlying data. Subsequently, to enhance the ability of transformer-based models to understand cross-dimensional dependencies, the Crossformer (Zhang & Yan, 2023) proposed a solution utilizing cross-dimension dependency, which improved the model's capability to capture interactions between different dimensions of time series data. Finally, the Channel Aligned Robust Blend Transformer (Xue et al., 2023) also introduces two changes: a dual transformer structure; and a robust loss function to alleviate the potential overfitting issue.

Apart from all the models presented above, there are also some models or frameworks that were not included in the results tables mainly because of a few reasons: being too specific and not relevant for "overall" time series forecasting, like the transformer proposed for energy forecast (Oliveira & Oliveira, 2023), the transformer suggested for traffic forecast (Cai et al., 2020), or even the transformer for the Influenza Prevalence Case (Wu et al., 2020); the second reason relies on the choice of the authors regarding the selection of the metrics to evaluate the model, as said earlier – that is, most of the authors chose to use *MSE* and *MAE* to evaluate their models, other models just like the SSDNet or the Hierarchical Multi-Scale Gaussian Transformer (Ding et al., 2020) chose not to do so, that said, it was decided to omit the ones that didn't present these metrics or similar in the tables for consistency reasons. Finally, the so-called frameworks like the Scaleformer (Amin Shabani et al., 2022) or the Tightly-Coupled Convolutional Transformer (Shen & Wang, 2022) that, despite being relevant to the research, don't qualify as a "real" transformer-based solution just because they mainly rely on other transformer-based solutions and add some changes on top of them. Yet, it is still especially important to mention that even though they are not present in the tables, it doesn't mean the changes proposed by these papers were not valuable. These changes are significant and meant to be taken just like the others as relevant to future work.

1.2.3 Results

After introducing most of the models, we will present the results. The univariate results will now be discussed, followed by the multivariate ones, for the transformer-based solutions as they are done in most literature. After doing so, a comparison and discussion will occur regarding the best results from transformer-based solutions *versus* linear solutions presented in the literature – Appendix A.2.

Univariate Time-series Forecasting: Under this setting, it is from Appendix A.1 that we can observe that: (1) The Informer and FEDformer models were the only models to perform tests with the ETTh1, ETTh2, ETTm1, ECL, ILI, and Electricity datasets. (2) For the ETTm2 dataset, the FEDformer, among the other models, performs better, although not prominently. It is also nice to note that, through the varying input lengths, we see an increase in the *MSE* and *MAE.* (3) Regarding the Weather dataset, the Informer and FEDformer are the only models to use this dataset to make tests. That said, and with varying input lengths, the FEDformer performs better, this time, prominently. (4) Finally, in the Exchange dataset, the implementation of the Non-stationary transformer outperforms all the other models. In conclusion, we can affirm that the best-performing transformer-based model under this setting is the FEDformer.

The results will vary a lot based on many factors, so it is necessary to point out again that the results gathered are based only on the respective papers of each model.

Multivariate Time-series Forecasting: Under this setting, it is from Appendix A.2, only regarding the ETT dataset, that it is possible to observe that: (1) There were three main models for getting the best results. The Autoformer, Crossformer, and PatchTST were the ones that got the best results. (2) In the ETTh1, it was the Crossformer and PatchTST that got the best results; In the ETTh2 and ETTm2 it was the Autoformer and PatchTST (Crossformer didn't use these datasets); And finally, in the ETTm1 all three models performed nicely. (3) It is necessary to mention that, although all three models outperformed all others, among these 3, it is possible to say that the PatchTST is, in this dataset, the best model to use. Not only do they get a significant portion of the best results, but they also get the best results where the input length is more significant, which is also a plus. Now, passing to Appendix A.4, where the ILI, ECL, and Weather datasets are regarded, it is possible to observe that: (1) For the Weather dataset, it is possible to say that PatchTST is, in a general way, the model that performs the best. Alongside the results from Crossformer, which only performed well in input lengths, other models didn't make any tests. (2) Regarding the ECL dataset, the only model that performed any tests under the multivariate setting was the Crossformer. That said, it is not possible to say that, for this dataset in particular, the considered model is better than other models listed. (3) Finally, it is on the ILI dataset that the PatchTST confirms, once again, its dominance regarding performance. It is possible to say that PatchTST is the best model for this set of datasets. Finally, in Appendix A.5 regarding the Traffic, Exchange, and Electricity datasets, it is possible to say that: (1) For the Traffic dataset, it is, once again, the PatchTST model that outperforms all the others (except for the cases regarding the 24 and 48 input lengths that only the Crossformer model used.) (2) For the Exchange dataset, implementing the Non-stationary transformer, just like in the univariate setting, outperformed all the others. It is also relevant to mention that, for larger input lengths, the results obtained by the Non-stationary transformer decreased a bit, and the FEDformer got a little better results. (3) Finally, in the Electricity dataset, the model that got the best results was the PatchTST, except regarding the 168 input length, that only the Pyraformer model used. In a general way, it is possible to say that, once again, PatchTST was the model that got the best results.

Transformer-based *versus* Linear solutions: After gathering the best transformer-based solutions, it is interesting to compare them with linear solutions using the same benchmark datasets from the literature. It's the case of the long-term time series forecasting – linear (LTSF-Linear) (Zeng et al., 2023), a simple linear model, as stated by the authors, consisting only of a weighted sum of historic L values to predict future T timesteps. This model consists of three separate models - Linear, NLinear and DLinear - each designed for its particular field (depending on the dataset) and varying only in pre-processing techniques. The best results obtained with this model are present in Appendix A.2. That said a compilation table was made so that it is possible to compare the results from this linear model and the best multivariate transformer-based solution, as it is present in Appendix A.6. It is then possible to observe that: (1) Among the two selected models it is challenging to decide which performs better - although practically all the best results belong to the PatchTST model, it is by a small margin most of the time.

1.3 Conclusions

In conclusion, the results detailed in Appendix A.1 reveal that the FEDformer consistently outperforms other models across all the datasets regarding the univariate setting. Notably, it excels in the ETTm2 dataset and demonstrates superior performance in the Weather dataset, establishing itself as the top-performing model. Changing the setting to the multivariate time series and referencing Appendixes A.3-A.5, three main models emerge as top performers – Autoformer, Crossformer, and PatchTST. Yet, overall, PatchTST stands out as the dominant model in multivariate forecasting, displaying its versatility and consistently superior performance across diverse datasets.

It is once again crucial to note that these conclusions are drawn from the respective papers and regarding each model, and the results may vary depending on specific factors like computational resources. Nonetheless, the comprehensive analysis presented here underscores the robust performance of the FEDformer and PatchTST in univariate and multivariate time series forecasting for transformer-based solutions, respectively, and LTSF-Linear as a robust linear solution for time series in general.

This systematic literature review's significance lies in identifying time series forecasting models that exhibit superior performance in different contexts – that is, not only regarding the business area of each dataset, but also the setting (univariate or multivariate). FEDformer excelled in the univariate setting, while PatchTST was superior in the multivariate setting. These findings are essential to this work since they provide valuable insights into which approaches may be most effective to attack the problem. It is also necessary to acknowledge certain limitations, one of them being that the conclusions are based on information available in the selected papers, introducing a potential bias and limitation. Additionally, one of the most important points is that time series forecasting is not easy *per se*.

For future research, and based on the limitations of the task in hand, it would be beneficial to investigate, explore, and implement a new state-of-the-art transformer-based solution that could combine some of the most interesting characteristics of the two best models presented in this literature review and also try to solve at least in part some of the problems associated with the use of this model such as (1) Memory bottleneck associated with time and space complexity, and (2) The fact that this model is insensitive to local context. Moreover, a deeper understanding of factors influencing model performance in specific scenarios could provide valuable insights for future improvements.

This systematic literature review offers a comprehensive overview of transformer-based solutions for time series forecasting, highlighting their practical nuances and results, providing clear guidance, and serving as a strong base for future work.

CHAPTER 2

Data and Models

In this chapter, we will explore the methods chosen and employed for the task of time series forecasting. Following, in some ways, the CRISP-DM (Wirth & Hipp, 2000) methodology, we will delve into the development of a novel hybrid transformer-based model, which incorporates the two best-performing models discussed in the previous chapter – although the novel model's architecture is presented later, the methodology used to develop it can be observed in the flowchart in Figure 4.

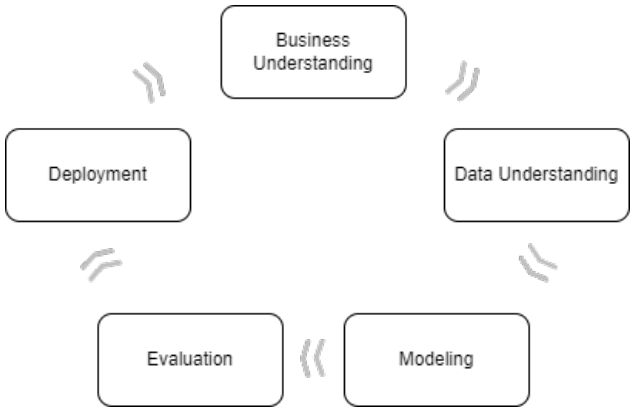

Figure 4 – CRISP-DM flowchart.

The focus will lie on explaining not only the data used but also the technical aspects of the integration of the two models into a new one – mainly the integration of the PatchTST layer into the input layer of the FEDFormerMS (the FEDFormer implementation of the Scaleformer framework).

2.1 Data used

So, regarding this particular section, the data used in the novel hybrid model will be presented. It will consist of the data used in most of the other models presented in the literature review so that all future comparisons of results and future discussions are reliable and comparable.

Four datasets will be used, spanning different areas, which will also be an excellent test for the novel model so that it will be possible to observe its behavior when presented with multidisciplinary data. These datasets are the Traffic dataset, the Weather dataset, the Illness dataset and the Exchange Rate dataset (all four datasets are available on the Autoformer[3] github page). All these four datasets are now to be thoroughly described and studied to provide more context for future methods and results. It is important to note that, as these datasets are widely used for time series forecasting, it is normal that they have already been worked on and need no further improvements to fit the novel model.

2.1.1 Traffic

The Traffic dataset contains traffic volume data, ranging from July 2016 to July 2018, with 17,544 rows, 863 columns and no missing data.

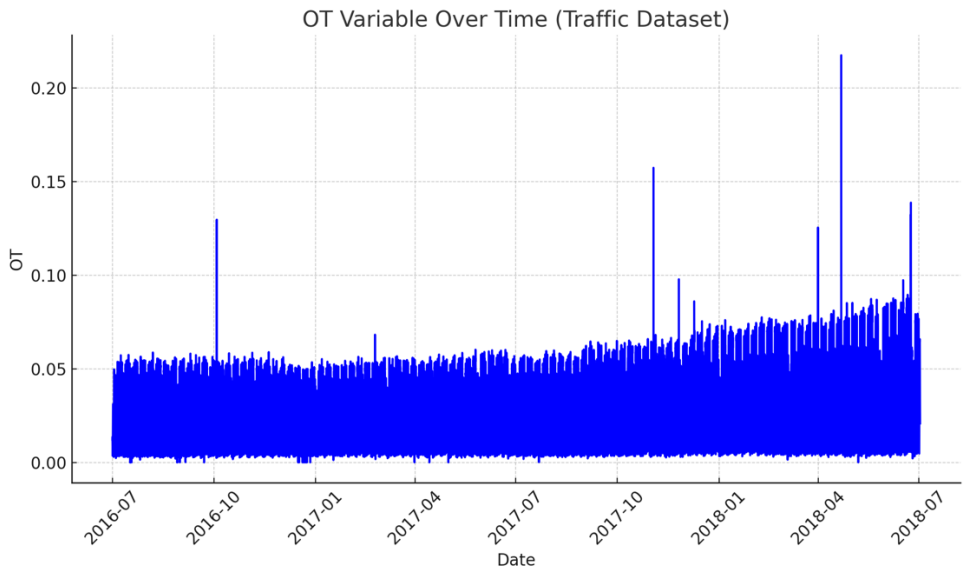

Figure 5 - Traffic dataset plot – target variable.

[3] https://github.com/thuml/Autoformer

As it is possible to see in Figure 5, the target variable (OT column) exhibits a consistent pattern throughout, registering fluctuations in traffic throughout the days (seen by all the spikes, maximums, and minimums). It is noticeable a slight upward trend since around 2017, yet not too extreme, indicating a gradual increase in traffic volume over time. It is evident, once again, that the daily patterns reveal significant variations, likely corresponding to the peak traffic hours during the day – the spikes, maximums, and minimums might indicate rush hours and off-peak hours, respectively.

	Mean	Mode	Median	Standard Deviation
Target Variable (OT)	0.032	0.005	0.034	0.019

Table 2 – Traffic dataset – target variable statistics.

Regarding the statistical measures of this variable, the mean and median values are quite close, as is observable in Table 2, which points to a symmetric distribution of the data. The value for the standard deviation also indicates the presence of regular fluctuations in the data.

2.1.2 Weather

The Weather dataset contains 7588 rows and 9 columns, with no missing values whatsoever. This dataset includes numerous weather features recorded from January 2020 to January 2021 and, as already mentioned, it is common for these datasets to be well maintained due to their benchmark nature.

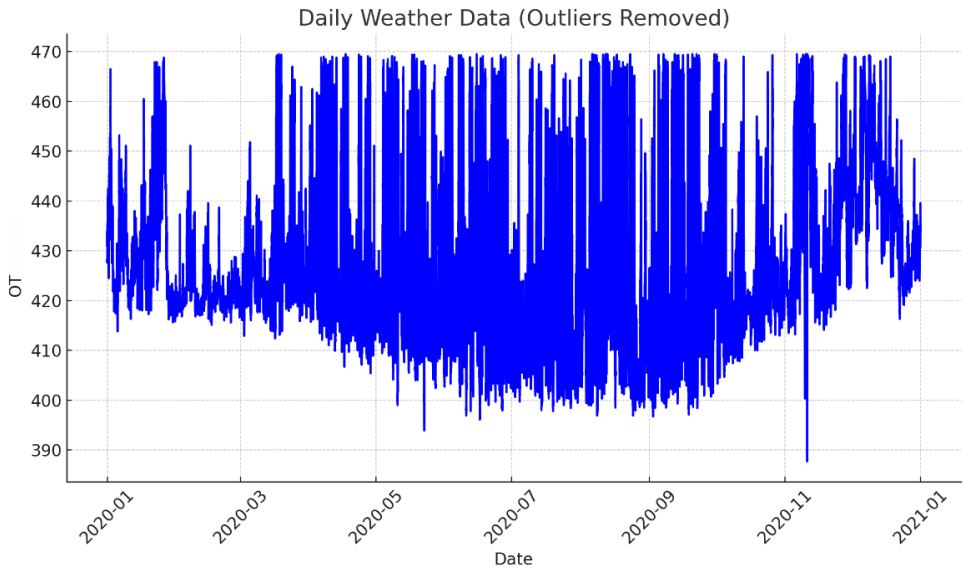

Figure 6 – Weather dataset plot – target variable.

Yet, it is noticeable in Figure 6, by plotting the target variable, that although there is a slight quadratic trend, the data exhibits slight daily variations - which is typical for weather data, as it can be seen in Figure 7, that displays data for one day.

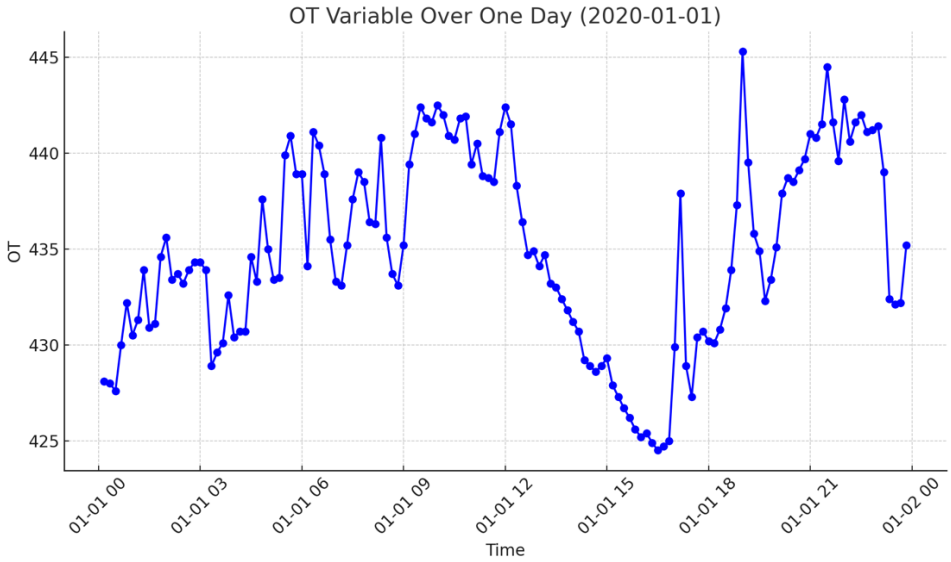

Figure 7 – OT variable plot (over one day).

It is yet to point out the two existing outliers that occurred right before March and September 2020, visible in Appendix A.7, which are errors of the dataset itself most likely originated from the data collection step. These outliers were replaced with similar values, and the correct data is now displayed in Figure 6

	Mean	Mode	Median	Standard Deviation
Target Variable (OT)	417.799	419.900	423.200	321.570

Table 3 –Weather dataset – target variable statistics.

Regarding the statistical measures of this variable presented in Table 3, the mean and median values are pretty close once again, indicating a relatively symmetric data distribution, as can also be seen in Figure 8. Lastly, the value for the standard deviation shows the variability in the data, which could be expected due to the nature of the weather changes.

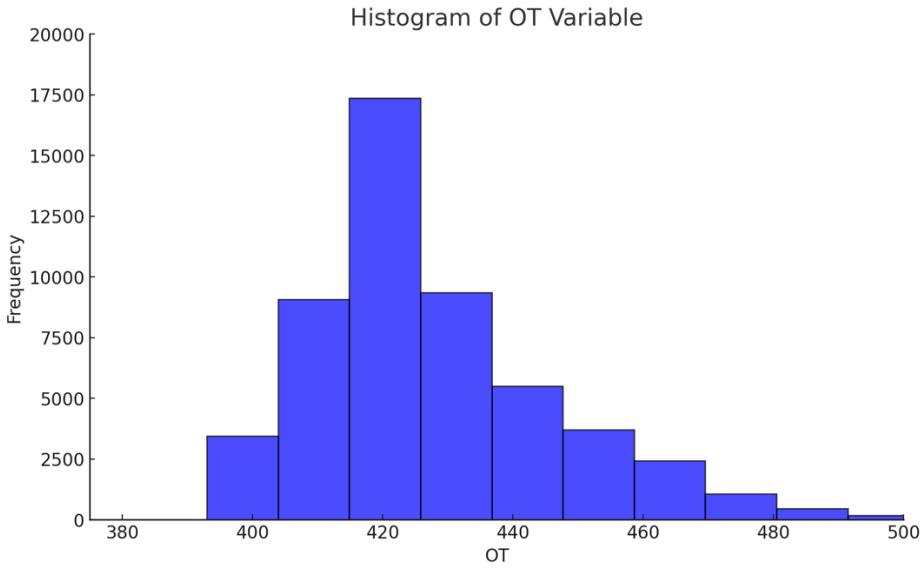

Figure 8 – OT variable histogram.

2.1.3 National Illness

The National Illness dataset contains data on illness activity levels across different regions, recorded weekly, from January 2002 to June 2020. It has 966 rows, 8 columns and no missing values.

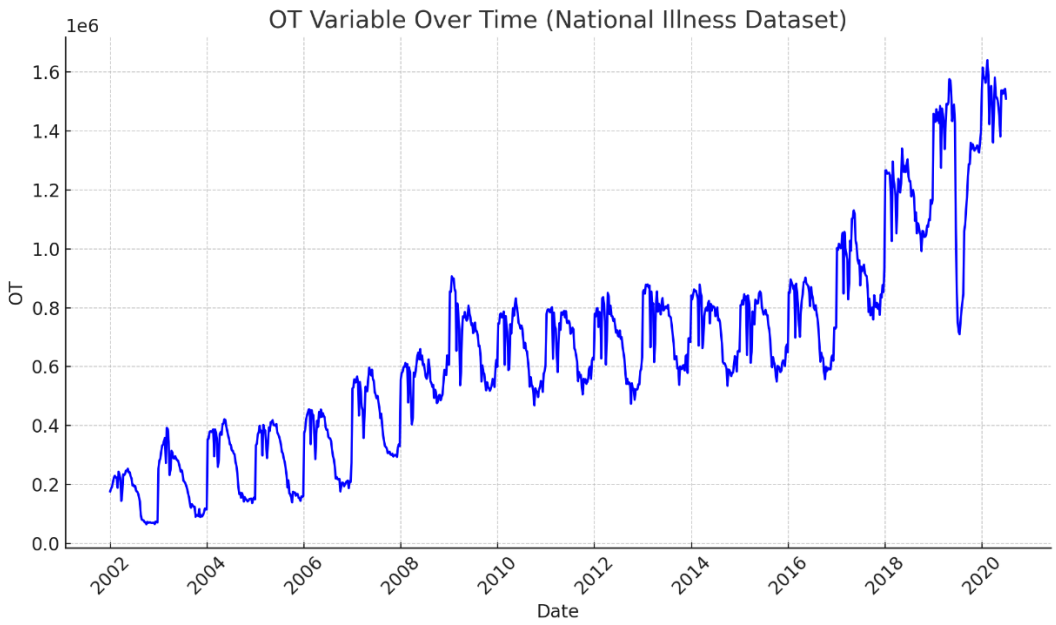

Figure 9 – Illness dataset plot – target variable.

Figure 9 shows the data exhibits noticeable weekly variations, with peaks corresponding to seasonal flu outbreaks. For example, around 2009, the swine flu pandemic, or the probable beginning of the COVID-19 pandemic around the end of 2019 is noticeable. Apart from that, no real outliers can be identified so no additional processing was performed.

	Mean	Mode	Median	Standard Deviation
Target Variable (OT)	651497.460	64699	618305	349018.888

Table 4 – Illness dataset – target variable statistics.

Regarding the statistical measures of the variable, Table 4 also shows that the large value for the standard deviation reflects a significant variability, which could be explained by the seasonal nature of illness data, where spikes surge during flu seasons and drop during off-peak periods.

2.1.4 Exchange Rate

The Exchange Rate dataset contains nine daily time series with 7588 entries and no missing values. This dataset records the exchange rates over the period ranging from January 1990 to October 2010, providing a view of the currency fluctuations.

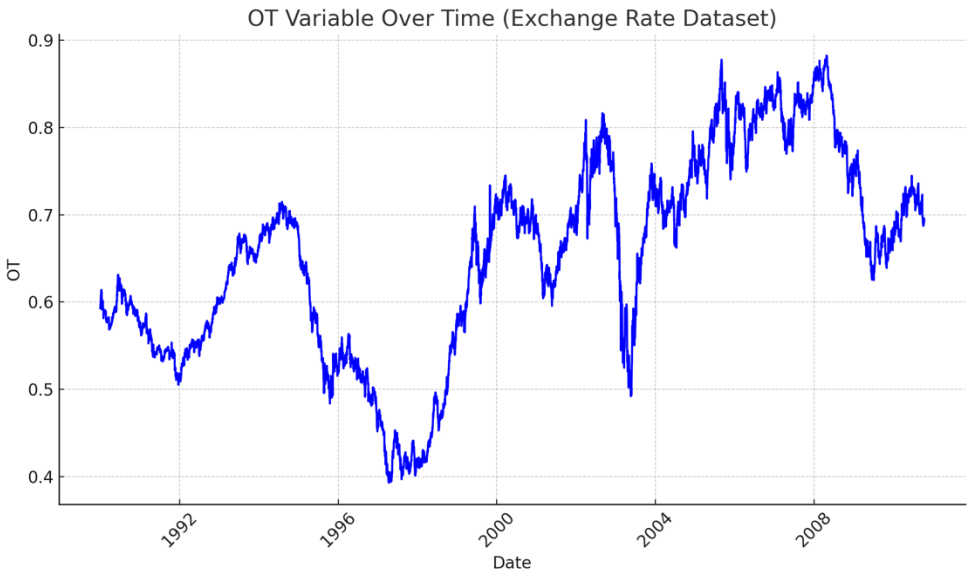

Figure 10 – Exchange rate dataset plot – target variable.

Figure 10 shows the data displays evident fluctuations over the years. There are periods of both rapid increase and decrease in the values, yet, it doesn't seem to present a real trend or seasonality typical to this type of data. No outliers were encountered either.

	Mean	Mode	Median	Standard Deviation
Target Variable (OT)	0.654	0.552	0.669	0.115

Table 5 – Exchange rate dataset – target variable statistics.

Regarding the statistical measures of the variable present in Table 5, the mean and median values provide insights into the central tendency of the data. The standard deviation indicates the extent of the variation, which highlights the volatility inherent to this type of data.

2.2 Layer integration

As one of the main ideas behind this thesis is the possible discovery of a novel model with the best results, it is in this section that is presented the implementation of the idea itself regarding the integration of the PatchTST layer into the FEDFormerMS model. To achieve that and provide context to better understand these models, a brief description of the original transformer model and the other three models (FEDFormerMS, PatchTST and the Novel Model) will also be made in what follows.

The original Transformer model revolutionized the field of sequence modeling as a whole. It can be characterized by using the self-attention mechanism to process input sequences (data), rather than sequentially as in traditional recurrent neural networks or other traditional models. This architecture consists of an encoder-decoder structure where the encoder processes the input sequence through multiple layers, each containing, precisely, a multi-head self-attention mechanism. This allows the model to focus on different parts of the sequence simultaneously and capture complex dependencies. The decoder generates the output sequence by similarly using self-attention layers. This dual attention mechanism is the feature that enabled the model to produce accurate and contextually relevant predictions in the sequence modeling field.

Regarding the FEDFormerMS – we can say that it is an implementation of the FEDFormer (T. Zhou et al., 2022) model that uses the Scaleformer framework to enhance multi-scale feature extraction. This model is designed to manage temporal sequences efficiently by capturing patterns across various scales. The architecture of FEDFormerMS includes a multi-scale decomposition layer that breaks down the input time series into components at different scales. This decomposition allows the model to focus on short-term and long-term patterns, enhancing its predictive abilities. Like the original transformer model, this model also uses a multi-head attention mechanism to emphasize relevant features across the decomposed data.

The PatchTST model is designed to effectively capture local dependencies within a time series through a unique approach that depends on patching – that is, the input time series is divided into smaller patches, each representing a localized data segment. This unique patching mechanism should, in theory, allow the model to focus on local temporal patterns, making it particularly expert at identifying short-term trends and variations in the data. Each patch is processed through a transformer encoder (already discussed briefly earlier), utilizing, once again, self-attention to capture dependencies within and between patches. The features extracted from these patches are then aggregated to form the final result.

Finally, regarding the novel model, the idea behind it is really what it seems – to integrate the layer responsible for most of the action in the PatchTST, namely this layers' capability to capture local dependencies model into the implementation of the FEDFormer of the scaleformer framework that, in itself, has the great ability of multi-scale feature extraction. In order to do this, the modules/layers of the PatchTST model regarding patching were gathered in the input section of the model and placed as the input layer of the FEDFormerMS in hopes of seeing, possibly, better results – illustration seen in Figure 11.

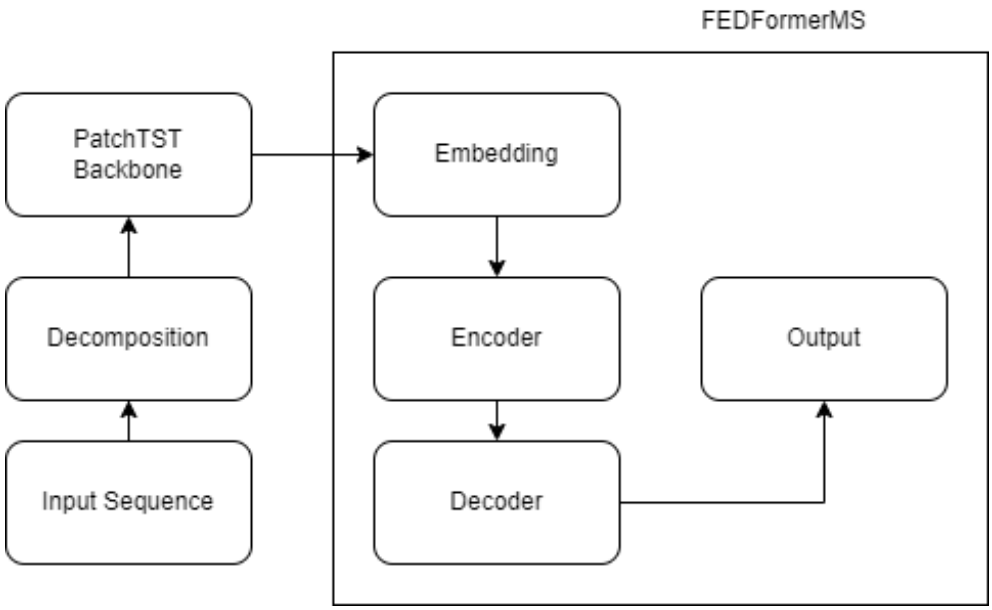

Figure 11 – Novel model architecture.

The rationale behind this process is to leverage the strengths of both models, aiming to achieve better results and performance by combining their capabilities. Process-wise, the input time series is split into patches using the PatchTST layer to better capture local dependencies, as already mentioned. After this step, the patched data is fed into the input layer of the FEDFormerMS, where the multi-scale feature extraction will occur.

It is by combining the best features of both models that this new hybrid model is expected to achieve better results (also regarding performance), when compared to each model individually.

2.3 Evaluation metrics

Finally, regarding the evaluation metrics to be used, once again, they will be assigned to the ones present in the selected literature as, in the future, when one wants to compare results, it will be not only helpful but truthful and scientifically correct if the comparison is made using the same evaluation metrics.

The metrics used in this novel model will also be the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). To better understand what they mean and the real meaning of the performance of the novel model, they will now be presented in their essence.

Starting with the Mean Squared Error (MSE) is simply the average of the squares of the errors, where the error is the difference between the predicted value, $\widehat{y_i}$, and the actual/real value, y_i, where n is the sample size. It is given by:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \qquad (2.1)$$

One aspect to consider for future result comparison is that MSE is sensitive to large errors once it squares the error term. Based on equation (2.1), it is also possible to conclude that a lower value for the MSE indicates a better model performance.

Regarding the metric Mean Absolute Error (MAE), it is possible to say that it is the average of the absolute differences between the predicted and actual/real values. It is given by:

$$\text{MAE} = \frac{1}{\text{n}}\sum_{\text{i}=1}^{\text{n}}|\text{y}_\text{i} - \widehat{y_\text{i}}| \qquad (2.2)$$

To have into consideration this metric, is the fact that, based on its modular nature, this metric provides a straightforward measure of the average magnitude of errors in a set of predictions, with no consideration for their direction, so it is possible to say that it is in the same scale as the target variable. Similarly to the MSE, this metric also indicated better model performance for lower values.

In conclusion, this chapter provides a comprehensive overview of the datasets to be used, the integration of the PatchTST layer into the FEDFormerMS as the base for this thesis aiming to deliver a superior performance to the individual 'parent' models that originate it, and finally a simple description of the evaluation metrics to be used in the novel model. The detailed discussion and analysis of the results obtained following this implementation will be held in the next chapter so that one can understand the real performance of the novel model compared to the two that originated it.

CHAPTER 3

Results and discussion

This chapter presents the results obtained from implementing the approach for the novel hybrid model and proposes to discuss them in the context of the selected models. This evaluation focuses on comparing the performance of the novel model against the FEDformer in the Scaleformer configuration and the PatchTST configuration. The comparison will be based on the evaluation metrics presented in the previous chapter - Mean Squared Error (MSE) and Mean Absolute Error (MAE). Finally, the discussion also addresses the time constraints encountered during the experimentation process, which required significant computational resources and time.

3.1 Results

By presenting and discussing these results, we aim to highlight the strengths and potential limitations of the novel hybrid model, as well as its relative performance against already established models. It is noted that several tests were performed, namely three different tests for each window for each dataset on the novel model – these tests were performed using the same settings, that is, the same window sizes, as the ones presented in the literature for each model.

For the other two original models, the results that will be considered are the best ones presented in the literature. The FEDformer model in the Scaleformer configuration has demonstrated robust performance across various datasets, as already discussed in the first chapter. This configuration leverages multi-scale feature extraction to capture patterns, providing a balanced approach to short-term and long-term dependency modeling regarding time series forecasting.

Finally, the configuration used to obtain the results for the novel model consisted of the following: (1) Data was split into 70%, 20%, and 10%, respectively, for train, test, and validation subsets. (2) Regarding training, 10 was the number of epochs chosen, with a batch size 32. (3) The window size varied depending on the dataset; as already mentioned, used the same values present in the literature for each dataset: 96,192,336 and 720 for the Traffic, Weather and Exchange Rate datasets, and 24,36,48 and 60 for the ILI dataset. (4) It is also relevant to note that the developed code[4] was executed in a notebook in Google Colab using NVIDIA A100 GPUs. (5) Regarding the implementation of the PatchTST part in the novel model, a patch length of 16 and a stride of 8 were used, meaning that the time series data was divided into patches of 16-time steps each, and these patches were created with an overlap where each new patch started 8-time steps after the previous one.

3.1.1 Scaleformer (FEDFormerMS)

In the Traffic dataset, the Scaleformer configuration achieved an MSE of 0.564 and an MAE of 0.351 at the 96 window size. At a 192 window, it achieved an MSE of 0.570 and an MAE of 0.349. At a 336 window, it achieved an MSE of 0.576 and an MAE of 0.439. At a 720 window, the MSE was 0.602 and the MAE was 0.360. These results indicate that while the Scaleformer performs well at capturing traffic patterns, there is a trend of increasing MSE and MAE as the window size increases, suggesting that longer windows may introduce more complexity and variability that the model struggles to capture effectively.

For the Weather dataset, the Scaleformer implementation achieved an MSE of 0.220 and an MAE of 0.289 at the 96 window size. For a 192 window, the MSE was 0.341 and the MAE was 0.385. At a 336 window, the MSE was 0.463 and the MAE was 0.455. At a 720 window, the MSE was 0.682 and the MAE was 0.565. These results show that the Scaleformer implementation could handle short-term weather variations well but struggles as the window size increases, possibly due to the increasing complexity and variability in the weather data over longer periods.

In the Illness dataset, the Scaleformer configuration reported an MSE of 2.745 and an MAE of 1.075 at the 24 window size. At a 32 window, the MSE was 2.748 and the MAE was 1.072. For a 48 window, the MSE was 2.793 and the MAE was 1.059. At a 64 window, the MSE was 2.678 and the MAE was 1.071.

[4] Developed code shall be provided upon request.

Finally, the Exchange Rate dataset presented at the 96 window an MSE of 0.109 and an MAE of 0.240. At a 192 window, the MSE was 0.241 and the MAE was 0.353. At a 336 window, the MSE was 0.471 and the MAE was 0.508. At a 720 window, the MSE was 1.259 and the MAE was 0.865.

Dataset	**Traffic**		**Weather**		**Illness**		**Exchange Rate**	
Metric	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
24	-	-	-	-	2.745	1.075	-	-
32	-	-	-	-	2.748	1.072	-	-
48	-	-	-	-	2.793	1.059	-	-
64	-	-	-	-	2.678	1.071	-	-
96	0.564	0.351	0.220	0.289	-	-	0.109	0.240
192	0.570	0.349	0.341	0.385	-	-	0.241	0.353
336	0.576	0.439	0.463	0.455	-	-	0.471	0.508
720	0.602	0.360	0.682	0.565	-	-	1.259	0.865

Table 6 – Results for the FEDFormerMS model.

These results presented in Figures 12 and 13 suggest, as it can also be observed in Table 6, that the Scaleformer configuration could be handling short-term volatility well but struggles with longer windows that could, in this case, be derived due to increased variability in exchange rates (assuming it could be precisely due to the nature of the data in question).

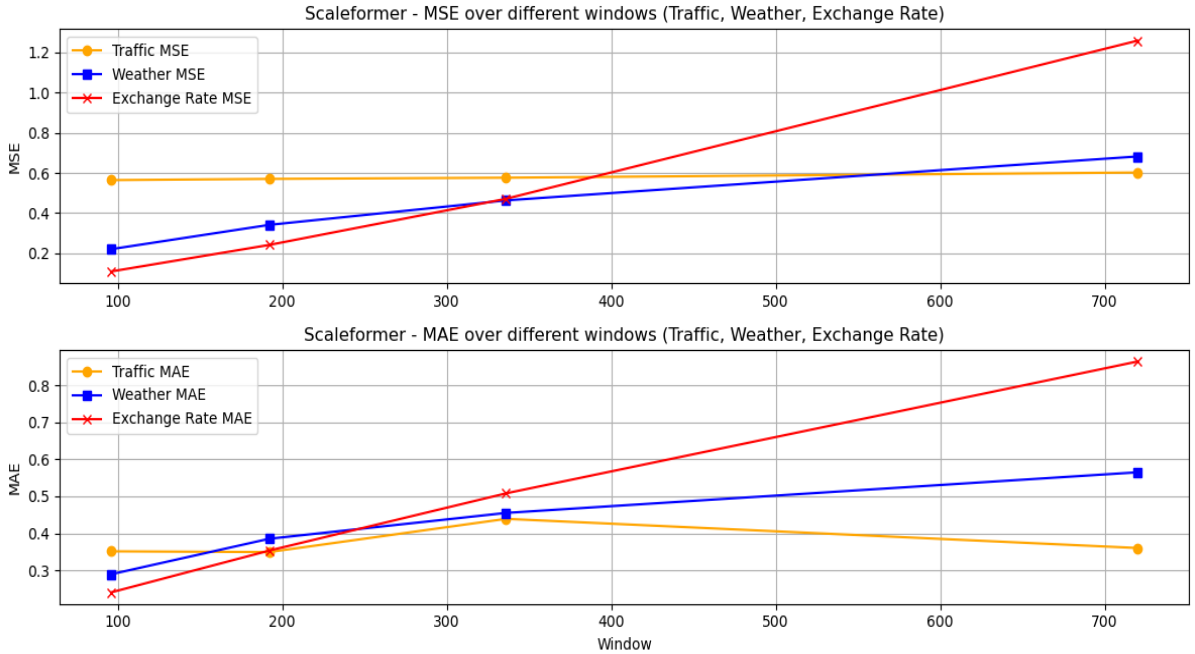

Figure 12 – Results for the Scaleformer model (Traffic, Weather, Exchange Rate).

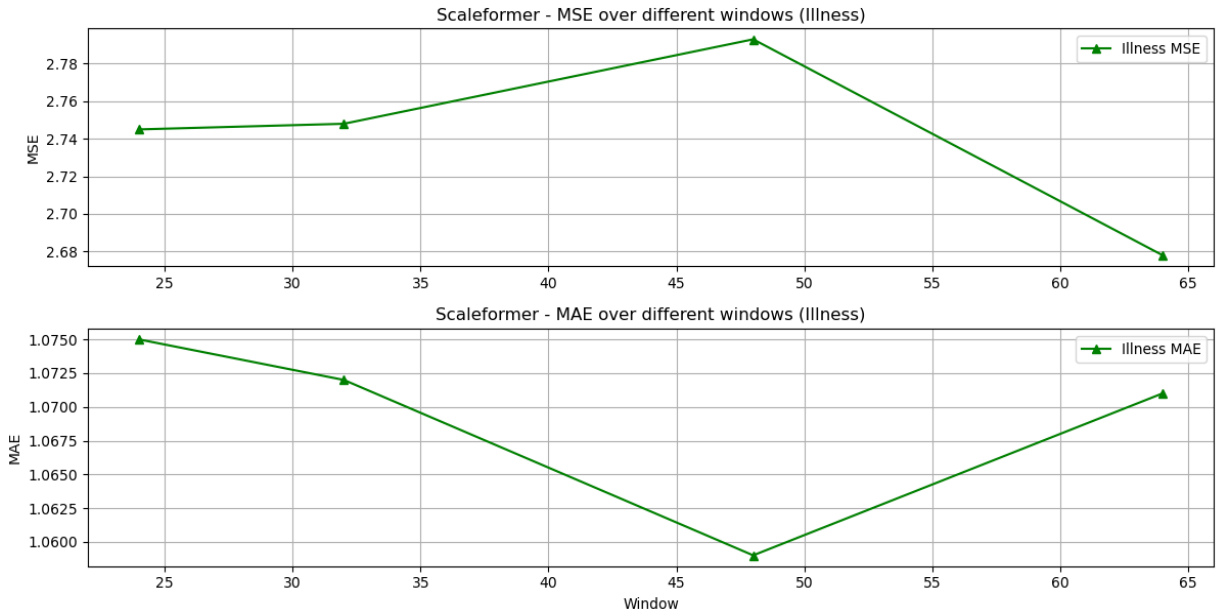

Figure 13 – Results for the Scaleformer model (Illness).

Regarding the interpretation, based on the smallest MAE values, it is possible to conclude the following: (1) For the Traffic dataset, the smallest MAE is 0.349 for the 192 window size. (2) For the Weather dataset, the smallest MAE is 0.289 for the 96 window size. (3) Regarding the Illness dataset, the smallest MAE is 1.059 for the 48 window size. (4) For the Exchange Rate dataset the smallest MAE is 0.240 for the 96 window size. That said, it is possible to conclude that this model performed better when encountering smaller window sizes.

Considering this and Question 1 formulated in the first section of this study, we can conclude that the Scaleformer configuration of the FEDformer model proved to be a robust model across various datasets, balancing short-term and long-term dependencies effectively. The results indicate, once again, that smaller window sizes, particularly the 96 window, could effectively capture complex patterns in the data, but performance tends to decrease as the window size increases, likely due to the increased complexity and variability in the data over longer periods, which can provide an answer to Question 2. It was also interesting to notice that, no matter the subject into which the data is inserted, this phenomenon stays constant when facing larger windows. However, it was noticed that, due to the nature of the data, the results may vary.

3.1.2 PatchTST (64)

Now, for the PatchTST model, all the following results will be regarding the variation that contemplates patches of 64 (that is why it is being referred to as PatchTST (64)), whereas this was the model's variation with the best results. This model also showed strong performance, particularly in capturing local dependencies within the time series data.

Dataset	**Traffic**		**Weather**		**Illness**		**Exchange Rate**	
Metric	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
24	-	-	-	-	1.319	0.754	-	-
32	-	-	-	-	1.579	0.870	-	-
48	-	-	-	-	1.553	0.815	-	-
64	-	-	-	-	1.470	0.788	-	-
96	0.360	0.249	0.149	0.198	-	-	-	-
192	0.379	0.256	0.194	0.241	-	-	-	-
336	0.392	0.264	0.245	0.282	-	-	-	-
720	0.432	0.286	0.314	0.334	-	-	-	-

Table 7 – Results for the PatchTST model.

As seen in Table 7 and Figure 14, in the Traffic dataset, the PatchTST (64) configuration achieved an MSE of 0.360 and an MAE of 0.249 at the 96 window size. At a 192 window, the MSE was 0.379, and the MAE was 0.256. At a 336 window, the MSE was 0.392, and the MAE was 0.264. At a 720 window, the MSE was 0.432, and the MAE was 0.286. These results could indicate that the patching mechanism effectively captures localized traffic patterns and maintains relatively stable performance across different window sizes, compared to the scale former, although there is a slight increase in error metrics with more oversized windows.

For the Weather dataset, the PatchTST (64) configuration achieved an MSE of 0.149 and an MAE of 0.198 at the 96 window size. At a 192 window, the MSE was 0.194, and the MAE was 0.241. At a 336 window, the MSE was 0.245, and the MAE was 0.282. At a 720 window, the MSE was 0.314, and the MAE was 0.334. These results indicate that the patching mechanism could be particularly effective for dealing with highly variable data (as weather data), maintaining relatively strong performance even as the window size increases.

In the Illness dataset, the PatchTST (64) configuration reported an MSE of 1.319 and an MAE of 0.754 at the 24 window size. At a 32 window, the MSE was 1.579, and the MAE was 0.870. At a 48 window, the MSE was 1.553, and the MAE was 0.815. Finally, at the 64 window, the MSE was 1.470 and the MAE was 0.788. Also, from Figure 15, these results suggest that while the patching mechanism is beneficial for capturing localized spikes and trends, its performance curiously decreases with intermediate window sizes.

For the Exchange Rate dataset, the PatchTST (64) unfortunately didn't have any results, so it would be impossible to make any judgements or get any insights.

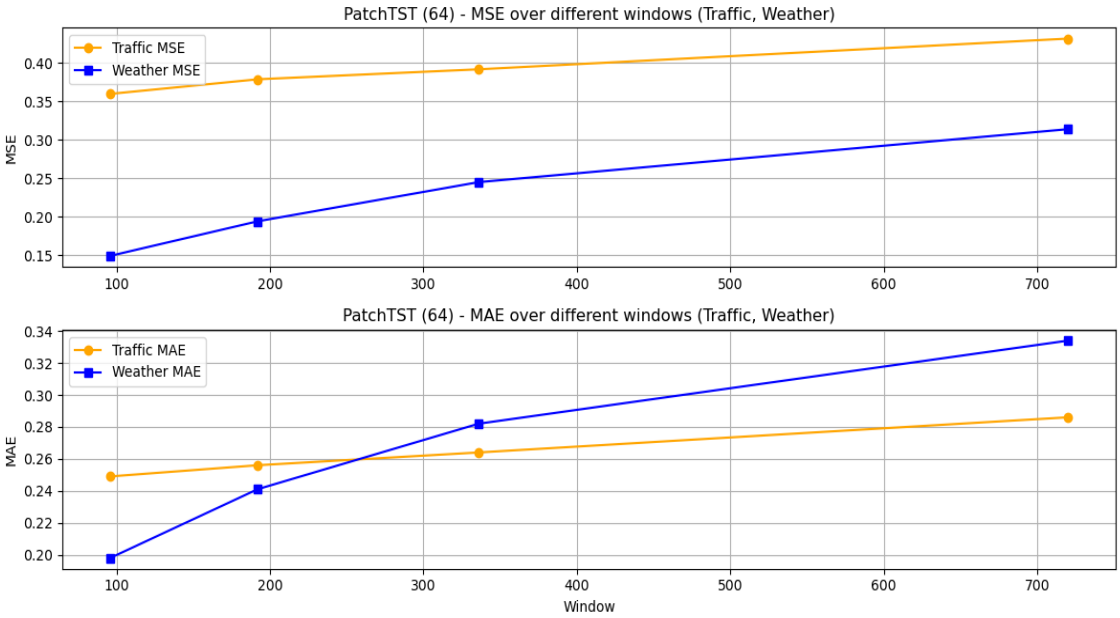

Figure 14 – Results for the PatchTST model (Traffic, Weather).

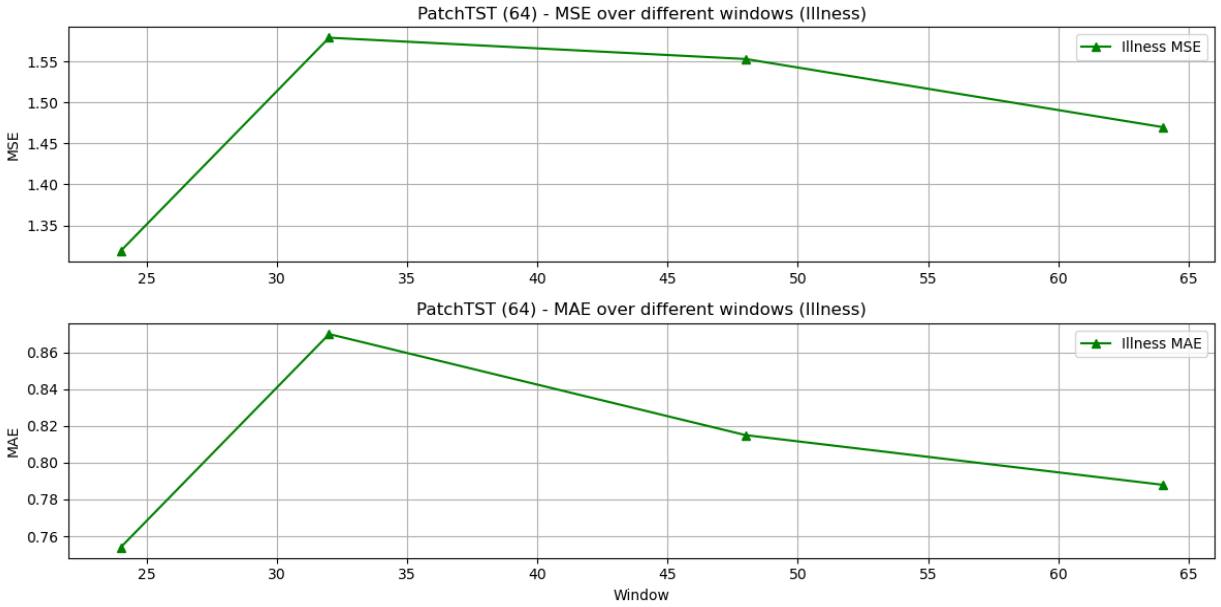

Figure 15 – Results for the PatchTST model (Illness).

Overall, the PatchTST (64) showed strong performance across all datasets, particularly excelling in capturing local dependencies – as advertised. This suggests, and once again taking into consideration Questions 1 and 2, that the patching mechanism enhances the model's ability to manage detailed, localized patterns within the data, although performance tends to decrease with larger window sizes due to increased complexity and variability. Although this last sentence is true, it is worth noting that although the errors seem to grow with the increase in the window, this increase for the PatchTST (64) is smaller than the increase in the Scaleformer implementation.

3.1.3 Novel hybrid model

Our novel hybrid model will then integrate the PatchTST layer responsible for the patching into the FEDformerMS input layer's architecture, aiming to combine the strengths of both configurations to get better results – it is essential to state that, because both the code for the PatchTST and FEDFormerMS models were implemented in Python using PyTorch, it was, in a general way, easier to develop the code for the novel model. The results indicate a mixed performance, with notable improvements in some areas and challenges in others. It is important to note that the following results were achieved using a patch length equal to 16, a stride of 8 (meaning each patch overlaps the next one by eight-time steps), and two subsequent layers of the PatchTST.

Dataset	Traffic		Weather		Illness		Exchange Rate	
Metric	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
24	-	-	-	-	2.272	0.958	-	-
32	-	-	-	-	2.701	1.066	-	-
48	-	-	-	-	2.745	1.067	-	-
64	-	-	-	-	2.644	1.036	-	-
96	0.555	0.324	0.218	0.288	-	-	0.122	0.246
192	0.569	0.332	0.303	0.343	-	-	0.189	0.320
336	0.586	0.333	0.401	0.414	-	-	0.316	0.411
720	0.613	0.347	0.655	0.551	-	-	1.290	0.859

Table 8 – Results for the Novel model.

In the Traffic dataset, and observing Table 8 and Figure 16, the hybrid model achieved an MSE of 0.555 and an MAE of 0.324 at the 96 window size. At a 192 window, the MSE was 0.569, and the MAE was 0.332. At a 336 window, the MSE was 0.586, and the MAE was 0.333. At a 720 window, the MSE was 0.613, and the MAE was 0.347. While this performance is slightly poorer than the PatchTST (64) configuration, it is comparable to the Scaleformer configuration, indicating that the hybrid model could effectively balance local and global dependencies in the data, especially for smaller windows. Yet the results show a trend in increasing MSE and MAE values as the window size increases, consistent with the behavior observed in the Scaleformer and PatchTST (64) configurations.

For the Weather dataset, the hybrid model achieved an MSE of 0.218 and an MAE of 0.288 at the 96 window size. At a 192 window, the MSE was 0.303, and the MAE was 0.343. At a 336 window, the MSE was 0.401, and the MAE was 0.414. At a 720 window, the MSE was 0.655, and the MAE was 0.551. This performance appears between the Scaleformer and PatchTST (64) configurations results, suggesting that the hybrid model captures weather variations reasonably well but does not significantly outperform the individual configurations. Once again, the increasing error metrics with larger windows indicate challenges in handling longer-term weather variability.

In the Illness dataset, the hybrid model reported an MSE of 2.272 and an MAE of 0.958 at the 96 window size. At a 192 window, the MSE was 2.701, and the MAE was 1.066. At a 336 window, the MSE was 2.745, and the MAE was 1.067. At a 720 window, the MSE was 2.644, and the MAE was 1.036. As seen in Figure 17, this performance is better than the Scaleformer configuration but slightly worse than the PatchTST (64) configuration, indicating that the hybrid model could be capturing seasonal trends correctly but may be struggling with localized spikes in the data (which could make total sense knowing what data we are talking about, subject-wise). The performance trend suggests that while the hybrid model can manage longer-term trends, it probably faces challenges with the increased variability in longer windows.

Finally, regarding the Exchange Rate dataset, a similar pattern was presented, with the hybrid model achieving an MSE of 0.122 and an MAE of 0.246 at the 96 window size. At a 192 window, the MSE was 0.189, and the MAE was 0.320. At a 336 window, the MSE was 0.316, and the MAE was 0.411. At a 720 window, the MSE was 1.290, and the MAE was 0.859.

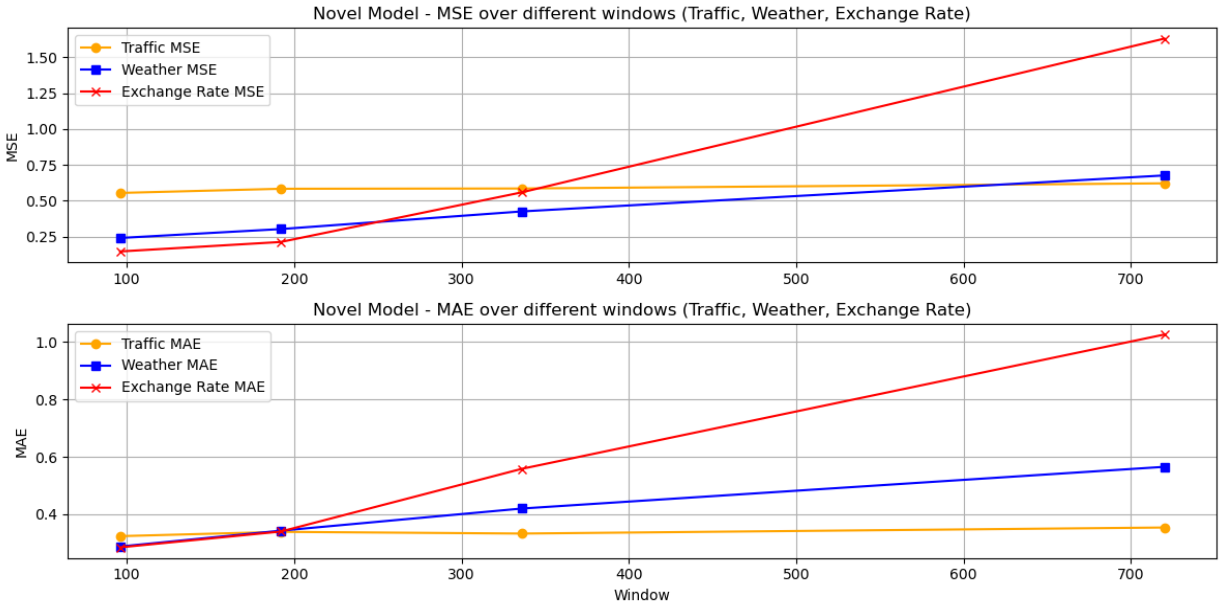

Figure 16 – Results for the Novel model (Traffic, Weather, Exchange Rate).

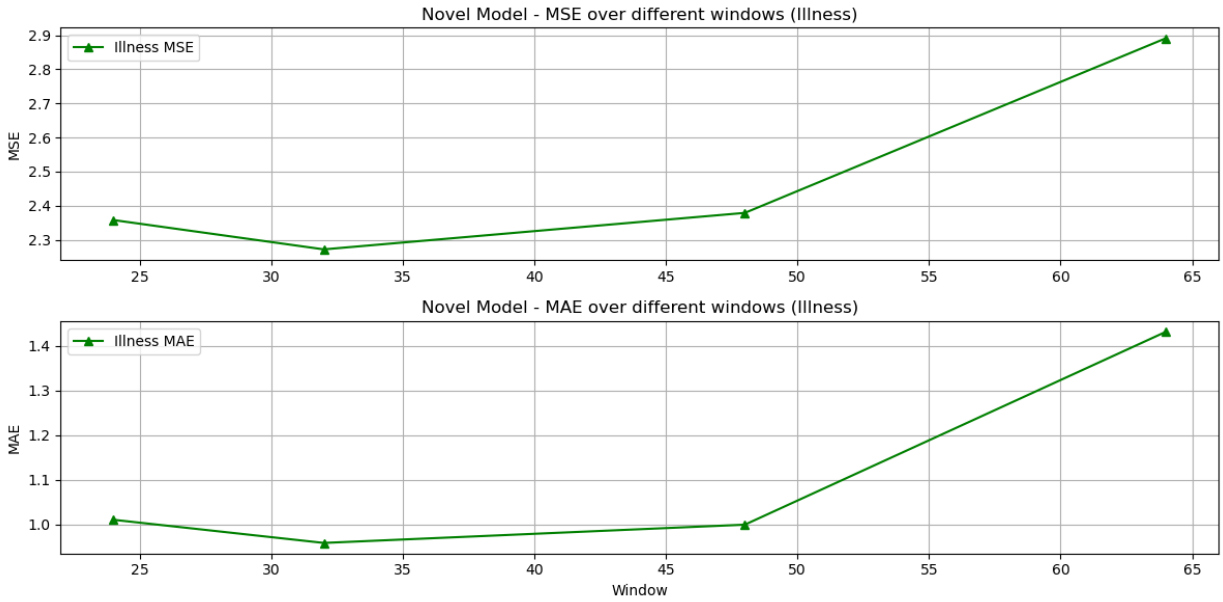

Figure 17 – Results for the Novel model (Illness).

This performance is only comparable to the Scaleformer configuration once the PatchTST (64) model doesn't contemplate results for this dataset. The results may indicate that the hybrid model could effectively handle volatility in the data but may not fully leverage the strengths (or at least not as much as the Scaleformer) of the patching mechanism provided by the integrated layer. The constant and recurring increasing error metrics with larger windows highlight the challenges in managing longer-term volatility in exchange rate data (which once again could make all sense knowing what data we are talking about, subject-wise).

Once again, and considering Question 2, it is essential to note that these results were obtained using arguments for the patch length, stride, and number of layers, the numbers mentioned earlier – and the results will vary when changing these arguments.

3.2 Discussion

It is also important to note that one significant challenge encountered with the hybrid model was the time constraint, particularly with the Weather dataset. Running the code for this dataset required substantial computational resources and time, highlighting the increased complexity and resource demands of this novel hybrid architecture. This challenge underscores the importance of considering computational efficiency alongside predictive performance in model development, which should be considered for future work regarding time duration for time series forecasting.

When comparing the performance of the three models, we observed a clear pattern related to window size limitations across all datasets. Each model exhibits a trend of increasing error metrics as the window size increases, which highlights the inherent complexity and variability that longer windows introduced – when grouped by dataset in order of the model, it is possible to conclude, as it is possible to observe in Figures 18 and 19.

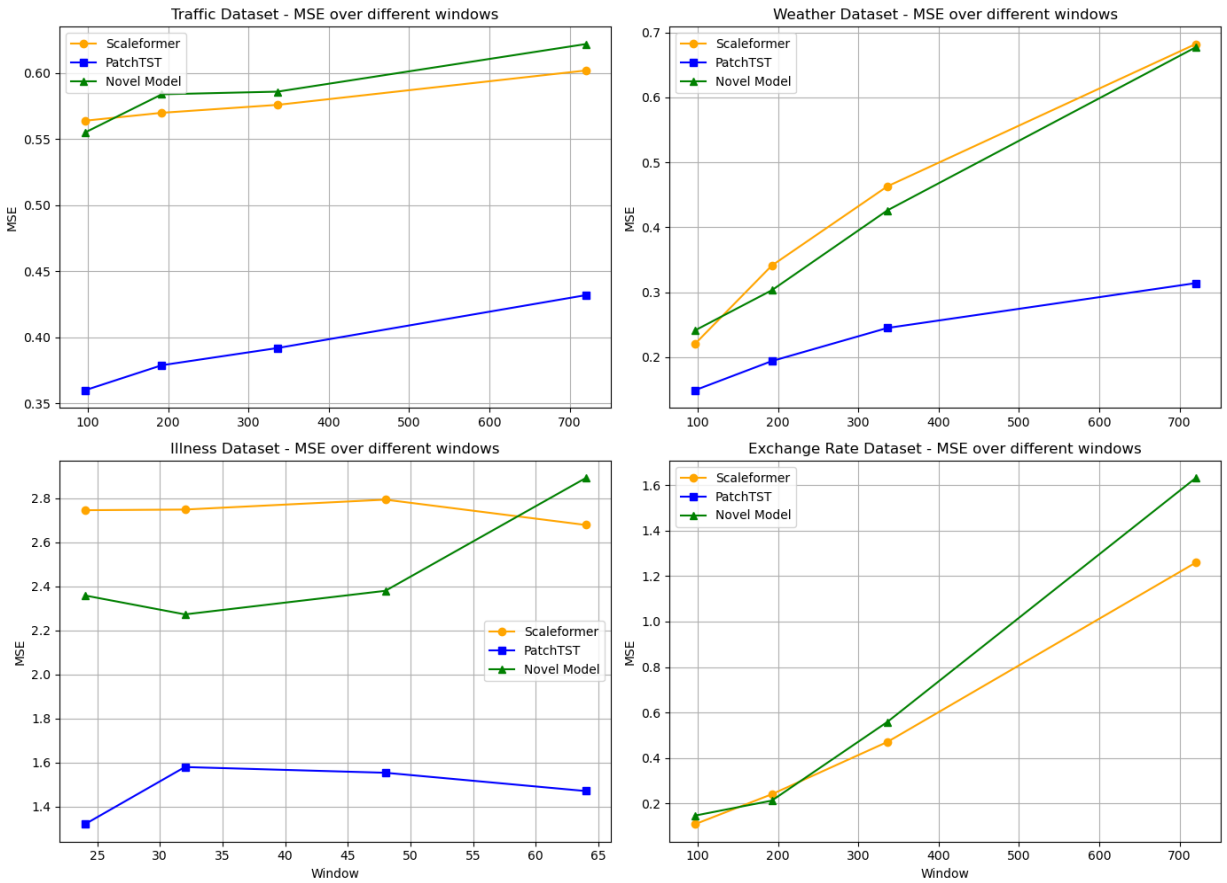

Figure 18 – MSE Results for the three models by dataset.

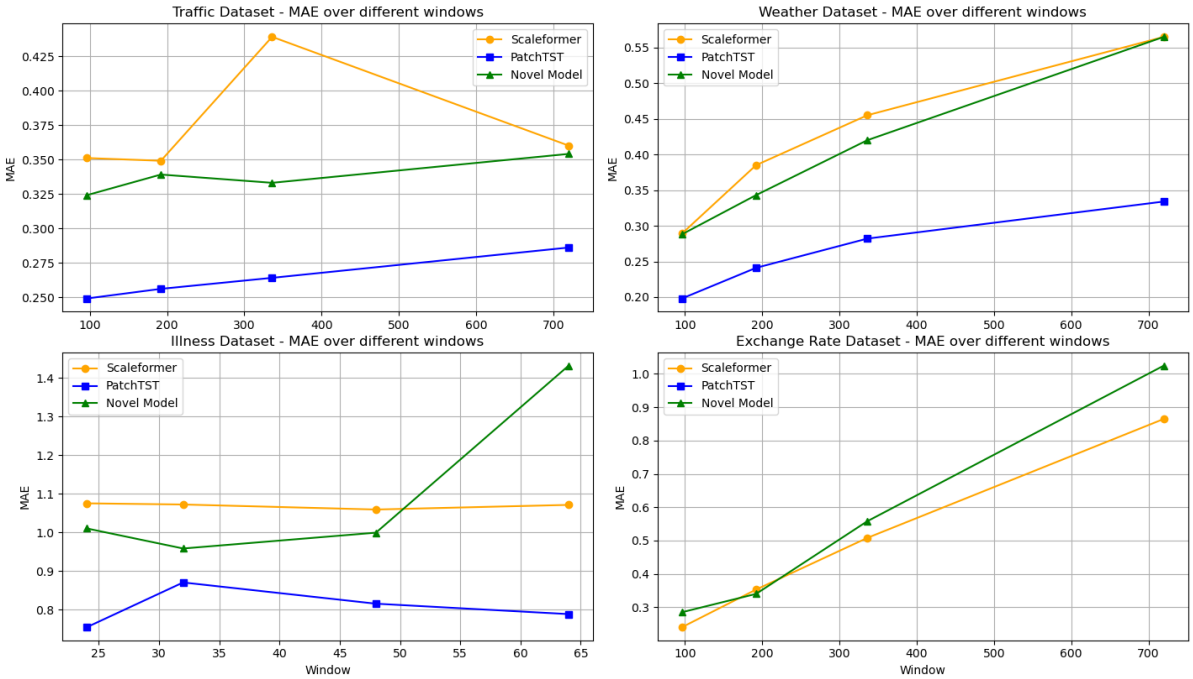

Figure 19 – MAE Results for the three models by dataset.

The Scaleformer implementation demonstrated robust performance, particularly with smaller window sizes. For instance, in the Traffic dataset, the Scaleformer achieved an MSE of 0.564 and an MAE of 0.351 at the 96 window size. However, as the window size increases to 192, 336, and 720 hours, the error metrics also increase, indicating that the model struggles to maintain accuracy over longer periods. This pattern is consistent across other datasets where the Scaleformer shows strong performance with short-term data but faces challenges with long-term variability.

On the other hand, the PatchTST (64) configuration excels in capturing local dependencies, which is reflected in its superior performance across all window sizes and datasets compared to the Scaleformer. In the Traffic dataset, the PatchTST (64) achieved an MSE of 0.360 and an MAE of 0.249 at the 96 window size, outperforming the Scaleformer. Even as the window size increases, the degradation in performance is less severe than that observed with the Scaleformer. For example, at a 720 window, the PatchTST (64) recorded an MSE of 0.432 and an MAE of 0.286, maintaining a relatively stable performance. On the other hand, the Scaleformer had worse results for this bigger window. This trend is also evident across other datasets, where the PatchTST (64) consistently ranks as the top-performing model due to its ability to oversee localized patterns effectively.

Finally, our novel hybrid model that integrates the PatchTST layer into the FEDformerMS architecture, aiming to balance the strengths of both configurations, returned results that indicate that the hybrid model generally performs better than the Scaleformer implementation, but not as well as the PatchTST (64) model. For example, in the Traffic dataset, the hybrid model achieved an MSE of 0.555 and an MAE of 0.324 at the 96 window size, positioning itself between the two individual models that originated it. As the window size increases, the hybrid model's performance remains closer to that of the PatchTST (64) than the Scaleformer, indicating a successful integration that leverages the strengths of both approaches, namely the integration of the PatchTST layer. However, as it is a universal problem, like the other models, the hybrid model also shows an inevitable increase in error metrics when presented with longer windows, highlighting precisely the universal challenge of managing longer-term dependencies and variability for the task of time series forecasting.

When ranking the models based on their performance across different datasets and window sizes, the PatchTST (64) model emerges as the best overall performer, followed by the novel hybrid model, with the Scaleformer configuration coming in third. The PatchTST (64) model consistently shows lower MSE and MAE values, indicating its robustness in handling both short-term and long-term dependencies more effectively than the other models. The novel hybrid model, while not outperforming the PatchTST (64), demonstrates a significant improvement over the Scaleformer, indicating that the integration of the PatchTST layer adds value by enhancing the model’s ability to capture localized patterns, although it could be dragged down by time and computation affairs. Finally, the Scaleformer, despite its robust performance with smaller window sizes, also struggles with larger windows. This limitation underscores the importance of model architecture in handling different types of time series data, particularly when dealing with longer-term forecasts. Once again, the consistent trend of increasing error metrics across all models with larger windows highlights a common challenge in time series forecasting: balancing short-term and long-term dependencies while managing the increased complexity and variability of longer prediction horizons.

In conclusion, while each model has strengths and weaknesses, the PatchTST (64) stands out, with pity, as the most effective configuration for time series forecasting across various datasets and window sizes. Following this, the novel hybrid model shows promising results, by bridging the gap between the PatchTST (64) model and the Scaleformer implementation, offering a balanced approach to this task that leverages the strengths of both models. Finally, The Scaleformer, while robust in certain scenarios, also faces notable challenges with longer windows, emphasizing the possible need for further refinement in handling long-term dependencies. Future work presented in the next chapter should focus on optimizing the novel model to improve its performance with possibly larger windows, ensuring more accurate and reliable forecasts across different time series datasets. An emphasis on time constraints and computational resources should also be placed on future improvement, where more tests should be done with varying arguments regarding patch length, stride, and number of layers, as these will influence the results and time-related issues.

CHAPTER 4

Conclusions and future work

This study explored the potential of a novel hybrid model for time series forecasting, integrating the PatchTST input layer into the FEDFormerMS architecture. The study aimed to leverage and merge the strengths of both configurations - PatchTST's ability to capture local dependencies and FEDFormerMS's robust multi-scale feature extraction- and answer the questions proposed in the first chapter. The obtained results indicated that while the novel model presented notable improvements in some areas (particularly in balancing local and global dependencies), it did not, unfortunately, consistently outperform the PatchTST model that originated it.

One significant challenge during this research was the time constraint, particularly with the Weather dataset. The increased complexity of the hybrid model derived from integrating two distinct and very different models required a substantial number of computational resources and time, highlighting the importance of considering computational efficiency alongside more extensive testing regarding the arguments to find the best configuration for this novel model. This constraint underscores the need for future work to improve accuracy and optimize computational efficiency. It could also raise some challenges while simultaneously leading to further development of Questions 2 and 3.

The analysis presented in this study showed that the PatchTST configuration consistently outperformed the Scaleformer implementation and our hybrid model in terms of Mean Squared Error (MSE) and Mean Absolute Error (MAE). The PatchTST model's superior performance could be attributed to its effective handling of localized patterns within the data. While showing promise, our hybrid model demonstrated performance that fell between the PatchTST and Scaleformer implementations, indicating a successful integration and highlighting obvious improvement areas.

Given the mixed results, it is evident that hybrid models combining various elements of different architectures could hold potential for improvement regarding the task of time series forecasting yet, and directly answering to Question 1, the transformer model proved to have success in time series forecasting (at least to some extent). Future research should explore Question 3 more deeply, namely integrating other transformer-based models or traditional forecasting techniques into hybrid architectures. This approach could harness the unique strengths of each model, potentially leading to better overall performance and making transformer-based models even more powerful for this task. Additionally, addressing the computational challenges is crucial alongside more extensive testing with more computational resources.

Throughout this study, as transformers are a prominent topic in the data science field, several alternative models and ideas have emerged. Notable examples include the already mentioned Informer, Autoformer, PatchTST, and the Temporal Fusion Transformer (TFT) models. These models are gaining traction and being incorporated into libraries such as Hugging Face's time series section[5]. Additionally, practitioners like Marco Peixeiro are also leveraging the mentioned frameworks, such as DARTS and NeuralForecast[6]. Finally, it is also to note that there are also implementations available in PyTorch, such as the Transformers-for-timeseries notebook on Google Colab[7] which is also a great way to break through this topic. These developments highlight the continuous evolution and innovation of the topic within the field, providing promising directions for future research.

In conclusion, this research contributes to the field of time series forecasting by providing insights into the potential and limitations of transformer-based models for time series forecasting. While our novel model did not consistently outperform existing solutions, it paved the way for future exploration into more sophisticated hybrid models that could present researchers with better results. Future work should focus on optimizing these models for both accuracy and computational efficiency, ensuring they can, even more effectively, handle the complexities of long-term time series forecasting. Through continued innovation and rigorous testing, it is possible to develop more robust and efficient forecasting models to significantly improve prediction accuracy in various application domains.

[5] https://huggingface.co/docs/transformers/model_doc/patchtst
[6] https://github.com/marcopeix/datasciencewithmarco
[7] https://colab.research.google.com/github/charlesollion/dlexperiments/blob/master/7-Transformers-Timeseries/Transformers_for_timeseries.ipynb

References

Amin Shabani, M., Abdi, A., Meng, L., & Sylvain Borealis, T. A. (2022). *SCALEFORMER: ITERATIVE MULTI-SCALE REFINING TRANSFORMERS FOR TIME SERIES FORECASTING*. https://github.com/BorealisAI/scaleformer.

Cai, L., Janowicz, K., Mai, G., Yan, B., & Zhu, R. (2020). Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*, *24*(3), 736–755. https://doi.org/10.1111/tgis.12644

Chimmula, V. K. R., & Zhang, L. (2020). Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos, Solitons and Fractals*, *135*. https://doi.org/10.1016/j.chaos.2020.109864

Ding, Q., Wu, S., Sun, H., Guo, J., Guo, J., & Laboratory, P. C. (2020). *Hierarchical Multi-Scale Gaussian Transformer for Stock Movement Prediction*. https://data.worldbank.org/indicator/CM.MKT.TRAD.CD?end=

Lara-Benítez, P., Gallego-Ledesma, L., Carranza-García, M., & Luna-Romera, J. M. (2021). *Evaluation of the Transformer Architecture for Univariate Time Series Forecasting*.

Lezmi, E., & Xu, J. (2023). *Time Series Forecasting with Transformer Models and Application to Asset Management*.

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., & Yan, X. (2019). *Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting*. http://arxiv.org/abs/1907.00235

Lim, B., Arık, S., Loeff, N., & Pfister, T. (2021). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

Lim, B., & Zohren, S. (2020). *Time Series Forecasting With Deep Learning: A Survey*. https://doi.org/10.1098/rsta.2020.0209

Lin, Y., Koprinska, I., & Rana, M. (2021). *SSDNet: State Space Decomposition Neural Network for Time Series Forecasting*. http://arxiv.org/abs/2112.10251

Liu, S., Yu, H., Liao, C., Li, J., Lin, W., Liu, A. X., & Dustdar, S. (2022). *PYRAFORMER: LOW-COMPLEXITY PYRAMIDAL AT-TENTION FOR LONG-RANGE TIME SERIES MODELING AND FORECASTING*. https://github.com/alipay/Pyraformer

Liu, Y., Wu, H., Wang, J., & Long, M. (2022). *Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting*. https://github.com/thuml/Nonstationary_Transformers.

Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2022). *A Time Series is Worth 64 Words: Long-term Forecasting with Transformers*. http://arxiv.org/abs/2211.14730

Oliveira, H. S., & Oliveira, H. P. (2023). Transformers for Energy Forecast. *Sensors*, *23*(15). https://doi.org/10.3390/s23156840

Shen, L., & Wang, Y. (2022). TCCT: Tightly-coupled convolutional transformer on time series forecasting. *Neurocomputing*, *480*, 131–145. https://doi.org/10.1016/j.neucom.2022.01.039

Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2018). A Comparison of ARIMA and LSTM in Forecasting Time Series. *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, 1394–1401. https://doi.org/10.1109/ICMLA.2018.00227

Sreelekshmy Selvin, Vinayakumar R, Gopalakrishnan E.A, Vijay Krishna Menon, & Soman K.P. (2017). *STOCK PRICE PREDICTION USING LSTM,RNN AND CNN-SLIDING WINDOW MODEL*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. http://arxiv.org/abs/1706.03762

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2022). *Transformers in Time Series: A Survey*. http://arxiv.org/abs/2202.07125

Wirth, R., & Hipp, J. (2000). *CRISP-DM: Towards a Standard Process Model for Data Mining*. https://cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf

Wu, H., Xu, J., Wang, J., & Long, M. (2021). *Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting*. http://arxiv.org/abs/2106.13008

Wu, N., Green, B., Ben, X., & O'Banion, S. (2020). *Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case*. http://arxiv.org/abs/2001.08317

Wu, S., Xiao, X., Ding, Q., Zhao, P., Wei, Y., & Huang, J. (2020). *Adversarial Sparse Transformer for Time Series Forecasting*.

Xue, W., Zhou, T., Wen, Q., Gao, J., Ding, B., & Jin, R. (2023). *Make Transformer Great Again for Time Series Forecasting: Channel Aligned Robust Dual Transformer*. http://arxiv.org/abs/2305.12095

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023). *Are Transformers Effective for Time Series Forecasting?* www.aaai.org

Zhang, Y., & Yan, J. (2023). *CROSSFORMER: TRANSFORMER UTILIZING CROSS-DIMENSION DEPENDENCY FOR MULTIVARIATE TIME SERIES FORECASTING*.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2020). *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*. http://arxiv.org/abs/2012.07436

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). *FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting*. http://arxiv.org/abs/2201.12740

Appendix

APPENDIX A.1 – Results for Univariate solutions for each prediction length.

	Model	**Informer**		**Autoformer**		**FEDFormer**		**Non-stationary Transformer**	
	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	24	**0.098**	**0.247**	--	--	--	--	--	--
	48	**0.158**	**0.319**	--	--	--	--	--	--
	168	**0.183**	**0.346**	--	--	--	--	--	--
	336	**0.222**	**0.387**	--	--	--	--	--	--
	720	**0.269**	**0.435**	--	--	--	--	--	--
ETTh2	24	**0.093**	**0.240**	--	--	--	--	--	--
	48	**0.155**	**0.314**	--	--	--	--	--	--
	168	**0.232**	**0.389**	--	--	--	--	--	--
	192	--	--	--	--	--	--	--	--
	336	**0.263**	**0.417**	--	--	--	--	--	--
	720	**0.277**	**0.431**	--	--	--	--	--	--
ETTm1	24	**0.030**	**0.137**	--	--	--	--	--	--
	48	**0.069**	**0.203**	--	--	--	--	--	--
	96	**0.194**	**0.372**	--	--	--	--	--	--
	288	**0.401**	**0.554**	--	--	--	--	--	--
	672	**0.512**	**0.644**	--	--	--	--	--	--
ETTm2	96	--	--	0.065	**0.189**	**0.063**	**0.189**	0.069	0.193
	192	--	--	0.118	0.256	**0.102**	**0.245**	0.109	0.249
	336	--	--	0.154	0.305	**0.130**	**0.279**	0.139	0.286
	720	--	--	0.182	0.335	**0.178**	**0.325**	0.180	0.331
Weather	24	**0.117**	**0.251**	--	--	--	--	--	--
	48	**0.178**	**0.318**	--	--	--	--	--	--
	96	--	--	--	--	**0.0035**	**0.046**	--	--
	168	**0.266**	**0.398**	--	--	--	--	--	--
	192	--	--	--	--	**0.0054**	**0.059**	--	--
	336	0.297	0.416	--	--	**0.0041**	**0.050**	--	--
	720	0.359	0.466	--	--	**0.0055**	**0.059**	--	--
ECL	48	**0.239**	**0.359**	--	--	--	--	--	--
	168	**0.447**	**0.503**	--	--	--	--	--	--
	336	**0.489**	**0.528**	--	--	--	--	--	--
	720	**0.540**	**0.571**	--	--	--	--	--	--
	960	**0.582**	**0.608**	--	--	--	--	--	--
ILI	24	--	--	--	--	**0.693**	**0.629**	--	--
	36	--	--	--	--	**0.554**	**0.604**	--	--
	48	--	--	--	--	**0.699**	**0.696**	--	--
	60	--	--	--	--	**0.828**	**0.770**	--	--
	96	--	--	--	--	**0.170**	**0.263**	--	--
	168	--	--	--	--	**0.173**	**0.265**	--	--
	336	--	--	--	--	**0.178**	**0.266**	--	--
	720	--	--	--	--	**0.187**	**0.286**	--	--
Exchange	96	--	--	0.241	0.387	0.131	0.284	**0.104**	**0.235**
	192	--	--	0.273	0.403	0.277	0.420	**0.230**	**0.375**
	336	--	--	0.508	0.539	**0.426**	0.511	0.432	**0.509**
	720	--	--	0.991	0.768	1.162	0.832	**0.782**	**0.682**
Electricity	96	--	--	--	--	**0.253**	**0.370**	--	--
	192	--	--	--	--	**0.282**	**0.386**	--	--
	336	--	--	--	--	**0.346**	**0.431**	--	--

	720	--	--	--	--	**0.422**	**0.484**	--	--

APPENDIX A.2 – Results for LTSF-Linear model.

	Model	**LTSF-Linear (Multivariate)**	
	Metric	MSE	MAE
ETTh1	96	0.374	0.394
	192	0.405	0.415
	336	0.429	0.427
	720	0.440	0.453
ETTh2	96	0.277	0.338
	192	0.344	0.381
	336	0.357	0.400
	720	0.394	0.436
ETTm1	96	0.299	0.343
	192	0.335	0.365
	336	0.369	0.386
	720	0.425	0.421
ETTm2	96	0.167	0.255
	192	0.221	0.293
	336	0.274	0.327
	720	0.368	0.384
Weather	96	0.176	0.232
	192	0.218	0.269
	336	0.262	0.301
	720	0.362	0.348
ILI	24	1.683	0.858
	36	1.703	0.858
	48	1.719	0.884
	60	1.819	0.917
Traffic	96	0.310	0.279
	192	0.423	0.284
	336	0.435	0.290
	720	0.464	0.307
Exchange	96	0.081	0.203
	192	0.157	0.293
	336	0.305	0.414
	720	0.643	0.601
Electricit	96	0.140	0.237
	192	0.153	0.249
	336	0.265	0.169

	720	0.297	0.203

APPENDIX A.3 – Results for Multivariate solutions (ETT dataset) for each prediction length.

	Model	**Informer**		**Autoformer**		**FEDFormer**		**Pyraformer**		**Crossformer**		**PatchTST**		**CARD**		**Non-stationary Transformer**	
	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	24	0.577	0.549	0.384	0.425	--	--	--	--	**0.305**	**0.367**	--	--	--	--	--	--
	48	0.685	0.625	0.392	0.419	--	--	--	--	**0.352**	**0.394**	--	--	--	--	--	--
	96	--	--	--	--	--	--	--	--	--	--	**0.370**	0.399	0.383	**0.391**	--	--
	168	0.931	0.752	0.490	0.481	--	--	0.808	0.683	**0.410**	**0.441**	--	--	--	--	--	--
	192	--	--	--	--	--	--	0.945	0.766	--	--	**0.413**	0.421	0.435	**0.420**	--	--
	336	1.128	0.873	0.505	0.484	--	--	--	--	0.440	0.461	**0.422**	**0.436**	0.479	0.442	--	--
	720	1.215	0.896	0.498	0.500	--	--	1.022	0.806	0.519	0.524	**0.447**	0.466	0.471	**0.461**	--	--
ETTh2	24	0.720	0.665	**0.261**	**0.341**	--	--	--	--	--	--	--	--	--	--	--	--
	48	1.457	1.001	**0.312**	**0.373**	--	--	--	--	--	--	--	--	--	--	--	--
	96	--	--		--	--	--	--	--	--	--	**0.274**	0.399	0.281	**0.330**	--	--
	168	3.489	1.515	**0.457**	**0.455**	--	--	--	--	--	--	--	--	--	--	--	--
	192	--	--	--	--	--	--	--	--	--	--	**0.339**	**0.379**	0.363	0.381	--	--
	336	2.723	1.340	0.471	0.475	--	--	--	--	--	--	**0.329**	**0.380**	0.411	0.418	--	--
	720	3.467	1.340	0.474	0.484	--	--	--	--	--	--	**0.379**	**0.422**	0.416	0.431	--	--
ETTm1	24	--	--	0.383	0.403	--	--	--	--	**0.211**	**0.293**	--	--	--	--	--	--
	48	--	--	0.454	0.453	--	--	--	--	**0.300**	**0.352**	--	--	--	--	--	--
	96	--	--	**0.255**	**0.339**	--	--	0.480	0.486	--	--	0.290	0.342	0.316	0.347	--	--
	168	--	--	--	--	--	--	--	--	**0.320**	**0.373**	--	--	--	--	--	--
	192	--	--	--	--	--	--	--	--	--	--	**0.332**	**0.369**	0.363	0.370	--	--
	288	--	--	**0.342**	**0.378**	--	--	0.754	0.659	0.404	0.427	--	--	--	--	--	--
	336	--	--	--	--	--	--	--	--	--	--	**0.366**	0.392	0.392	**0.390**	--	--
	672	--	--	0.434	0.430	--	--	0.857	0.707	0.569	0.528	--	--	--	--	--	--
	720	--	--	--	--	--	--	--	--	--	--	**0.416**	**0.420**	0.458	0.425	--	--
ETTm2	24	--	--	**0.153**	**0.261**	--	--	--	--	--	--	--	--	--	--	--	--
	48	--	--	**0.178**	**0.280**	--	--	--	--	--	--	--	--	--	--	--	--
	96	--	--	0.255	0.339	0.203	0.287	--	--	--	--	**0.165**	0.255	0.169	**0.248**	0.192	0.274
	192	--	--	0.281	0.340	0.269	0.328	--	--	--	--	**0.220**	**0.292**	0.234	**0.292**	0.280	0.330
	288	--	--	**0.342**	**0.378**	--	--	--	--	--	--	--	--	--	--	--	--
	336	--	--	0.339	0.372	0.325	0.366	--	--	--	--	**0.274**	**0.329**	0.294	0.339	0.334	0.361
	672	--	--	**0.434**	**0.430**	--	--	--	--	--	--	--	--	--	--	--	--
	720	--	--	0.422	0.419	0.421	0.415	--	--	--	--	**0.362**	**0.385**	0.390	0.388	0.417	0.413

APPENDIX A.4 – Results for Multivariate solutions (Weather, ECL, and ILI datasets) for each prediction length.

	Model	**Informer**		**Autoformer**		**FEDFormer**		**Pyraformer**		**Crossformer**		**PatchTST**		**CARD**		**Non-stationary Transformer**	
	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Weather	24	--	--	--	--	--	--	--	--	**0.294**	**0.343**	--	--	--	--	--	--
	48	--	--	--	--	--	--	--	--	**0.370**	**0.411**	--	--	--	--	--	--
	96	--	--	0.266	0.336	0.217	0.296	--	--	--	--	**0.149**	0.198	0.150	**0.188**	0.173	0.223
	168	--	--	--	--	--	--	--	--	**0.473**	**0.494**	--	--	--	--	--	--
	192	--	--	0.307	0.367	0.276	0.336	--	--	--	--	**0.194**	0.241	0.202	**0.238**	0.245	0.285
	336	--	--	0.359	0.395	0.339	0.380	--	--	0.495	0.515	**0.245**	**0.282**	0.260	**0.282**	0.321	0.338
	720	--	--	0.419	0.428	0.403	0.428	--	--	0.526	0.542	**0.314**	**0.334**	0.343	0.353	0.414	0.410
ECL	48	--	--	--	--	--	--	--	--	**0.156**	**0.255**	--	--	--	--	--	--
	168	--	--	--	--	--	--	--	--	**0.231**	**0.309**	--	--	--	--	--	--
	336	--	--	--	--	--	--	--	--	**0.323**	**0.369**	--	--	--	--	--	--
	720	--	--	--	--	--	--	--	--	**0.404**	**0.423**	--	--	--	--	--	--
	960	--	--	--	--	--	--	--	--	**0.433**	**0.438**	--	--	--	--	--	--
ILI	24	--	--	3.483	1.287	2.203	0.963	--	--	3.041	1.186	**1.1319**	**0.754**	--	--	2.294	0.945
	36	--	--	3.103	1.148	2.272	0.976	--	--	3.406	1.232	**1.439**	**0.834**	--	--	1.825	0.848
	48	--	--	2.669	1.085	2.209	0.981	--	--	3.459	1.221	**1.553**	**0.815**	--	--	2.010	0.900
	60	--	--	2.770	1.125	2.545	1.061	--	--	3.640	1.305	**1.470**	**0.788**	--	--	2.178	0.963

APPENDIX A.5 – Results for Multivariate solutions (Traffic, Exchange, and Electricity datasets) for each prediction length.

	Model	**Informer**		**Autoformer**		**FEDFormer**		**Pyraformer**		**Crossformer**		**PatchTST**		**CARD**		**Non-stationary Transformer**	
	Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Traffic	24	--	--	--	--	--	--	--	--	**0.491**	**0.274**	--	--	--	--	--	--
	48	--	--	--	--	--	--	--	--	**0.519**	**0.295**	--	--	--	--	--	--
	96	--	--	0.613	0.388	0.562	0.349	--	--	--	--	**0.360**	**0.249**	0.419	0.269	0.612	0.338
	168	--	--	--	--	0.562	0.346	--	--	**0.513**	**0.289**	--	--	--	--	--	--
	192	--	--	0.616	0.382	--	--	--	--	--	--	**0.379**	**0.256**	0.443	0.276	0.613	0.340
	336	--	--	0.622	0.337	0.570	0.323	--	--	0.530	0.300	**0.392**	**0.264**	0.460	0.283	0.618	0.328
	720	--	--	0.660	0.408	0.596	0.368	--	--	0.573	0.313	**0.432**	**0.286**	0.490	0.299	0.653	0.355
Exchange	96	--	--	0.197	0.323	0.139	0.276	--	--	--	--	--	--	--	--	**0.111**	**0.237**
	192	--	--	0.300	0.369	0.256	0.369	--	--	--	--	--	--	--	--	**0.219**	**0.335**
	336	--	--	0.509	0.524	0.426	**0.464**	--	--	--	--	--	--	--	--	**0.421**	0.476
	720	--	--	1.447	0.941	**1.090**	0.800	--	--	--	--	--	--	--	--	1.092	**0.769**
Electricity	96	--	--	0.201	0.317	0.183	0.297	--	--	--	--	**0.129**	**0.222**	0.141	0.233	0.169	0.273
	168	--	--	--	--	--	--	**0.719**	**0.256**	--	--	--	--	--	--	--	--
	192	--	--	0.222	0.334	0.195	0.308	--	--	--	--	**0.147**	**0.240**	0.160	0.250	0.182	0.286
	336	--	--	0.231	0.338	0.212	0.313	1.533	0.291	--	--	**0.163**	**0.259**	0.173	0.263	0.200	0.304
	720	--	--	0.254	0.361	0.231	0.343	4.312	0.346	--	--	**0.197**	0.290	**0.197**	**0.284**	0.222	0.321

APPENDIX A.6 – Linear *versus* Transformer-based results.

		LTSF-Linear (Multivariate)		**PatchTST**	
		MSE	MAE	MSE	MAE
ETTh1	96	**0.374**	**0.394**	**0.370**	0.399
	192	0.405	**0.415**	0.413	0.421
	336	0.429	**0.427**	**0.422**	0.436
	720	**0.440**	**0.453**	0.447	0.466
ETTh2	96	0.277	**0.338**	**0.274**	0.399
	192	0.344	0.381	**0.339**	**0.379**
	336	0.357	0.400	**0.329**	**0.380**
	720	0.394	0.436	**0.379**	**0.422**
ETTm1	96	0.299	0.343	**0.290**	**0.342**
	192	0.335	**0.365**	**0.332**	0.369
	336	0.369	**0.386**	**0.366**	0.392
	720	0.425	0.421	**0.416**	**0.420**
ETTm2	96	0.167	0.255	**0.165**	**0.255**
	192	0.221	0.293	**0.220**	**0.292**
	336	**0.274**	**0.327**	**0.274**	0.329
	720	0.368	**0.384**	**0.362**	0.385
Weather	96	0.176	0.232	**0.149**	**0.198**
	192	0.218	0.269	**0.194**	**0.241**
	336	0.262	0.301	**0.245**	**0.282**
	720	0.362	0.348	**0.314**	**0.334**
ILI	24	1.683	0.858	**1.1319**	**0.754**
	36	1.703	0.858	**1.439**	**0.834**
	48	1.719	0.884	**1.553**	**0.815**
	60	1.819	0.917	**1.470**	**0.788**
Traffic	96	**0.310**	0.279	0.360	**0.249**
	192	0.423	0.284	**0.379**	**0.256**
	336	0.435	0.290	**0.392**	**0.264**
	720	0.464	0.307	**0.432**	**0.286**
Exchange	96	**0.081**	**0.203**	--	--
	192	**0.157**	**0.293**	--	--
	336	**0.305**	**0.414**	--	--
	720	**0.643**	**0.601**	--	--
Electricity	96	0.140	0.237	**0.129**	**0.222**
	192	0.153	0.249	**0.147**	**0.240**
	336	0.265	0.169	**0.163**	**0.259**
	720	0.297	**0.203**	**0.197**	0.290

APPENDIX A.7 – Weather dataset plot with outliers.