Machine Learning Techniques and Large Language Models for Property Rental Price Prediction in Ireland

Madina Sagatova

A Thesis Submitted in Partial Fulfilment

of the requirements for the

Degree of

Master of Science in Data Analytics

Diagram

Description automatically generated with medium confidence

May 2025

Supervisor: David McQuaid

**CCT College Dublin**

**Assessment Cover Page**

| **Module Title:** | Capstone Project |
| --- | --- |
| **Assessment Title:** | Machine Learning Techniques and Large Language Models for Property Rental Price Prediction in Ireland |
| **Supervisor Name:** | David McQuaid |
| **Student Full Name:** | Madina Sagatova |
| **Student Number:** | 2021255 |
| **Assessment Due Date:** | 16/05/2025 |
| **Date of Submission:** | 16/05/2025 |

**Declaration**

| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |
| --- |

# Acknowledgement

Words cannot express my sincere gratitude to my supervisor, David McQuaid, for his continuous encouragement, invaluable feedback and guidance throughout this capstone project.

I also would like to express my deepest gratitude to the faculty and staff at the CCT College Dublin for providing a supportive academic environment and the resources necessary to complete this master's course.

Lastly, I would be remiss in not mentioning my family, especially my mom, my sister and boyfriend. Their love and belief in me had motivated me during this process.

[**Acknowledgement 2**](#_nfvxogo52te2)

[**List of Figures 5**](#_ll5pmfkhcvol)

[**List of Tables 6**](#_74spj0n90h5r)

[**List of Acronyms 7**](#_4gu6elpwe5vd)

[**Abstract 1**](#)

[**1. INTRODUCTION 2**](#_g6gmjlxnosx1)

[1.1 Background and Context 2](#_oguznct9xrv5)

[1.2 Research Problem and Gap 2](#_htjwacdqvijd)

[1.3 Research Question 3](#_z9gwy0m1n7cw)

[1.4 Research Objectives 3](#_p3tm71lpcqqd)

[1.5 Report Overview 3](#_63od71taeu9t)

[**2. LITERATURE REVIEW 4**](#_v3xck4xrmryu)

[2.1 Traditional and Machine Learning Approaches to Price Prediction 4](#_xzx6st90krcp)

[2.2 Time Series Forecasting Models 6](#_8ywdu1n4mb5g)

[2.3 LLMs in Time Series and Real Estate Forecasting 7](#_218go8l39vfp)

[2.3.1 LLM Applications in Real Estate Forecasting 7](#_1e1nsk60y71e)

[2.3.2 LLM Applications in Time Series Forecasting 7](#_witfkqx00l3p)

[2.3.3 Few-Shot Learning and Prompting Techniques 8](#_i2sy2o9m533z)

[2.3.4 Limitations and Critical Reflections 9](#_as2hsxx183g8)

[**3. METHODOLOGY 10**](#_yyqnehfltcmg)

[3.1 Primary Research Methodology 10](#_tebrxgod4j74)

[3.2 Research Design/Framework 10](#_lkrm9eio3lw6)

[3.2.1 Data Acquisition Stage 11](#_m7yhskw7vlq0)

[3.2.2 Data Exploratory and Preprocessing Stage 11](#_wy0lm6qkxg2y)

[3.2.3 Model Selection Stage 11](#_17wh7z3mjfee)

[3.2.4 Models Evaluation Stage 11](#_zez3zmjm44pz)

[3.3 Ethical and Licensing Considerations 11](#_2xrj42hqfoey)

[3.3.1 Data Privacy and Anonymity 11](#_ct2ybvmboyat)

[3.3.2 Fairness and Bias 12](#_8iropj60hbfk)

[3.3.3 Licensing and Intellectual Property 12](#_1sq4qzrzu1qs)

[3.4 Data Collection 13](#_yhu8muflsm9j)

[3.5 Data Preprocessing 13](#_gi6mbmkdtf3z)

[3.5.1 Handling Missing Data 13](#_ary73p5t63oe)

[3.5.2 Location Filtering 14](#_ruxloaitd3ny)

[3.5.3 Time Formatting 14](#_rpv4ttd8nrin)

[3.5.4 Outlier Detection 14](#_ey1ws182umkb)

[3.5.5 Stationarity Test 15](#_w4qac6pcw4la)

[3.5.6 Time Series Decomposition 16](#_t8s1qtqje9mc)

[3.5.7 Autocorrelation Analysis 17](#_e6fklqqe8skt)

[3.6 Models Selection 19](#_uyx1oaom40y0)

[3.6.1 SARIMA 19](#_ebjmcgf4bx6o)

[3.7.2 Prophet 20](#_z2fga5ocn7w3)

[3.6.3 LLMs: zero and few-shot prediction using ChatGPT 21](#_mllihayqtvz)

[3.6.4 Time-LLM 23](#_1hkoe8vpbsbh)

[3.6.5. Amazon Chronos 25](#_g5hb5ecpq4kk)

[3.7 Performance Metrics 26](#_4ljy0jerizhc)

[3.7.1. Mean Squared Error (MSE) 26](#_eov7kda5qxbe)

[3.7.2. Mean Absolute Error (MAE) 26](#_9bvoyqq71f2p)

[3.7.3. R² Score 26](#_77z2tkfcxbcy)

[3.7.4. MAPE 27](#_3q6beuj4dua9)

[3.8 Hyperparameter Tuning 27](#_k3nuxvk35g1c)

[3.8.1 SARIMA Hyperparameter Tuning 27](#_wkbgp0ehreo4)

[3.8.2 Prophet Hyperparameter Tuning 27](#_41ao9xcmrzt6)

[3.8.3 Zero and Few-Shot Hyperparameter Tuning 28](#_zhj32z947jrm)

[3.8.4 Time-LLM Hyperparameter Tuning 28](#_rx2gillerykz)

[3.8.5 Chronos Hyperparameter Tuning 29](#_1nhx3tcrxh10)

[**4. RESULTS 29**](#_r20nbmzi3vj3)

[4.1 SARIMA – Evaluation Results 29](#_zb6t956x60x3)

[4.2 Prophet – Evaluation Results 31](#_sq0o8obti3yu)

[4.3 LLMs: zero and few-shot prediction – Evaluation Results 33](#_4r353o6zpqla)

[4.3.1 ChatGPT-3.5 33](#_5bb5u6s35g9n)

[4.3.2 ChatGPT-4-turbo 36](#_htyyk5si2ngm)

[4.3.3 Time-LLM(GPT-2) – Evaluation Results 38](#_iyvr82losq5r)

[4.3.4 Chronos – Evaluation Results 40](#_so2shjldv56i)

[4.3.5 Evaluation Results and Model Comparison 45](#_crgob1p7dpnv)

[**5. CONCLUSION 46**](#_fs9axs7eooim)

[5.1 Summary of Key Findings 46](#_cutvao9af2bs)

[5.2 Limitations of the Study 47](#_gxlrnldfiqln)

[5.2.1 Experimental Environment 47](#_luqea13fse26)

[5.2.2 Probabilistic Challenges 47](#)

[5.3 Recommendations for Future Work 48](#_snvd9wrz4npy)

[**References 49**](#_b05xt8g5vtu3)

[**Appendices 56**](#)

[**Appendix A: Summary of Previous Research on Real Estate Price Forecasting in Ireland 56**](#)

[**Appendix B: Capstone Project Timeline 58**](#)

[**Appendix C: : Forecasting Methodology Workflow Diagram 59**](#_5f2q477qcm9z)

| List of Figures |  |
| --- | --- |
| Figure 1: Flowchart of the proposed methodology. (Source: Author's analysis) ………. | 11 |
| Figure 2: Boxplot of Average Rent (€) in Dublin. (Source: Author's analysis) ................ | 15 |
| Figure 3: Additive and multiplicative decomposition of quarterly average rent prices in Dublin from 2007 to 2024. (Source: Author's analysis) ………………………………....... | 16 |
| Figure 4: ACF and PACF of first-differenced series in Dublin rent time series. (Source: Author's analysis) ………………………………………………………………….................. | 17 |
| Figure 5: ACF and PACF of seasonally differenced series in Dublin rent time series. (Source: Author's analysis) …………………………………………………………………... | 18 |
| Figure 6: SARIMA(1,1,0)(1,1,0)[4] Dublin rent prices forecasting (Source: Author's analysis) ………………………………………………………………………………………... | 29 |
| Figure 7: SARIMA(1,1,2)(0,1,1)[4] Dublin rent prices forecasting – Auto-Tuned (Source: Author's analysis) …………………………………………………………………... | 30 |
| Figure 8: Prophet Dublin rent prices forecasting with default settings (Source: Author's analysis) ……………………………………………………………………………................. | 31 |
| Figure 9: Prophet Dublin rent prices forecasting with tuned settings (Source: Author's analysis) ……………………………………………………………………………................. | 31 |
| Figure 10: 0- (A) and 1-shot (B) Dublin rent prices forecasting using ChatGPT-3.5 (Source: Author's analysis) …………………………………………………………………... | 33 |
| Figure 11: 5- (A) and 10-shot (B) Dublin rent prices forecasting using ChatGPT-3.5 (Source: Author's analysis) …………………………………………………………………... | 33 |
| Figure 12: 20- (A) and 30-shot (B) Dublin Rent Prices Forecasting using ChatGPT-3.5 (Source: Author's analysis) …………………………………………………......................... | 34 |
| Figure 13: 0- (A) and 1-shot (B) Dublin rent prices forecasting using ChatGPT-4-turbo (Source: Author's analysis) ……………………………………………................................ | 35 |
| Figure 14: 5- (A) and 10-shot (B) Dublin rent prices forecasting using ChatGPT-4-turbo (Source: Author's analysis) ……………………………………………....................... | 36 |
| Figure 15: 20- (A) and 30-shot (B) Dublin rent prices forecasting using ChatGPT-4-turbo (Source: Author's analysis) ……………………………………….............................. | 36 |
| Figure 16: Dublin rent prices forecasting using Time-LLM (Source: Author's analysis) . | 37 |
| Figure 17: Dublin rent prices forecasting using Time-LLM after tuning (Source: Author's analysis) …………………………………………………………………………....... | 38 |
| Figure 18: Dublin rent prices forecasting using Chronos (T5-small) (Source: Author's analysis) ………………………………………………………………………………………... | 40 |
| Figure 19: Dublin rent prices forecasting using Chronos (T5-base) (Source: Author's analysis) ………………………………………………………………………………………... | 42 |
| Figure 20: Dublin rent prices forecasting using Chronos (T5-large) (Source: Author's analysis) ………………………………………………………………………………………... | 43 |
| Figure 21: Additional experiment using Chronos (T5-small (A), T5-base (B), T5-large (C)) 0.9 split to predict Dublin rent prices (Source: Author's analysis) ………................. | 43 |
| Figure B1: Capstone Project Timeline and Progress Tracker (Source: Author's analysis) ………..................………..................………..................………..................…... | 57 |
| Figure C1: Forecasting Methodology Workflow Diagram (Source: Author's analysis) … | 58 |

| List of Tables |  |
| --- | --- |
| Table 1. Dataset Features and Descriptions ………………………………………………... | 13 |
| Table 2. Prompt Structure Example ………………………………………………………….. | 21 |
| Table 3. Time-LLM Prompt Structure Example………………………………………………. | 23 |
| Table 4. Evaluation Metrics: SARIMA(1,1,0)(1,1,0)[4] ……………………………………… | 28 |
| Table 5. Evaluation Metrics: SARIMA (1,1,2)(0,1,1)[4] ……………………………………... | 29 |
| Table 6. Evaluation Metrics: SARIMA Comparison Summar ………………………………. | 30 |
| Table 7. Evaluation Metrics: Prophet Forecast with default settings ……………………… | 31 |
| Table 8. Evaluation Metrics: Prophet Forecast with tuned settings ……………………….. | 32 |
| Table 9. Evaluation Metrics: Prophet Comparison Summary ……………………………… | 32 |
| Table 10. Evaluation Metrics: Zero and Few-shot Prediction using ChatGPT-3.5 ………. | 34 |
| Table 11. Evaluation Metrics: Zero and Few-shot Prediction using ChatGPT-4-turbo ….. | 37 |
| Table 12. Evaluation Metrics: Time-LLM (input size=16, batch size=16) ………………… | 38 |
| Table 13. Evaluation Metrics: Time-LLM (input size=8, batch size=16) after tuning ……. | 39 |
| Table 14. Evaluation Metrics: Time-LLM Comparison Summary ………………………….. | 39 |
| Table 15. Evaluation Metrics: Chronos (T5-small) ………………………………………….. | 40 |
| Table 16. Evaluation Metrics: Chronos (T5-base) …………………………………………... | 41 |
| Table 17. Evaluation Metrics: Chronos (T5-large) ………………………………………….. | 42 |
| Table 18. Evaluation Metrics: Chronos Comparison Summary …………………………… | 43 |
| Table 19. Evaluation Metrics: Chronos (0.9 split) Comparison Summary ………………... | 44 |
| Table 20. Evaluation Metrics: Final Forecasting Model Comparison ……………………... | 44 |
| Table 21. Hardware Specifications …………………………………………………………… | 46 |
| Table A1. Summary of Previous works on Real Estate Prices Forecasting in Ireland ….. | 55 |

# List of Acronyms

| ANN | Artificial Neural Networks |
| --- | --- |
| ADF | Augmented Dickey-Fuller |
| ACF | Autocorrelation Function |
| AR | Autoregressive |
| ARIMA | Autoregressive Integrated Moving Average |
| ARMA | Autoregressive Moving Average |
| Autoformer | Autoregressive Transformer for Long-Term Forecasting |
| BPNN | Back Propagation Neural Networks |
| BERT | Bidirectional Encoder Representations from Transformers |
| CSO | Central Statistics Office |
| COT | Chain of Thought |
| R² | Coefficient of Determination |
| CNN | Convolutional Neural Network |
| DLinear | Decomposition-based Linear Model for Time Series Forecasting |
| ESTformer | Enhanced Spatial-Temporal Transformer |
| EDA | Exploratory Data Analysis |
| XGBoost | Extreme Gradient Boosting |
| GDPR | General Data Protection Regulation |
| GARCH | Generalized Autoregressive Conditional Heteroskedasticity |
| GPT | Generative Pre-trained Transformer |
| GPT4TS | Generative Pre-trained Transformer for Time Series |
| InstructGPT | Instruction-Tuned Generative Pre-Trained Transformer |
| IQR | Interquartile Range |
| LLM | Large Language Model |
| LLaMA | Large Language Model Meta AI |
| LightGBM | Light Gradient Boosting Machine |
| LSTM | Long Short-Term Memory |
| ML | Machine Learning |
| MAE | Mean Absolute Error |
| MAPE | Mean Absolute Percentage Error |
| MSE | Mean Squared Error |
| MA | Moving Average |
| NER | Named Entity Recognition |
| NLP | Natural Language Processing |
| PACF | Partial Autocorrelation Function |
| PatchTST | Patch-based Time Series Transformer |
| RMSE | Root Mean Squared Error |
| SARIMA | Seasonal Autoregressive Integrated Moving Average |
| SVM | Support Vector Machine |
| SVR | Support Vector Regression |
| T5 | Text-to-Text Transfer Transformer |
| Time-LLM | Time Series Forecasting with Large Language Models |
| VAR | Vector Autoregression |
| Zero-shot-CoT | Zero-Shot Chain of Thought |

# 

# Abstract

This study investigates the application and effectiveness of Large Language Models (LLMs) for time series predictions, focusing on Dublin rent prices. This research explores the potential of LLM-based models to improve the forecasting accuracy compared to the traditional SARIMA model and the modern Prophet model developed by Facebook. The LLM-based methods include zero- and few-shot predictions using the ChatGPT-3.5 and ChatGPT-4-turbo, the reprogramming framework Time-LLM and the pre-trained model based on LLM architecture – Chronos model. The result demonstrates that although few-shot LLMs had the lowest point-wise error metrics, they struggle to capture trend and seasonality patterns as effectively as the SARIMA model. The Chronos model is sensitive to random seed selection, but it shows promise in short-term predictions. Nevertheless, this study provides valuable insights into the feasibility of utilising LLMs for complex time series prediction tasks.

# 1. INTRODUCTION

## 1.1 Background and Context

Forecasting has always captivated people and has come a long way since ancient times, when early humans observed the sky to gauge weather conditions for hunting (Hyndman and Athanasopoulos, 2018). In the present time, forecasting is driven by data and Machine Learning (ML) algorithms, informing decisions across a wide range of industries, including the real estate market.

A wide range of techniques and models available for building predictions, and time series forecasting is one of the most widely used approaches. Time series is defined as a set of quantitative observations recorded at multiple points in time (Kirchgässner, Wolters and Hassler, 2012). In this capstone research forecasting future rent prices in Ireland is of interest.

Time series data is following special characteristics including trend, seasonality, cyclical movement and unexpected variation. The trend characterises the long run upward or downward movement in the time series data. Seasonality manifests as regular and repetitive fluctuations in time series. Cyclical changes are repeating movements that are observed over a longer period and occur less frequently than seasonal fluctuations. Unexpected variation, also called noise, refers to random, stochastic movements in the data. Understanding these internal structures supports the development of efficient forecasting models (Pal and Prakash, 2017).

This research applies both classical and modern forecasting approaches. A traditional time series model, Seasonal Autoregressive Integrated Moving Average (SARIMA), is utilised alongside Prophet, a more recent machine learning-based forecasting tool developed by Facebook. In addition to SARIMA and Prophet, we employed an LLM-based approach. Specifically, we explored the utilisation of Chat Generative Pre-trained Transformer (ChatGPT) (OpenAI, 2024), state-of-art LLM, in various few-shot scenarios (0-shot, 1-shot, 5-shot, 10-shot and 15-shot) (Chen and Si, 2024) to evaluate its performance in forecasting rental prices and recent effective frameworks such as Time Series Forecasting with Large Language Models (Time-LLM) (Jin et al., 2024) reprogramming framework and Chronos (Ansari et al., 2024), that forecasting differently by treating temporal data as tokenised sequences.

## 1.2 Research Problem and Gap

ML algorithms have been widely studied and applied to rental or housing price prediction. In the present time, there is increasing interest in researching and developing "foundational models" for time series forecasting as a result of the emergence of LLMs and their zero-shot and few-shot learning capabilities. When it comes to LLMs, three main methods have been investigated: directly prompting pretrained LLMs in natural language (Gruver et al., 2023; Xue and Salim, 2023), fine-tuning LLMs for time series tasks (Zhou et al., 2023a; Jin et al., 2023) and a recent third approach, which involves pre-training LLMs directly on tokenised time series data (Ansari et al., 2024).

Despite these promising achievements, the application of LLM-based forecasting techniques in real estate markets, particularly for predicting rental prices in Ireland, is still mainly unexplored. Most existing studies are focused on financial time series forecasting (Wang et al., 2024; Zhang et al., 2024) or general benchmarking datasets such as traffic (de Zarzà et al., 2023) or bike rental forecasting (Baron and Karpinski, 2025), leaving a gap in understanding how LLMs perform in domain-specific environments like the Irish housing rental sector.

## 1.3 Research Question

This capstone project aims to answer the following research question:

How effective are LLM-based methods compared to ML models in forecasting rental prices in the Irish property market?

## 1.4 Research Objectives

*Primary Objective:*

The primary objective of this capstone project is to evaluate and compare the forecasting performance of ML techniques and LLM-based approaches in predicting rental prices in the Irish real estate market.

*Secondary Objectives:*

To achieve the primary objective, the following secondary objectives must be fulfilled:

* To evaluate the effectiveness of ML-based forecasting models in predicting rental prices based on property features.
* To assess the effectiveness of LLMs to improve prediction accuracy for rental price forecasting in Ireland.
* To provide an analysis comparing forecasting ML models and LLM-based techniques that explores how each method approaches the prediction task, the quality of insights they provide, and their suitability for various aspects of the Irish rental market.

## 1.5 Report Overview

The remainder of the report is structured as follows. In the Literature Review chapter, an overview of existing scientific studies on rental price prediction, including traditional ML algorithms, time series forecasting methods, and the application of LLMs for forecasting tasks, is provided. In the Methodology chapter, the research framework for predicting rental prices in Ireland is outlined, including the selection of machine learning techniques and LLMs, the data collection process, and the ethical considerations. The Evaluation chapter outlines the steps taken to evaluate and analyse the forecasting results of the selected models. The experimental findings and the comparative analysis of the forecasting performances are described in the Results chapter. And lastly, the Conclusion chapter summarises the key findings, discusses the limitations of the current study, and offers recommendations for the future research.

# 2. LITERATURE REVIEW

## 2.1 Traditional and Machine Learning Approaches to Price Prediction

This section covers ML models being applied to predict housing and rental prices. There were various technologies used to forecast house and rental prices including time series (Milunovich, 2020; Lim et al., 2016), deep learning (Wang et al., 2019; Jiang and Shen, 2019; Wang et al., 2021), traditional ML (Rawool et al., 2021; Adetunji et al., 2022) and hybrid methods (Pinter, Mosavi and Felde, 2020; Özöğür Akyüz et al., 2023; Zhan et al., 2023).

The Hedonic Price Theory, introduced by Rosen (1974), laid the foundation for Hedonic Regression Models. These models play an important role in studying real estate prices. Their widespread use in analysing the effect of various factors on house prices make them a robust tool for market segmentation. However, they have some limitations, in particular in capturing the complex non-linearity and model specification (Wang et al., 2014; Abdellah Sellam et al., 2024).

At first, Linear Regression was one of the traditional ML algorithms that captured attention, offering computational efficiency and ease of understanding (Cook, 1977). However, to address Linear Regression’s limitations such as capturing the high-dimensional and non-linear complexities, researchers investigated regularisations approaches like Lasso and Ridge Regressions. For instance, the study by Manjula et al (2017) explored different regression algorithms to achieve high accuracy in housing prices forecasting. They concluded that the simple Linear Regression model tended to underfit, exhibiting high bias, while the highly complex model tended to overfit, leading to high variance. The researchers also highlighted that overfitting can be reduced by implementing Ridge and Lasso Regressions.

Support Vector Regression (SVR) was proposed by Vapnik et al. (1997), offering an optimal solution to handle non-linearity in data. A comparative study by Li et al. (2009) was performed using SVR and Back Propagation Neural Networks (BPNN) to predict real estate price in Taiwan efficiently. The SVR model with trial-and-error method demonstrated the best result with MAPE (Mean Absolute Percentage Error) and Coefficient of Determination (R²) values.

The Decision Tree models were mostly being used as a problem identifier in forecasting prices in real estate modeling for decision making. The study by Zhang (2021) proposed an objective housing prediction scheme built on Decision Tree. The author chose five important features and applied grid search for optimisation. The results indicated that the Decision Tree scheme outperformed other ML models, and the dominant feature affecting housing prices was the number of houses. However, the Decision Tree model can easily suffer from overfitting (Bramer, 2007).

In response to Decision Tree limitation, techniques like Random Forest, which combines multiple models, were developed to improve generalisation (Ho, 1995). Wang and Wu (2018) in the “A new ML approach to house price estimation” used Random Forest for estimating house prices and compared its accuracy with Linear Regression. They used data from single-family house assessments in Arlington County, Virginia with 27,649 houses. The study demonstrated that Random Forest outperformed Linear Regression based on R² and RMSE (Root Mean Squared Error) values, especially when the location related features were included.

In contrast to Random Forest, Extreme Gradient Boosting (XGBoost) algorithm builds trees sequentially, each one correcting the mistakes of the previous tree. This approach allows XGBoost to handle various data structures, improving prediction accuracy (Chen and Guestrin, 2016). In order to assess and predict rental prices, Ming et al. (2021) examined 33,224 pieces of property rental data from Chengdu, China. They began the analysis by selecting key features such as area, transportation access, rent collection mode, etc. Using data visualisation techniques, they showed that the Chengdu's tenants were more interested in small-sized apartments and joint rentals. In this study, XGBoost outperformed Random Forest Regressor and Light Gradient Boosting Machine (LightGBM) in terms of prediction accuracy, achieving 0.85 after parameter tuning.

Deep learning techniques have gained attention in recent years for their ability to model complex and non-linear relationships. They are usually applicable for large datasets and to predict property prices, it is important to consider both the dependent and independent variables involved in the model (Chiarazzo et al. 2014). A study on this issue by Lam, Yu and Lam (2008) found that Artificial Neural Networks (ANN), particularly the 4-layer network, achieved the best performance in predicting house prices in Hong Kong with MAE = 1.95. The potential of Convolutional Neural Network (CNN) in forecasting house prices in Taiwan was documented by Zhan et al. (2020). The data for the research was conducted from 2013 to 2018 and included two types of housing transactions: “land + building” and “land + building + park”. The experimental comparison of two deep learning models indicated that CNN outperformed BPNN.

## 2.2 Time Series Forecasting Models

Time series forecasting is a fundamental technique for predicting future values that depend on time. The application of time series is broad, including finance (Tsay, 2010), energy (Deb et al., 2017), healthcare (Kaushik et al., 2020), and real estate (Gupta and Miller, 2009). Traditional statistical forecasting methods including Autoregressive (AR) and Autoregressive Moving Average (ARMA) models, were initially introduced by Whittle (1951). These foundational models were later extended to account for non-stationary data, resulting in the Autoregressive Integrated Moving Average (ARIMA) model, widely popularised by Box and Jenkins (1976). SARIMA, one of the most popular ARIMA variations, uses seasonal autoregressive, Moving Average (MA), integrated and seasonal differencing components to manage cyclical fluctuations.

In a comparative study, Crawford and Fratantoni (2003) assessed the forecasting performance of three time series forecasting models such as Markov regime-switching, ARIMA, and Generalized Autoregressive Conditional Heteroskedasticity (GARCH). They utilised state-level repeat sales home price indices from five U.S. states (in particular, California, Florida, Massachusetts, Ohio, and Texas). The authors revealed that the Markov regime-switching mode provided the best in-sample fit, which reflects its strength in modeling structural changes in housing markets. In contrast, when they evaluated out-of-sample forecast accuracy, the simpler ARIMA model generally outperformed both the regime-switching and GARCH models. A recent study conducted by Alburshaid and Al-Alawi (2024) applied SARIMA modeling to Bahrain's real estate market. Their findings demonstrated that the SARIMA effectively forecasted transaction volumes and handled seasonal variations in the real estate data, reinforcing its applicability to cyclical housing markets.

Facebook's Prophet model has demonstrated strong forecasting performance in a variety of domains. For instance, Garlapati et al. (2021) compared the ARIMA and Facebook Prophet models for forecasting stock prices. Their findings disclosed that although ARIMA provided accurate short-term forecasts, Prophet was a reliable option for time series prediction due to its ability to address missing data and analyse complex seasonal patterns. The study by Satrio et al. (2021) applied both ARIMA and Prophet models to forecast COVID-19 cases in Indonesia. The results of this research showed that the Prophet model outperformed ARIMA in capturing the pandemic's seasonality and complex trends, providing more accurate forecasts for proved cases, recoveries, and fatalities. In the sales industry, Jha and Pande (2021) applied Facebook's Prophet model to forecast supermarket sales. Their proposed research contrasted Prophet and ARIMA models, underlying that Prophet delivered better prediction accuracy, low error score and better model fit.

Kim, Kwon, and Choi's (2020) study investigated the impact of public rental costs on nearby housing prices in Busan, South Korea. They proposed Long Short-Term Memory (LSTM) techniques to construct a model for predicting prices based on the dataset with 547,740 apartment transactions. In this work the accuracy of the performance was conducted by comparing LSTM against one of the traditional time-series models called Vector Autoregression (VAR). The authors chose the VAR model because of its high-level performance in forecasting and its ability to cope with multiple variables. In regard to the results, the LSTM model performed more efficiently than traditional VAR in terms of RMSE.

## 2.3 LLMs in Time Series and Real Estate Forecasting

In this section, we discuss the application of LLM in predictive modelling. Applications beyond natural language processing, such as predictive tasks, have demonstrated the promise of LLMs such ChatGPT-4, ChatCPT-3, and Bidirectional Encoder Representations from Transformers (BERT) (Zarzà et al., 2023; Kim et al., 2024).

### 2.3.1 LLM Applications in Real Estate Forecasting

With regards to the real estate market, Heidari and Rafatirad (2020) introduced a novel method applying BERT to semantic CNNs for rent investment safety based on web information. The study demonstrated the capability of LLMs to effectively integrate and process various types of data sources.

Chen and Si (2024) examined rental price prediction in Shanghai using a dataset of 2,609 house transaction records in 2021. They utilised five ML models including Multiple Linear Regression, Ridge Regression, Lasso Regression, Decision Tree and Random Forest, along with LLM approach using ChatGPT. The LLM approach was evaluated in four scenarios: 0-shot, 1-shot, 5-shot and 10-shot. The 10-shot forecasting method indicated a result with the highest R² value (0.80). The authors came to the conclusion that ML approaches were effective and provided robust techniques for rental price forecasting. In addition, the ability of LLMs to manage unstructured data could increase predictive accuracy.

### 2.3.2 LLM Applications in Time Series Forecasting

Gruver et al. (2023) demonstrated that LLMs can serve as pretrained time series forecasters by encoding numerical values as text. They converted numerical sequences into textual input for token-based forecasting and find that LLMs, including Generative Pre-trained Transformer (GPT-3) and Large Language Model Meta AI (LLaMA-2) demonstrated zero-shot extrapolation capabilities that were comparable to or even exceed the performance of specialised time series models trained on downstream tasks. While results illustrated promise, the models may have difficulties with uncertainty calibration and tokenisation issues.

With a focus on NASDAQ-100 stock, Yu et al. (2023) investigated the LLMs’ implementation for forecasting financial time series. The experiment compared GPT-4 and Open LLaMA in zero-shot and few-shot settings with traditional approaches such as ARMA-GARCH and Gradient Boosting Trees. Findings showed that LLMs could surpass traditional statistical models and ML techniques. The performance of the LLM-based models was notably improved after applying the Chain of Thought (COT) prompting strategy.

By reprogramming LLMs, Jin et al. (2024) presented a novel framework for time series forecasting known as Time-LLM. To align the two modalities, they converted the input time series data into a text format before feeding it into the frozen LLM. To improve the input context and direct the transformation of reprogrammed input patches, the authors proposed Prompt-as-Prefix technique. Time-LLM was evaluated against a wide range of state-of-the- art models, such as Patch-based Time Series Transformer (PatchTST), Enhanced Spatial- Temporal Transformer (ESTformer), Autoregressive Transformer for Long-Term Forecasting (Autoformer), Generative Pre-trained Transformer for Time Series (GPT4TS) and Decomposition-based Linear Model for Time Series Forecasting (DLinear). The findings revealed that Time-LLM was a capable time series learner that outperformed various forecasting models, particularly in zero-shot and few-shot learning scenarios. The study emphasised that removing either the patch reprogramming or the Prompt-as- Prefix technique led to a notable decrease in performance.

Another novel framework for pretrained probabilistic time series models, called Chronos, was introduced by Ansari et al. (2024). The framework tokenised time series values using scaling and quantisation techniques into a fixed vocabulary and processed through a cross-entropy loss function. The authors built Chronos models on the T5 (Text-to-Text Transfer Transformer) architecture (ranging from 20M to 710M parameters) and trained on a diverse set of publicly available and synthetically generated datasets using Gaussian processes to reinforce generalisation. The comprehensive benchmark of 42 datasets showed Chronos outperformed traditional forecasting methods and achieved strong zero-shot performance on unseen data.

### 2.3.3 Few-Shot Learning and Prompting Techniques

The growing body of research on prompt engineering techniques has contributed to improving LLM performance. The study by Brown et al. (2020) introduced GPT-3 (175 billion parameters) and demonstrated its ability to perform various Natural Language Processing (NLP) tasks with zero-shot, one-shot, and few-shot learning. The authors established that larger models, when given carefully crafted prompts, could generalise to new tasks without explicit fine-tuning. They highlighted the effectiveness of in-context learning for various applications, such as translation, reasoning, and text completion. Nevertheless, they identified some datasets where GPT-3’s few-shot learning still struggled.

The potential employment of LLMs for automated short answer scoring was investigated by Chamieh et al. (2024). The researchers evaluated zero-shot and few-shot performance of GPT-3.5, GPT-4, LLaMa-70b, LLaMa-13b, LLaMa-7b compared to pre-trained language models – BERT and classical Support Vector Machine (SVM). The results demonstrated that GPT-4 achieved performance very close to the upper bound BERT and outperformed SVM model. However, LLMs did not yet match the accuracy of supervised models due to overfitting in certain tasks.

To improve LLM reasoning abilities, Kojima et al. (2022) introduced "Zero-Shot Chain of Thought" (Zero-shot-CoT) prompting. According to the study, adding "Let's think step by step" into queries considerably improved performance on tests involving logical, symbolic and arithmetic reasoning. Zero-shot-CoT outperformed Zero-shot LLM performance, the accuracy on MultiArith increased from 17.7% to 78.7% and GSM8K from 10.4% to 40.7% with large-scale Instruction-Tuned Generative Pre-Trained Transformer (InstructGPT) model (text-davinci-002).

An optimised prompting technique for Named Entity Recognition (NER) in few-shot scenarios was demonstrated in Cheng et al. 's work from 2024. The authors divided the prompts into three different parts: the task definition, followed by the few-shot demonstration, and the output format. The task definition directed LLMs in performing NER tasks. The few-shot demonstration supported LLMs in understanding NER task objectives through individual output demonstration and the output format limited LLMs’ output to avoid providing unessential results. This method improved accuracy by providing structured examples within prompts and demonstrates that LLMs could generalise well with minimal labeled data.

### 2.3.4 Limitations and Critical Reflections

Along with the literature highlighting success of LLMs for predictive modeling, some authors proffer a more critical perspective. For instance, Perez et al. (2021) critically examined the genuine few-shot potentials of LLMs. The authors found that regardless of the widespread implementation of LLMs within time series forecasting, they did not appear to sufficiently amend the forecasting performance. The results were often becoming better when the LLM component was eliminated or replaced with a basic attention layer. They suggested focusing on the tasks that could be unlocked by LLMs, such as time series reasoning or social understanding.

Opposite research by Beneduce et al. (2024) tested capabilities of 15 LLMs to act as zero-shot next-location predictors. The results indicated that LLMs could achieve up to 36.2% accuracy. LLMs may justify their decisions despite the size of the models, which emphasised LLMs’ strength in interpretability.

A review of the core literature on the application of LLMs models in forecasting tasks and real estate price prediction reveals several limitations that highlight the need for further research in this area. Recent studies showed that LLM-based applications in the forecasting field started to gain noticeable attention. However, further exploration is still required to understand the LLMs’ effectiveness in domain-specific context such as real estate rental forecasting. Although LLMs have demonstrated promising capabilities in financial datasets, their forecasting power in geographically localised housing markets, like Ireland’s rental sector, remains insufficiently examined. In addition to that, there is limited research comparing these models directly with classical forecasting approaches under the same experimental conditions.

# 3. METHODOLOGY

This chapter of the report describes the methodology used to guide the research, concentrating on the development along with the evaluation of forecasting models on historical rental data.

## 3.1 Primary Research Methodology

The primary research method used in this project is a quantitative experimental approach (Bryman, 2016; Ghanad, 2023). Using the same dataset of rental prices in Ireland, the study creates control and experimental groups.

The control group includes traditional forecasting models, which are trained and evaluated using the full set of relevant features. The control group with these models serve as the baseline for evaluating forecasting performance. With regards to the experimental group, the group consists of LLMs. To analyse what group performed better, the control or the experimental, standard forecasting metrics are used to compare the forecasting accuracy.

Unlike traditional forecasting models, this primary research methodology facilitates an evidence-based experimentation of the forecasting capabilities of LLMs.

## 3.2 Research Design/Framework

To conduct the project, the research framework has been structured around the following primary stages:

### 3.2.1 Data Acquisition Stage

The first stage of the research starts with the data gathering phase and focuses on acquiring a suitable dataset of historical rental prices in Ireland. The dataset will be selected from publicly available open-source platforms, ensuring the data is relevant to the research objectives.

### 3.2.2 Data Exploratory and Preprocessing Stage

Following the selection of an appropriate dataset, the next fundamental stage focuses on conducting Exploratory Data Analysis (EDA) (Tukey, 1977). The focus of this phase is to understand the underlying structure and quality of the obtained data. Key steps include data inspection, handling missing data and outliers, data visualisation behaviour to identify patterns. The insights gathered from this stage are important for guiding feature engineering and model selection (Pyle, 1999).

### 3.2.3 Model Selection Stage

Once the EDA is performed, the process proceeds to the model selection stage. The choice of the traditional models will be guided by the nature and characteristics of the original dataset and the findings from the EDA phase. Regarding the choice of LLMs, it will be based on relevant findings from literature review.

### 3.2.4 Models Evaluation Stage

Assessing predictive performance of the chosen models will be performed in the evaluation stage. To ensure fairness in comparison, each model will be evaluated on the same train-test splits. To assess model performance, standard forecasting metrics will be employed to quantify the accuracy of each selected model.

Figure 1 illustrates the proposed methodology in the flowchart format.



Figure 1: Flowchart of the proposed methodology. (Source: Author's analysis)

## 3.3 Ethical and Licensing Considerations

### 3.3.1 Data Privacy and Anonymity

The data collection process complies with academic ethical standards, which includes the proper attribution, data access permissions, and respect to the intellectual property. The dataset used in this research was sourced from the Central Statistics Office (CSO) (Central Statistics Office, 2019) of Ireland via their open data portal (RIQ02). The used dataset provides aggregated quarterly rent prices and does not include personally identifiable information (PII). Hence, there is no potential risk of identifying any individual landlords or tenants.

* The data complies with the Statistics Act of 1993, the General Data Protection Regulation (GDPR), and statistics confidentiality rules (European Union, 2016).
* Attribution to the CSO is provided as required by their Copyright Policy (CSO Copyright Policy, no date).
* The dataset is available under the Creative Commons Attribution 4.0 (CC-BY) license (Creative Commons, no date).
* In full accordance with legal and ethical requirements, no effort was taken to re- identify anonymised data (European Union, 2016).

### 3.3.2 Fairness and Bias

It is acknowledged that forecasts of rental prices may have an impact on decisions with social consequences (for instance, market behaviour or policy suggestions), despite the fact that demographic or socioeconomic data was not incorporated into the forecasting models. This study on rental price forecasting does not present predictions as pricing advice or policy recommendations. All forecasting results in this research are presented as academic forecasts only as technical insights based on historical trends and not intended for financial decision-making or regulatory guidance.

### 3.3.3 Licensing and Intellectual Property

All used models in this research were utilised in accordance with their respective licenses.

* GPT-2 (Radford et al., 2019) is open source, released under a modified MIT License, which permits unlimited usage for commercial and research purposes as long as proper acknowledgement is provided. The license can be found at:

https://github.com/openai/gpt-2/blob/master/LICENSE

* GPT-3.5 and GPT-4-turbo (OpenAI, 2024): Proprietary LLMs, accessible through OpenAI's API and requires compliance with OpenAI’s licensing terms and citation in academic work.
* Both the Amazon Chronos forecasting models and the NeuralForecast library to utilise the Time-LLM, used in this work are open source and licensed under the Apache License 2.0 (Apache, 2004). As long as the original license and copyright notices remain intact, this permissive license permits both personal and commercial use, modification, and redistribution of the software. These licensing conditions guarantee that the tools can be used in practical and academic projects while adhering to open-source standards.

## 3.4 Data Collection

The dataset for this research was downloaded from the CSO website using API, available at: <https://data.cso.ie/table/RIQ02>.

The dataset contains quarterly data of historical rental prices in Ireland from 2007 until 2024. The downloaded dataset includes 1 273 776 observations and 5 features.

The features and their corresponding descriptions for the obtained dataset are presented in Table 1. The license is provided and observed in the Ethics Considerations section.

**Table 1. Dataset Features and Descriptions**

| **Dataset Feature** | **Description** |
| --- | --- |
| Average Rent (€) | the average rental prices in euros |
| Quarter | the period in which the rent price was recorded (e.g., 2007Q4) |
| Number of Bedrooms | the number of bedrooms in the rented property |
| Property Type | the type of the rented property (e.g., Apartment) |
| Location | the area where the property is rented (e.g., Dublin) |

(Source: Author's analysis)

## 3.5 Data Preprocessing

Data preprocessing is one of the most essential parts that uses a variety of methods to guarantee the dataset is consistent, clear and relevant for model training (García et al., 2015).

### 3.5.1 Handling Missing Data

The data exploration showed that the dataset contains 917 534 missing values, which means that 72.03% data from the original data is missing. In order to reduce the impact of missing values and maintain consistency, the dataset was filtered to keep the most representative entries.

Further analysis of missing values in the "Average Rent (€)" column revealed significant differences between “Property Type” and “Number of Bedrooms”. For instance:

* Among property types, the lowest percentage of missing values in “Average Rent (€)” was observed in "All property types" at 37.11%. In contrast, the remaining property types - “Apartment”, “Detached house”, “Other flats” “Semidetached house”, “Terrace house” - had noticeably higher percentage of missing values: 60.50%, 89.01%, 92.11%, 74.08% and 79.38%, respectively.
* Among the number of bedrooms categories, “All bedrooms” had the lowest percentage of missing values in Average Rent (€), at 52.58%. Though, the following categories - “1 to 2 bed", “1 to 3 bed”, “Four plus bed”, “One bed”, “Three bed”, and “Two bed” - were higher at 70.95%, 57.22%, 86.25%, 87.01%, 74.64%, and 75.57%, respectively.

Due to the high proportion of missing values within certain property types and number of bedrooms categories, the dataset was filtered to focus on "All property types" and “All bedrooms”, which are more complete. This provided a more reliable foundation for model forecasting.

### 3.5.2 Location Filtering

In the original dataset, the “Location” column contained 446 unique values. These values represent a broad variety of geographic areas across Ireland, including counties, suburbs and districts of cities. This substantially increases data sparsity and inconsistency, particularly when it comes to "Average Rent (€)" missing numbers. With the goal to standardise the locations, the dataset was filtered to contain only 26 major counties in Ireland, including Dublin, Cork, Galway, and others. Following the application of the filtering criteria, the filtered dataset consisted of 1.768 records across 26 counties, with 68 data points for each county.

Furthermore, it was decided to target the dataset to the Dublin location solely. Dublin was selected for the prediction since it is the capital of Ireland. The final Dublin dataset confers a clear structured and consistent quarterly time series of 68 average rent prices (data points), making it suitable for forecasting.

### 3.5.3 Time Formatting

Due to the Dublin dataset being a time series, it is essential to transform the “Quarter” column into a standard datetime format. This will guarantee temporal consistency in the quarterly time series. To enable chronologically accurate order, each element (e.g., " 2007Q4") was mapped to a matching start date (e.g., "2007-10-01").

### 3.5.4 Outlier Detection

With the aim to detect any extreme values in the rent prices that could skew the forecasting findings, the detection of outliers was performed on the Dublin dataset. Firstly, the Interquartile Range (IQR) technique was applied to the “Average Rent (€)” feature. This is one of the most useful statistical methods, which estimate a data point as an outlier if it is more than 1.5 × IQR ‍ above the third quartile or below the first quartile. Explaining in other terms, low outliers are below Q1 - 1.5×IQR and high outliers are above Q3 + 1.5×IQR (Vinutha, Poornima and Sagar, 2018). Secondly, the visual method using a boxplot was implemented to inspect any possible abnormality (Walfish, 2006). Figure 2 illustrates a box plot of the “Average Rent (€)” values for Dublin, where the IQR spans from €1103 to €1725 and the median around €1330, with no extreme outliers.

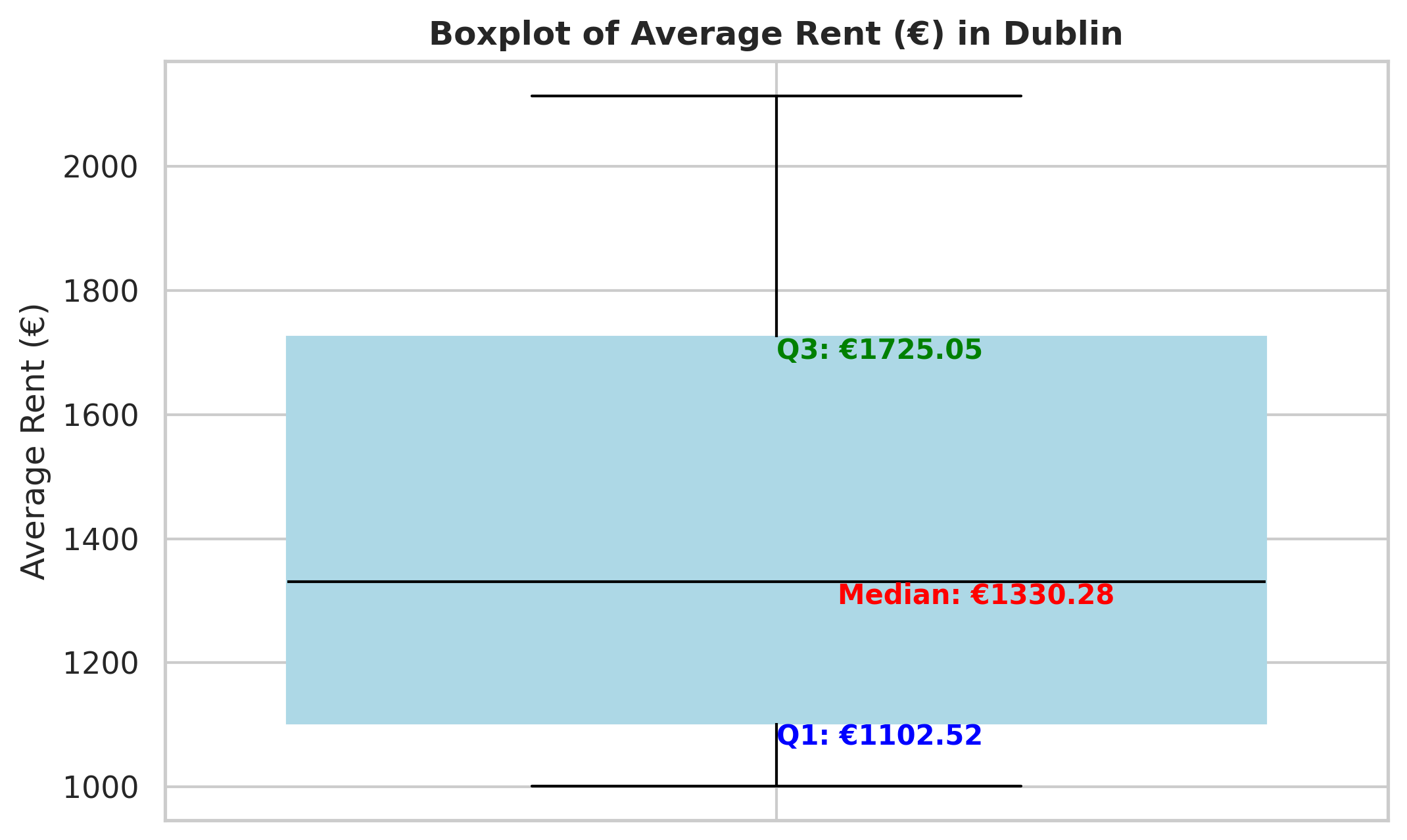


Figure 2: Boxplot of Average Rent (€) in Dublin. (Source: Author's analysis)

Using these two, statistical and visual techniques, no severe outliers were identified. This indicates that the trend of rent prices in the Dublin dataset is generally stable over time.

### 3.5.5 Stationarity Test

Stationarity is one of the fundamental properties to consider when working with time series. A time series known as stationary, when statistical attributes – such as mean, variance, and autocorrelation – remain stable over time (Palma, 2016).

To check if the average rent time series for Dublin is stationary or not, a Hypothesis test called the Augmented Dickey-Fuller (ADF) test was applied. The ADF test's Null Hypothesis (H0) states that the time series has a unit root and is thus non-stationary. The ADF test calculates a test statistic, and the p-value associated with this statistic indicates how strongly the evidence opposes the null hypothesis. If the p-value is less than a chosen significance level (usually set at 0.05), the null hypothesis is rejected in favor of the alternative hypothesis, which concludes that the time series is stationary (Mushtaq, 2011).

The ADF test on Dublin rent time series returned a statistic of 0.163 and a p-value of 0.970. This is much higher than the significance level of 0.05. This suggests that the series is non-stationary as the null hypothesis of a unit root cannot be rejected.

In an attempt to amend this, first-order differencing was applied to the Dublin rent time series (Hyndman and Athanasopoulos, 2018). After first differencing, the ADF test was conducted again and returned a statistic of -3.201 and a p-value of 0.0199, both of which are below the 0.05 threshold. The differenced rent series is stationary, as demonstrated by the rejection of the null hypothesis.

This transformation was required in order to stabilise the Dublin rent series and prepare it for time series forecasting models.

### 3.5.6 Time Series Decomposition

The Dublin rent series was further examined using the classical decomposition method. The classical decomposition process was divided into two types: multiplicative and additive (Brockwell and Davis, 2013). In relation to a time series with seasonal period m:

* m = 4 for quarterly data,
* m = 12 for monthly data,
* m = 7 for daily data with a weekly pattern (Hyndman and Athanasopoulos, 2018).

Hence a time series using an additive model, the elements are added as follows:

|  | (1) |
| --- | --- |

where, represents the data, is the seasonal component, represents the trend-cycle component, and is the remainder element, all at period (Hyndman and Athanasopoulos, 2021).

Whereas the multiplicative model suggests that the components are multiplied together as follows:

|  | (2) |
| --- | --- |

(Hyndman and Athanasopoulos, 2021).

In this study, using a quarterly period (lag = 4), both additive and multiplicative seasonal decomposition models were tested. Figure 3 shows that both models effectively extracted the underlying trend, seasonality, and residual components.

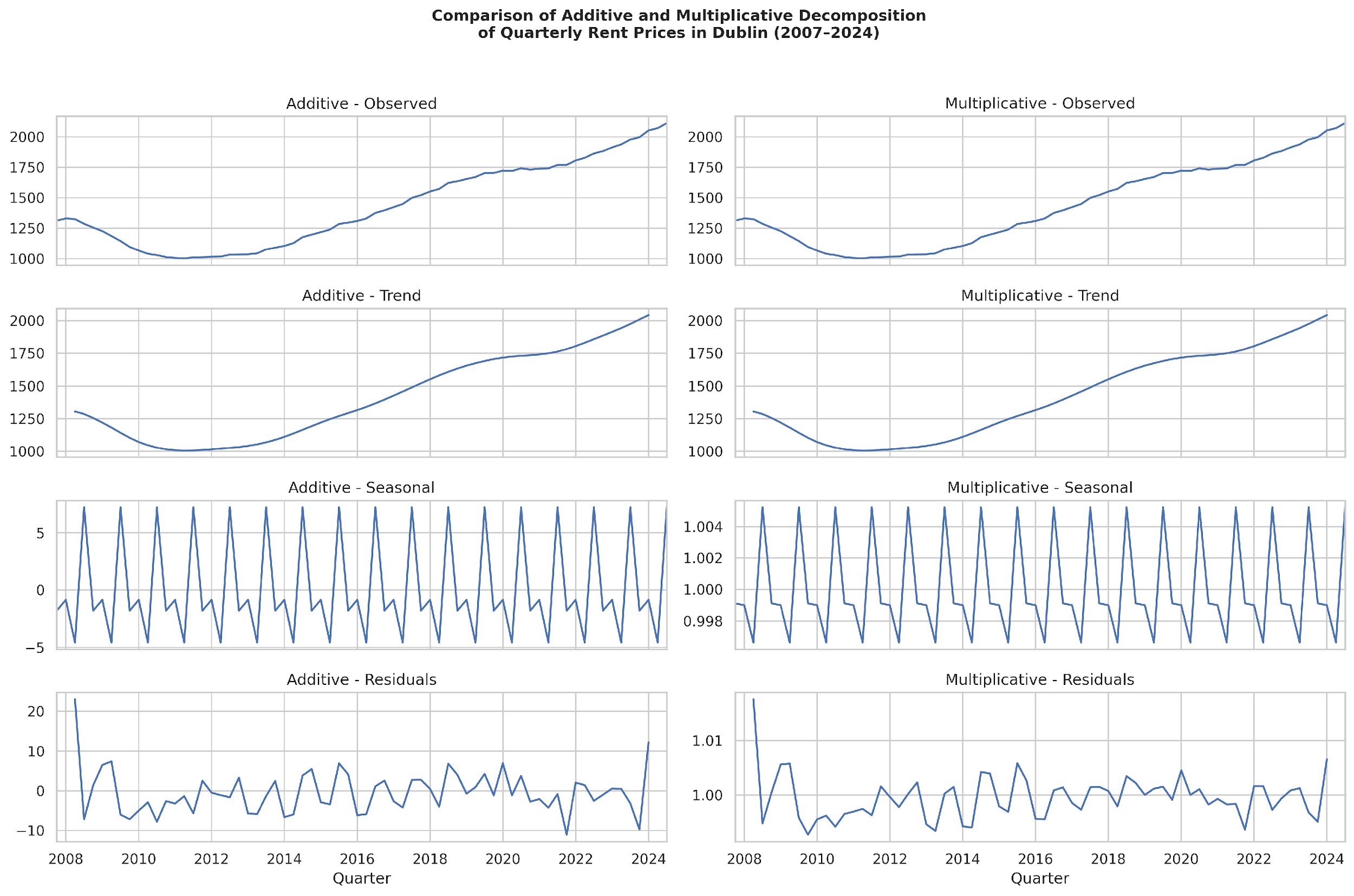


Figure 3: Additive and multiplicative decomposition of quarterly average rent prices in Dublin from 2007 to 2024. (Source: Author's analysis)

The seasonal plots from both the additive and multiplicative models demonstrated a regular repeating pattern every four quarters. That indicates there is a presence of quarter seasonality in the Dublin rent series. Additionally, the trend component displayed an upward trend in Dublin rental costs over time, especially after 2012.

The additive model was considered to be more suitable for further research since the size of seasonal variations was mostly constant across time, in place of growing with the trend.

### 3.5.7 Autocorrelation Analysis

The further analysis is aimed at understanding the relationship between observations at various time points. Autocorrelation measures the correlation between a time series and its own past values at different lags across successive time periods. The Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) are two essential instruments for this (Box et al., 2015).

The ACF examines the correlation between observations in a time series for a set of lags. The PACF is similar to the ACF, despite it displays the correlation between two observations that cannot be explained by their shorter lags. By removing the impact of intermediate observations, the PACF isolates the direct relationship between observations at different lags (Nelson, 1998).

With regards to the Dublin rent prices time series, the ACF and PACF plots were created. Initially, the first-order differenced series was plotted to visualize changes over time. The ACF and PACF of first-order differencing series are illustrated in Figure 4.

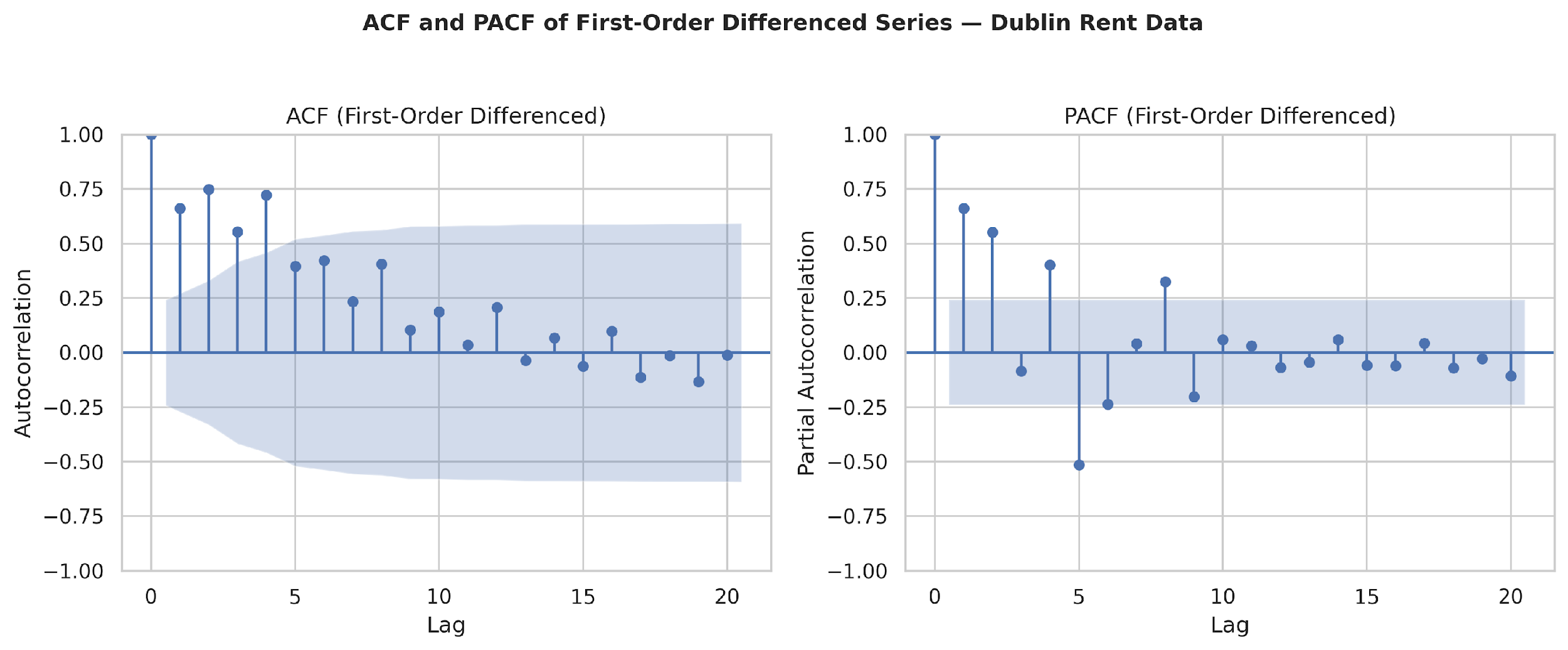


Figure 4: ACF and PACF of first-differenced series in Dublin rent time series. (Source: Author's analysis)

The ACF plot shows a gradual decrease with prominent spikes at lags 2, 4, 6, 8, 10, 12, 14 and 16. This indicates high autocorrelation at both short-term and seasonal intervals. The pattern at lags 4, 8, and 12 supports a quarterly seasonal component, while the extra spikes at intermediate lags (e.g., lags 2 and 6) imply short-term persistence or overlapping cyclical behaviour within the Dublin rent time series.

Examining the PACF plot, the lags 1 and 2 exhibit high spikes, suggesting strong short-term autoregressive correlations and the direct correlation is concentrated within the first two lags.

Subsequent step was to plot differenced Dublin rent data at a seasonal lag of 4 (quarterly) (Hyndman and Athanasopoulos, 2018). Figure 5 displays the AFC and PACF plots of the seasonally differenced Dublin rent time series.

The AFC plot shows a slow, steady decrease with positive autocorrelation continuing until lag 15 and mild negative autocorrelation after lag 15. On the other hand, the PACF plot drops drastically after lag 1, indicating that the first lag captures most of the direct autocorrelation.

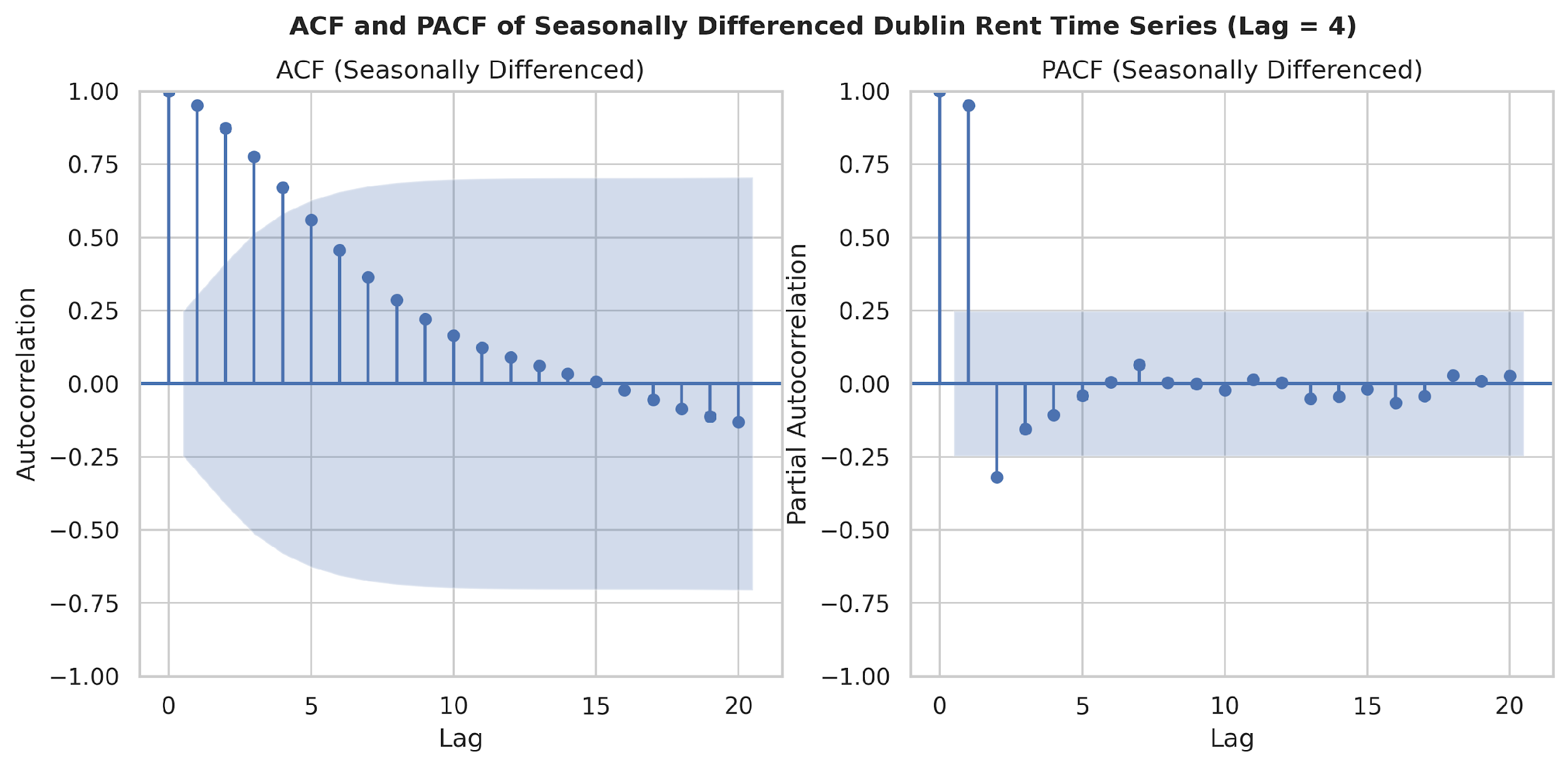


Figure 5: ACF and PACF of seasonally differenced series in Dublin rent time series. (Source: Author's analysis)

This autocorrelation analysis suggests that SARIMA is an appropriate traditional forecasting model to choose. In addition, it provides beneficial guidance for choosing the right forecasting model parameters in the following modelling stage.

## 3.6 Models Selection

### 3.6.1 SARIMA

SARIMA is an extension of the non-seasonal ARIMA model, specifically designed to handle seasonal patterns within data. SARIMA is a powerful technique for predicting since it can identify both short-term and long-term dependencies in the data. SARIMA model combines four components:

1. MA component: models the relationship between the current data point and past prediction errors.

2. AR component: captures how the current value and its past values relate to one another. This facilitates the model's learning from repeated patterns over time.

3. Integrated (I) component: Refers to differencing, to transform non-stationary data into stationary.

4. Seasonal component (S): allows the model to explicitly learn seasonal patterns by adding seasonal lags to the AR, I, and MA components (Milenković et al., 2016).

Formally, the SARIMA model represented as SARIMA(p, d, q)(P, D, Q, s), where:

* AR(p): component of order p
* MA(q): component of order q
* I(d): component of order d
* Seasonal AR(P): component of order P
* Seasonal MA(Q): component of order Q
* Seasonal I(D): component of order D
* and s: seasonal period

Mathematically, the SARIMA model represented as follows:

|  | (3) |
| --- | --- |

Where, is the AR polynomial of order , is the seasonal AR polynomial of order , is the MA polynomial of order , is the seasonal MA polynomial of order , is the non-seasonal differencing operator of order, is the seasonal differencing operator of order , is the seasonal period, is the white noise sequence at time and is the backshift operator such that (Box, Jenkins and Reinsel, 2008; Cryer and Chan, 2008).

Based on the analysis of the ACF and PACF plots of the first order and seasonally differenced series in Section 3.4.7 (Autocorrelation Analysis), the following model parameters were selected:

* Non-seasonal components:

p = 1 (the high PACF spike at lag 1),

d = 1 (stationarity achieved after applying first order differencing),

q = 0 (the ACF showed no sign of short-term MA structure).

* Seasonal components:

P = 1 (the PACF shows a high spike in lag 1),

D = 1 (seasonal pattern removed with the first seasonal differencing),

Q = 0 (the ACF plot illustrate a non-seasonal MA component is unnecessary),

s = 4 (quarterly seasonality).

Therefore, the SARIMA model configuration: SARIMA(1, 1, 0)(1, 1, 1)[4].

### 3.7.2 Prophet

Prophet (Taylor and Letham, 2018) is an additive model and a procedure for time series forecasting developed by Facebook’s Core Data Science team. Non-linear trends in Prophet fit with yearly, weekly, and daily seasonality and holiday effects. It is most effective with time series exhibiting clear seasonality and several seasons of historical data (Jadon, Milczek and Patankar, 2021).

In contrast to traditional models such as SARIMA, Prophet employs a decomposable time series model, where forecast is the sum of trend, seasonality, and holiday components. The following equation represents this combination of the components:

|  | (4) |
| --- | --- |

Where, models the non-linear trend, captures periodic changes (e.g., daily or weekly seasonality), represents holidays events and is the error term (Taylor and Letham, 2018).

Although Prophet is not a traditional time series forecasting models like SARIMA, it was chosen for this research due to several practical advantages:

1. Firstly, Prophet is highly robust to non-stationary data, missing values and generally handles the outliers effectively (Facebook, na date; Taylor and Letham, 2018). As was reviewed in Stationary analysis, the Dublin rent time series exhibited non-stationary behaviour;
2. Secondly, Prophet automatically models multiple seasonality using Fourier series (Facebook, na date; Taylor and Letham, 2018). In the case of the Dublin rent time series, the data exhibited a quarterly seasonal pattern;
3. Lastly, Prophet's user-friendly architecture enables effective model development without requiring manual parameter tuning or extensive preprocessing (Facebook, na date).

The Prophet default parameters were selected as followed:

* yearly seasonality was enabled, while weekly and daily seasonality components were removed, because the Dublin dataset is quarterly;
* Manually set up a custom quarterly seasonality, with a period of 91.25 days and a Fourier order of 5.

### 3.6.3 LLMs: zero and few-shot prediction using ChatGPT

1. Prompt-as-Prefix Approach for Rent Price Forecasting

Prompting is a straightforward method for task-specific activation of LLMs. Prompt-as- Prefix (Jin et al., 2023) is a concept that improves the LLMs’ adaptivity to downstream tasks and it is possible to smoothly incorporate several data modalities as prompt prefixes. This concept was adapted in work by Chen and Si (2024) and showed promising results. Inspired by their research, this project investigates if time series forecasting of rental prices in Dublin could be improved using a comparable LLM-based prompting technique.

Each prompt was constructed using three components:

1. **Dataset Context.**

In zero-shot setting the LLMs was provided with a domain-specific description to indicate that the problem included forecasting average rent prices for Dublin, characterised by a consistent increasing trend and quarterly seasonality. In the few-shot configurations the geographic domain was not referenced, only trend and seasonality. The given context about the dataset furnishes the LLM with vital background information on the input time series.

1. **Task Instruction.**

In the task instruction part, the LLMs was asked to forecast the *n* next values equal to the size of testing dataset and return them exactly as two Python lists:

* the first list is for future quarterly dates starting from the first quarter data point in the testing dataset in YYYY-MM-DD format;
* and the second list corresponds to the predicted rent values in float format.

The task instruction serves as a guide for the LLMs, specifying what the model needs to predict and how the output should be structured for the forecasting task.

1. **Input Statistics (Few-Shot Examples).**

When performing few-shot forecasting, the most recent n observations were taken from the training dataset and included in the prompt in the string format “Quarter: Rent Value” (e.g., Q10-2007:1312.69). The number of observations were 0, 1, 5, 10, 20, and 30 across experiments 0, 1, 5, 10, 20, 30-shot, respectively. Providing the LLM some examples with a sequence of recent historical values to facilitate pattern recognition.

Table 2 presents an example of the prompt structure used in the Prompt-as-Prefix approach for this study.

**Table 2. Prompt Structure Example**

| **Components** | **Example Content** |
| --- | --- |
| Dataset Context | **Zero-Shot**: You are a rent prediction model. The data has a clear upward trend and quarterly seasonality. You must forecast the next {testing data length} average rent values for Dublin.  **Few-Shot**: You are a rent prediction model. The data has a clear upward trend and quarterly seasonality. |
| Task Instruction | Please predict the next {testing data length} average quarterly rents for Dublin. Return exactly two Python lists:  - Dates: A list of {testing data length} future quarterly dates starting from '{start date}', spaced 3 months apart, in YYYY-MM-DD format  - Values: A list of {testing data length} float rent values (do not include € or any placeholders). Only return the two lists — no explanation. |
| Input Statistics | Here are the last {n shot} data points: {training data} |

(Source: Author's analysis)

1. Forecasting Setup
2. Zero-Shot Forecast.

In the zero-shot prediction setting, the LLM was not given any historical rent values and asked to predict average rent prices for Dublin, therefore the prediction is based only on the provided prompts. This approach evaluates the LLM‘s capacity to comprehend the task as well as accurately predict without relying on specific samples of data.

1. Few-Shot Forecast.

In the few-shot forecasting scenario, a prefix containing the most recent n rent observations from training dataset was formatted as "Quarter: Value" pairs and added to the prompt. The number of included examples varied in the different configurations: 1-shot, 5-shot, 10-shot, 15-shot, and 20-shot forecasting. This setting uses the capability of the LLM to learn the underlying trend from the given data points.

1. Model Configuration

The two state-of-the-art LLMs from OpenAI, ChatGPT-3.5 and ChatGPT-4-turbo, were used in the present research. These models were accessed through the OpenAI API using the gpt-3.5-turbo and gpt-4-turbo endpoints (OpenAI, 2024), respectively. The models were employed in their default setting with no retraining or fine-tuning on the target tasks. The zero-shot and few- shot prompting were the only ways of doing forecasting using the model’s in-context learning.

## 3.6.4 Time-LLM

Following the investigation of LLM-based for the forecasting of time series, the Time-LLM framework was also utilised to assess its capability in the forecasting of Dublin rent prices. In the Time-LLM, the input sequence is tokenised through patching and is realigned to a low-dimensional word embedding space with multi-head attention. The outputs are then passed through a frozen pre-trained language model along with the descriptive statistics feature embeddings. The language model outputs are then flattened and then sent through a linear layer to generate a forecast (Jin et al., 2023).

The prompt construction with three components were applied as shown in Table 3:

**Table 3. Time-LLM Prompt Structure Example**

| **Components** | **Example Content** |
| --- | --- |
| Dataset Context | This dataset contains quarterly average rent prices in Dublin. Prices follow an upward trend with clear seasonal variation. |
| Task Instruction | Use the historical values to forecast the next quarterly average rent prices. |
| Input Statistics | The input has a minimum value of 1001.70, a maximum of 1670.52, and a median of 1207.71. The top autocorrelated lags are [0, 2, 4, 6, 8]. |

(Source: Author's analysis)

*Time-LLM Parameters*. The following parameters were set to train the Time-LLM model:

* Forecast horizon: the same as the test set size, since the goal is to forecast future values for the test period;
* Input size: 16.

An input window of 16 was selected because it captures 4 years of historical data and provides 31 usable sequences, which is sufficient to consider seasonality and long-term trend in Dublin rent prices.

* Batch size: 16.

The batch size controls the number of input sequences to be processed in parallel during training. The size of 16 was chosen in consideration of the number of available training windows to avoid the processing of the data in batches being greater than the available data. In other words, the aim is to find a balance between training stability and the rate of processing without overfitting the small dataset.

* Maximum steps: 10.

Maximum steps parameter was capped at 10 to prevent overfitting due to the small training size. With the Time-LLM architecture being built on top of a frozen LLM and the only training being on the light layers, there was no need for extensive training to produce reliable forecasts.

* Language model: openai-community/gpt2 (Radford et al., 2019).

More powerful models such as GPT-3.5 or GPT-4-turbo are not implemented with the open-source Time-LLM. Therefore, the openai-community/gpt2 pre-trained model from Hugging Face was selected to be used with the Time-LLM framework.

* Padding: Enabled.

The padding setting was activated to facilitate the proper alignment of the short sequences during training, specifically helpful when the historical windows are small.

* Temporal frequency: Quarterly.

The temporal frequency was set to quarterly to align with the data granularity of the Dublin rent data.

### 3.6.5. Amazon Chronos

Our last selection was the Chronos family of pretrained time-series forecasting models built on top of language model architectures by Amazon. Chronos is a probabilistic model that treats time-series as the time-series as a sequence and employs Google Artificial Intelligence’s (AI’s) T5 model to train it. As was said previously, zero-shot forecasting is the capability of the models to produce forecasts on unseen datasets. According to the original paper, this is achieved through the employment of synthetic data and actual data to train the Chronos models (Ansari et al., 2024).

The Chronos model works as follows:

1. No Use of Explicit Timestamps.

Chronos models do not represent timestamps (for example, hours, days, months) directly. Rather, all the time data is treated as a sequence of values — exactly as a sentence is an array of words — and depends on the sequence of the values to express temporal structure (Ansari et al., 2024).

2. Tokenisation of Numerical Values.

First, the values of the time series are mean-scaled to normalise the range, and then quantizing (binning/grouping values) into a fixed vocabulary. Every quantised value is considered as a token, as is the case for words in NLP (Ansari et al., 2024).

3. Sequence modeling using T5.

These token sequences are processed through a T5-type transformer model. The model is trained to predict the future tokens (or values) from the past context, using the Sequence-to-Sequence (seq2seq) architecture:

Encoder accepts the context window (previous values);

Decoder produces the forecast tokens (future values) (Ansari et al., 2024).

4. Probabilistic Forecasting through Sampling.

Chronos facilitates probabilistic forecasting through the sampling of various output sequences from the decoder. The result allows Chronos to represent uncertainty and generate prediction intervals instead of point forecasts (Ansari et al., 2024).

5. Post-processing Outputs.

The dequantised sequence of tokens is converted into continuous numerical values. The forecasting distribution is then summarised by calculating quantiles (for instance, the 10th, 50th, and 90th percentiles) from the sample sequences (Ansari et al., 2024).

In this research three Chronos Models were chosen, including chronos-t5-small, chronos-t5-base and chronos-t5-large. The three models were tested using the same set of configuration parameters:

* Probabilistic forecasts were created by sampling 36 future paths and taking the 10th, 50th, and 90th percentiles to represent forecast uncertainty.

## 3.7 Performance Metrics

The forecasting performance of all the models employed within this work were evaluated and compared using the below metrics:

### 3.7.1. Mean Squared Error (MSE)

MSE measures the amount of error in statistical models. It is the evaluation of the average squared difference between the observed and the predicted values. The MSE equals zero when the model has no error. The higher the model error, the greater the value of the MSE (Adhikari and Agrawal, 2013). The mathematical interpretation of MSE is:

|  | (5) |
| --- | --- |

Where, is the observed value, is the corresponding predicted value and is the number of observations (Das, Jiang and Rao, 2004).

### 3.7.2. Mean Absolute Error (MAE)

MAE is an effective measure to analyze the accuracy of regression models. It measures the average absolute difference between the predicted values and the target values. In contrast to other metrics, MAE does not square the errors, which implies that it attributes the same weight to all errors, regardless of their direction. This makes MAE very helpful when one is interested in knowing the magnitude of the errors without considering whether they are overestimations or underestimations (Adhikari and Agrawal, 2013). The formula to compute MAE is as follows:

|  | (6) |
| --- | --- |

Where, is the number of data points, is the predicted value and is the actual target value for data point (Hodson, 2022).

### 3.7.3. R² Score

R² is another measure of how well a statistical model predicts. The lowest possible R² equals to 0 and the highest equals to 1 (Albeladi, Zafar and Mueen, 2023). To put it briefly, the better a model predicts, the closer its R² will be to 1. The mathematical formula of R² is:

|  | (7) |
| --- | --- |

Where, - sum of squared residuals and - total sum of squares (Chicco, Warrens and Jurman, 2021).

### 3.7.4. MAPE

MAPE is a commonly used metric in time series forecasts, representing average absolute error as a percentage of the actual values (Adhikari and Agrawal, 2013). The mathematical formula of MAPE is:

|  | (8) |
| --- | --- |

Where, ​ is the actual value at time , is the forecasted value at time and is the total number of observations (Shcherbakov et al., 2013).

## 3.8 Hyperparameter Tuning

### 3.8.1 SARIMA Hyperparameter Tuning

For the SARIMA model, hyperparameter tuning was performed using Auto-SARIMA (Chaturvedi et al., 2022), which was implemented via the auto\_arima() function from the pmdarima library. This function automatically selects the best combination of AR, I and MA, as well as the seasonal component. The process was performed through a stepwise search to minimise information criteria such as the Akaike Information Criterion (AIC). During the tuning process, a quarterly frequency (m=4) was specified and seasonal=True was enabled.

The auto\_arima identified that the best-fitting model was SARIMA (1,1,2)(0,1,1)[4]. These parameters were chosen based on the lowest AIC score equaling 290.36.

### 3.8.2 Prophet Hyperparameter Tuning

Grid search (Petro and Pavlo, 2019; Menke, 2012) algorithm was implemented as the hyperparameter tuning technique for the Prophet model in order to explore how different parameter combinations affect predicting accuracy. The grid search was chosen for the following key features:

1. Effectiveness: The grid search is a great option for the Dublin rent time series since it is well-suited for small datasets and for models with a relatively small number of tunable parameters (Hossain and Timmer, 2021);
2. Automation: The grid search evaluates every possible combination of provided hyperparameter values, this reduces the need for manual intervention (Belete and Huchaiah, 2021);
3. Configurability: The grid search allows the researchers to control the tuning process (Ogunsanya, Isichei and Desai, 2023).

The tuned parameters are listed below:

1. Changepoint prior scale: This is one of the most impactful parameters that controls the flexibility of the trend. In other words, how much the trend changes at the trend changepoints. The tested values are: 0.001, 0.01, 0.05 and 0.1.
2. Seasonality prior scale: This parameter controls the flexibility of the seasonality components. While small values reduce the seasonality's magnitude, large values allow more seasonal variation. Tested values were 0.01, 0.1, 1.0, and 10.0.
3. Seasonality mode: The options are additive or multiplicative.
4. Add quarterly seasonality: A custom quarterly seasonal component was added using a Fourier order of 5 and a period of 91.25 days.

The best-performing configuration using the grid search was:

* Changepoint prior scale: 0.05;
* Seasonality prior scale: 1.0;
* Seasonality mode: multiplicative.

### 3.8.3 Zero and Few-Shot Hyperparameter Tuning

If the performance of the models like SARIMA and Prophet can be improved by modifying internal parameters,zero-shot and few-shot LLM-based forecasting does not involve the standard hyperparameter tuning process. Instead, the performance mostly depends on:

1. Prompt Engineering (Perez et al., 2021).

Well-structured prompts are essential for guiding the LLMs’ reasoning. During the experiment, it was noted that without a clear instruction, the ChatGPT-3.5 and ChatGPT- 4-turbo generated extra narrative text in addition to the specified lists (“Dates” and “Values”). This behaviour rendered it impossible to extract predicted numbers cleanly. For the purpose to exclude this behaviour and achieve a structured output, an instruction such as *"Only return the two lists — no explanation."* was added to the prompt command.

1. Temperature Adjustment (Davis et al., 2024).

This parameter controls the randomness of the output. Then lower the temperature, the more deterministic and focused prediction result. To remove randomness and achieve consistent results, the temperature was adjusted to 0 for all zero-shot and few-shot LLM- based forecasting experiments.

### 3.8.4 Time-LLM Hyperparameter Tuning

To optimise the Time-LLM performance, the grid search (Petro and Pavlo, 2019; Menke, 2012) algorithm was implemented over two important hyperparameters:

* input\_size: Is an autoregressive input size. Tested values were 4, 6, and 8.
* batch\_size: Is the number of different series in each batch. Tested values were 8, 16, and 32.

The grid research returned the following best-performing configuration:

*input\_size:* 8,

*batch\_size:* 16.

### 3.8.5 Chronos Hyperparameter Tuning

Chronos models, in contrast to standard forecasting models, are pre-trained transformer-based architectures designed for zero-shot and few-shot forecasting, often achieving competitive performance without the need for further fine-tuning. The underlying model weights are fixed, and users mainly interact with the model by inputting a historical context window and setting prediction parameters. As a result, no hyperparameter tuning was performed on the Chronos model itself. Instead, ensuring a fair evaluation, we applied consistent parameters to all Chronos models (small, base, and large) (Ansari et al., 2024).

# 4. RESULTS

## 4.1 SARIMA – Evaluation Results

Using both configurations obtained during the model selection and tuning phases, the SARIMA model was employed to forecast Dublin’s average rent prices. Table 4 demonstrates the performance of SARIMA (1,1,0)(1,1,0)[4] that was manually specified, using the evaluation metrics. The manually SARIMA model achieved the MSE of 3274.39, the MAE of 49.95, and the strong R² value of 0.80, indicating good predictive performance. Additionally, the MAPE was 2.74%, meaning that the average difference between the forecasted value and the actual value was 2.74%, indicated a high level of forecasting accuracy.

**Table 4. Evaluation Metrics: SARIMA(1,1,0)(1,1,0)[4]**

| **Metric** | **Value** |
| --- | --- |
| MSE | 3274.39 |
| MAE | 49.95 |
| R² | 0.80 |
| MAPE | 2.74% |

(Source: Author's analysis)

Figure 6 illustrates the actual versus predicted Dublin rental prices using SARIMA(1,1,0)(1,1,0)[4] configurations. The model relatively closely followed the seasonal increasing pattern in the test data, meaning the model captured seasonal patterns effectively.

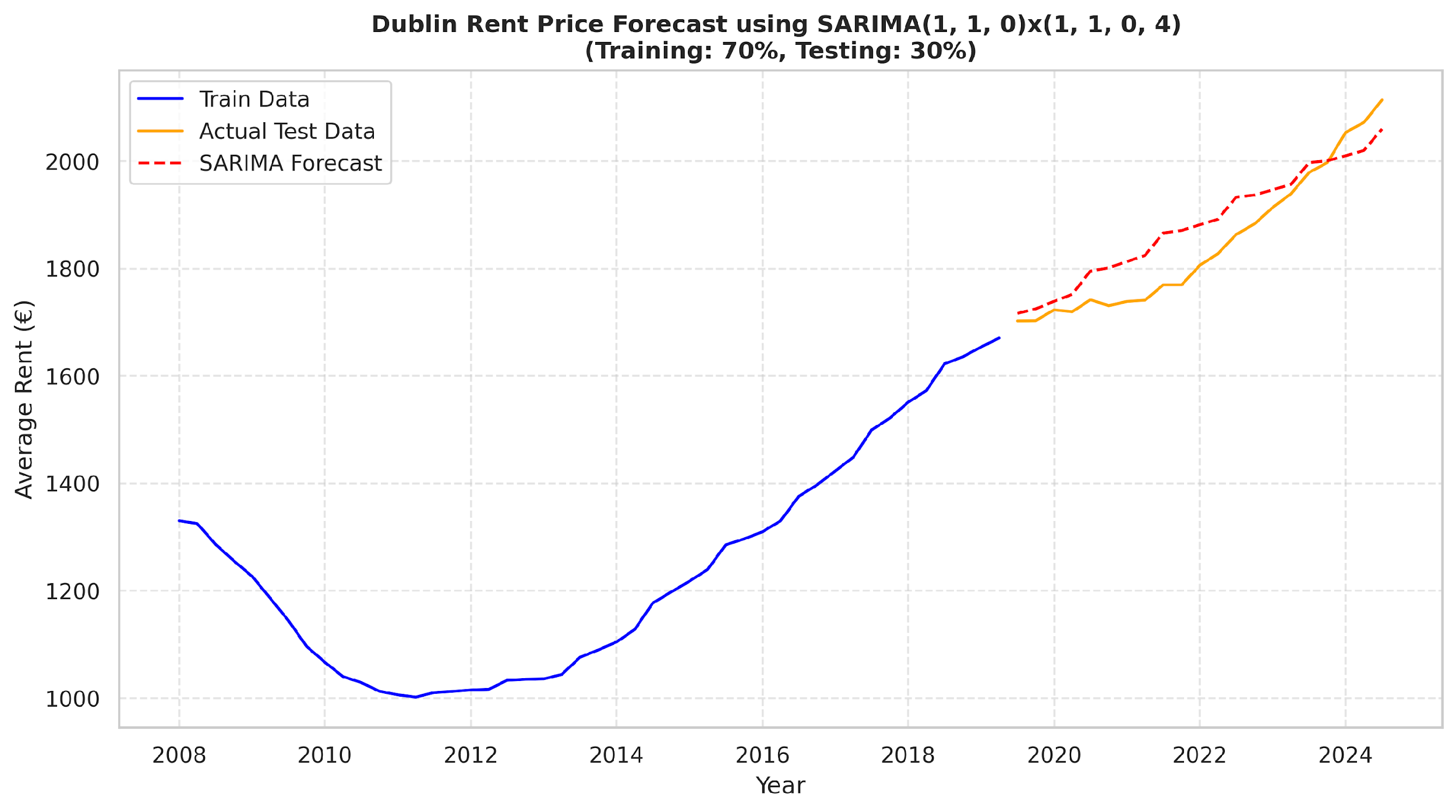


Figure 6: SARIMA(1,1,0)(1,1,0)[4] Dublin rent prices forecasting  
 (Source: Author's analysis)

The second SARIMA model was configured as SARIMA(1,1,2)(0,1,1)[4] from the tuning approach. The evaluation results of this model showed in Table 5. The model explained just over half of the variance in the test dataset, as seen by the lower R² value of 0.54. Both error metrics, the MSE and MAE recorded notably higher values of 7821.30 and 80.56, respectively. The MAPE stood at 4.38%, which is lower than the accuracy of the manual SARIMA model.

**Table 5. Evaluation Metrics: SARIMA (1,1,2)(0,1,1)[4]**

| **Metric** | **Value** |
| --- | --- |
| MSE | 7821.30 |
| MAE | 80.56 |
| R² | 0.54 |
| MAPE | 4.38% |

(Source: Author's analysis)

Figure 7 presents the rental prices forecast in Dublin created by the auto-tuned SARIMA model. As shown in the figure below, it performed worse than the manually chosen SARIMA model, even though it still generally matched the test data. The predicted line diverged considerably from the actual values, in particular from 2021 through early 2024.

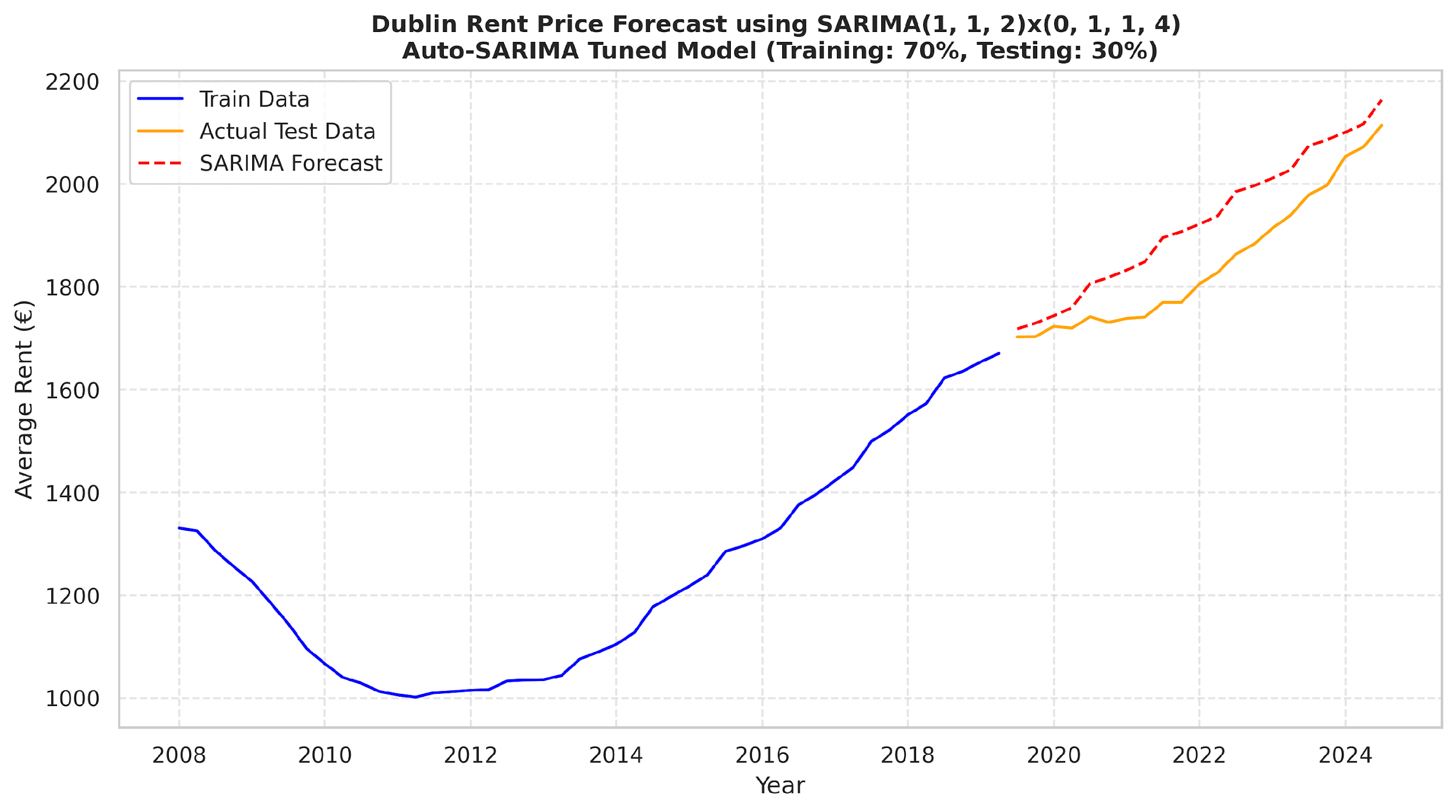


Figure 7: SARIMA(1,1,2)(0,1,1)[4] Dublin rent prices forecasting – Auto-Tuned  
 (Source: Author's analysis)

**Comparison Summary**

The forecasting results of both models were organised into a single table (Table 6) for comparative purposes.

**Table 6. Evaluation Metrics: SARIMA Comparison Summary**

| **SARIMA Order** | **MSE** | **MAE** | **R²** | **MAPE** |
| --- | --- | --- | --- | --- |
| (1,1,0)(1,1,0)[4] | 3274.39 | 49.95 | 0.80 | 2.74% |
| (1,1,2)(0,1,1)[4] | 7821.30 | 80.56 | 0.54 | 4.38% |

(Source: Author's analysis)

As seen from the table above, across all the evaluation metrics, the manually chosen SARIMA model performed noticeably better than the auto-tuned version. The SARIMA model with parameters (1,1,0)(1,1,0)[4] achieved the lower MSE, MAE and higher R2 values. Moreover, the model recorded a lower MAPE (2.74%), highlighting better forecasting performance compared to the auto-tuned SARIMA model.

## 4.2 Prophet – Evaluation Results

The default Prophet model, formatted with quarterly seasonality and additive trend components, performed noticeably worse. The Prophet forecast line in Figure 8, substantially diverges from the actual rent prices. Especially, underestimating the seasonal and exponential rise shown in the test data from 2020.

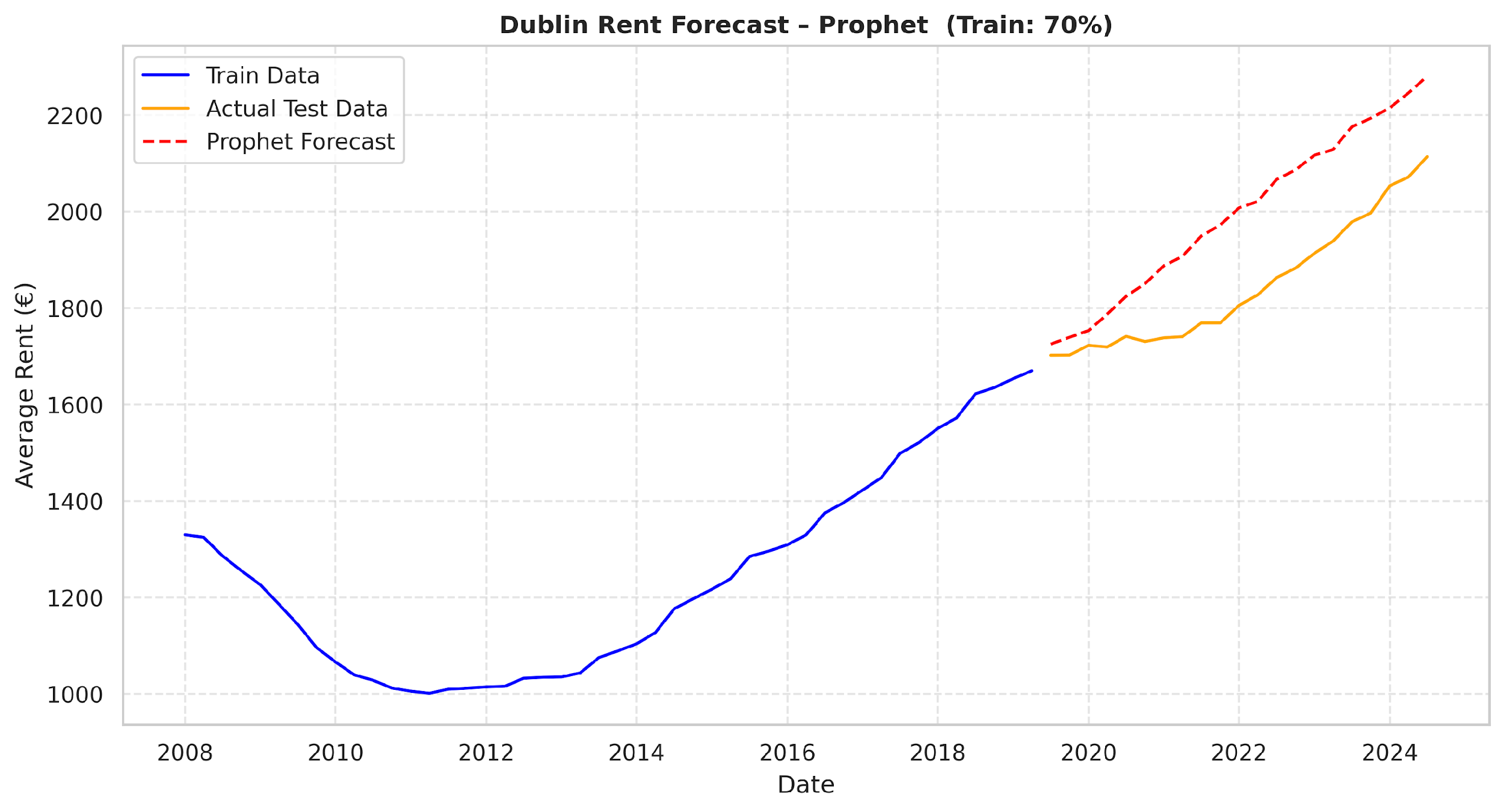


Figure 8: Prophet Dublin rent prices forecasting with default settings (Source: Author's analysis)

This is also confirmed by the evaluation metrics in Table 7:

**Table 7. Evaluation Metrics: Prophet Forecast with default settings**

| **Metric** | **Value** |
| --- | --- |
| MSE | 26324.16 |
| MAE | 150.12 |
| R² | - 0.56 |
| MAPE | 8.03% |

(Source: Author's analysis)

The negative R² value (-0.56) clarifying that the default Prophet model performed worse than a horizontal mean predictor and its generalisation is under default parameters. The model's MAPE reached 8.03%.

After tuning the Prophet model’s fundamental parameters (changepoint prior scale=0.05, seasonality prior scale=1.0 and seasonality mode=multiplicative), the model improved the performance prediction, which is illustrated in Figure 9 below.

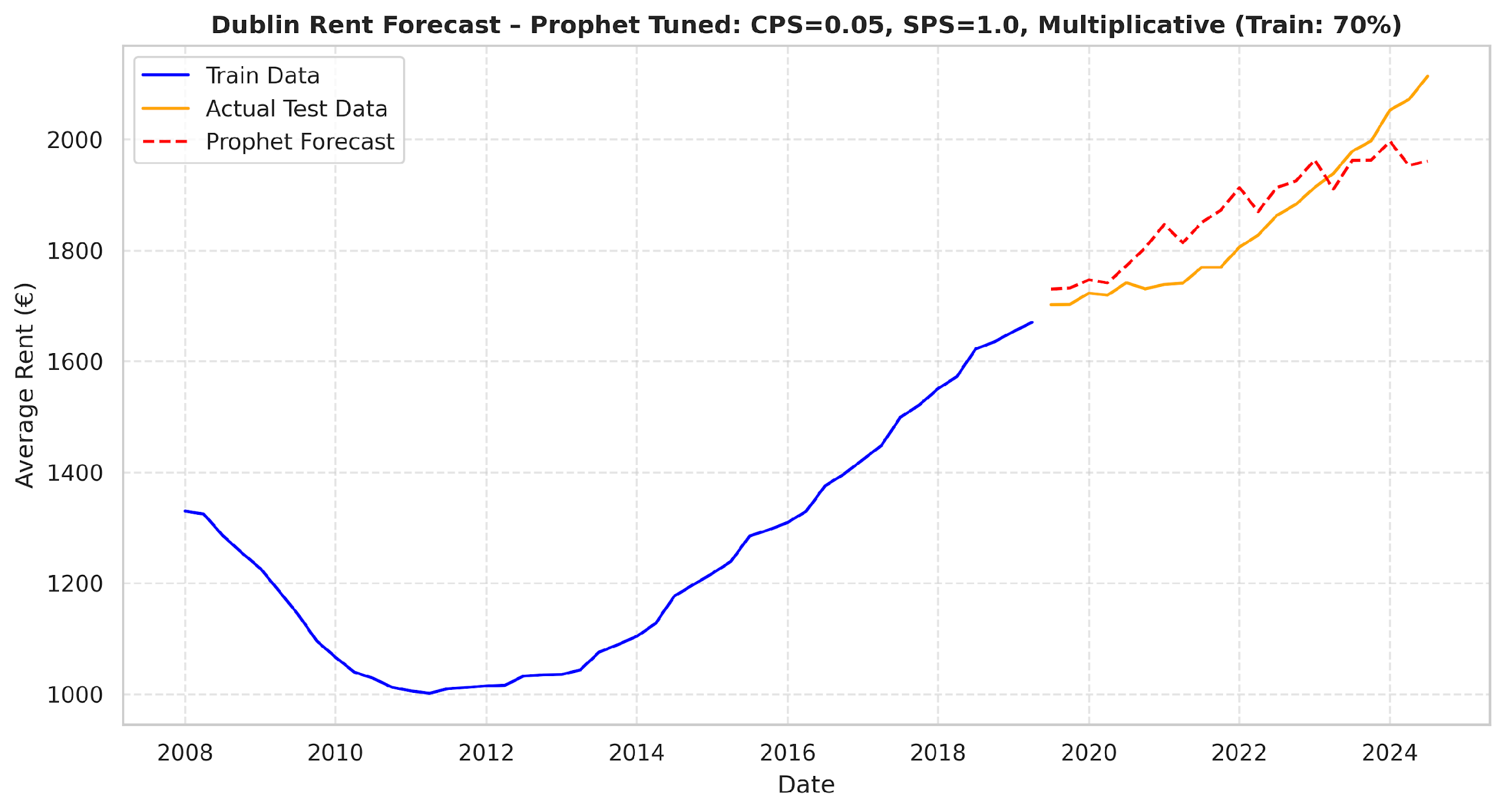


Figure 9: Prophet Dublin rent prices forecasting with tuned settings (Source: Author's analysis)

The tuned Prophet model was more accurately able to capture the curve and rate of rise in Dublin rent price series. Moreover, the evaluation metrics also improved substantially (Table 8):

**Table 8. Evaluation Metrics: Prophet Forecast with tuned settings**

| **Metric** | **Value** |
| --- | --- |
| MSE | 5066.18 |
| MAE | 60.61 |
| R² | 0.70 |
| MAPE | 3.26% |

(Source: Author's analysis)

The R² value of the tuned Prophet model increased to 0.70 outperforming its default version. The MSE decreased to 5066.18, and the MAE dropped to 60.61, both indicating improved predictive accuracy. Furthermore, the MAPE was reduced to 3.26%, corresponding to a better forecasting accuracy.

**Comparison Summary**

As the evaluation findings in Table 9 demonstrate, the Prophet model's performance was much improved by hyperparameter tuning approach. By altering the changepoint and seasonality prior scales and enabling multiplicative seasonality, the Prophet model not only decreased error metrics and improved the R² value but also decreased the MAPE from 8.03% to 3.26%. This reflected a notable improvement in forecast precision achieved through parameter tuning.

**Table 9. Evaluation Metrics: Prophet Comparison Summary**

| **Prophet type** | **MSE** | **MAE** | **R²** | **MAPE** |
| --- | --- | --- | --- | --- |
| Default Prophet | 26324.16 | 150.12 | - 0.56 | 8.03% |
| Tuned Prophet | 5066.18 | 60.61 | 0.70 | 3.26% |

(Source: Author's analysis)

## 4.3 LLMs: zero and few-shot prediction – Evaluation Results

### 4.3.1 ChatGPT-3.5

Firstly, ChatGPT-3.5 version was applied in zero- and few-shot prediction on Dublin rent prices. The prediction performance obtained from ChatGPT-3.5 in zero- and one-shot are shown in Figure 10. The zero-shot forecast, which was predicted without any contextual examples, drastically overestimated the future rent prices. Looking at the one-shot forecast, which incorporated only one in-context example, showed a slight movement to the real prices in the test dataset. However, the forecast followed a simplistic linear upward trend. Both, zero- and one-shot predictions showed a lack of regard to the fundamental seasonal pattern presented under the historical data.

Figure 10: 0- (A) and 1-shot (B) Dublin rent prices forecasting using ChatGPT-3.5 (Source: Author's analysis)

Moving to the 5- and 10-shot prediction from ChatGPT-3.5, shown in the Figure 11 below, there was a clear improvement in the prediction performance. Increasing the number of in-context examples facilitated the model’s alignments with the real rent prices in comparison to the zero- and one-shot forecasting. The 10-shot forecast closely followed the actual rent values and obtained better accuracy, particularly in the later quarters of the test period. Nonetheless, both forecasts remained devoid of seasonal awareness.

Figure 11: 5- (A) and 10-shot (B) Dublin rent prices forecasting using ChatGPT-3.5 (Source: Author's analysis)

Thereafter, 20- and 30-shot using ChatGPT-3.5 were applied to forecast the Dublin rent prices. As illustrated in Figure 12 both prediction results demonstrated decreases in the accuracy, even though there were more in-context examples. The 20-shot prediction continued to demonstrate the overall growing trend, though it increasingly overestimated rent prices beyond the test set's actual trajectory. To be more precise, predicted values were curving upward more steeply than the observed values.

In contrast, the 30-shot forecast appeared closer to the actual rent levels. This demonstrated that increasing the number of shots did not ensure better forecasting results.

Figure 12: 20- (A) and 30-shot (B) Dublin Rent Prices Forecasting using ChatGPT-3.5 (Source: Author's analysis)

**Comparison Summary**

The evaluation metrics for ChatGPT-3.5 at different shot levels are displayed in Table 10. The findings demonstrated that switching from zero-shot to few-shot learning considerably improved predicting accuracy. The R² improved from an extremely negative value -13.12 (0-shot) to 0.89 (30-shot). The MSE and MAE values decreased from 238050.59 and 452.84 in

the 0-shot to 1848.15 and 35.91, respectively, in the 30-shot levels. Similarly, the MAPE value decreased from 24.03% to 1.96%.

The 30-shot configuration, which produced the lowest MSE of 1848.15, lowest MAE of 35.91, lowest MAPE of 1.96%, and greatest R² of 0.89, surprisingly provided the best overall performance.

**Table 10. Evaluation Metrics: Zero and Few-shot Prediction using ChatGPT-3.5**

| **Shot Level** | **MSE** | **MAE** | **R²** | **MAPE** |
| --- | --- | --- | --- | --- |
| 0-shot | 238050.59 | 452.84 | - 13.12 | 24.03% |
| 1-shot | 27894.82 | 153.10 | - 0.65 | 8.16% |
| 5-shot | 2005.62 | 37.24 | 0.88 | 2.03% |
| 10-shot | 1962.54 | 36.81 | 0.88 | 2.02% |
| 20-shot | 6942.54 | 74.40 | 0.58 | 3.98% |
| 30-shot | 1848.15 | 35.91 | 0.89 | 1.96% |

(Source: Author's analysis)

These results supported the visual interpretation: adding more in-context examples could increase the prediction accuracy. However, consistent and repeatable predictions were only seen at the 0-shot, 1-shot, and 5-shot levels, despite the temperature parameter being set to 0 (to guarantee deterministic output).

### 4.3.2 ChatGPT-4-turbo

Secondly, ChatGPT-4-turbo version was applied in zero- and few-shot rental prices prediction in the Dublin dataset.

Figure 13 illustrates that both, zero and one-shot predictions using ChatGPT-4-turbo followed a similar pattern to those forecasted by ChatGPT-3.5. The plot A showed that zero-shot prediction overestimated future rent prices, which created a linear upward trend disconnected from the actual test data. This type of forecast was happening due to the lack of any contextual information. Regarding the one-shot prediction, the alignment slightly improved, although the linear upward bias was still present.

Figure 13: 0- (A) and 1-shot (B) Dublin rent prices forecasting using ChatGPT-4-turbo (Source: Author's analysis)

Subsequently, the 5-shot and 10-shot scenarios using ChatGPT-4-turbo were applied. The forecasting results are shown in Figure 14. The 5- and 10-shot prediction also improved, similarly to when ChatGPT-3.5 was applied. The forecasts were closer to the actual prices in test data. Nevertheless, they remained linear in nature and failed to adjust for the short-term fluctuations or underlying seasonality shown in the historical data.

Figure 14: 5- (A) and 10-shot (B) Dublin rent prices forecasting using ChatGPT-4-turbo (Source: Author's analysis)

Figure 15 below represents forecasts of 20- and 30-shot prediction using ChatGPT-4- turbo. Whereas at first both forecasts the 20-shot forecast seemed to be more in accordance with the overall upward trend in rent prices, it started to overpredict the prices in quarters after 2020.

This behaviour partially mirrored the forecast trend observed with GPT-3.5 at higher shot scenarios, where ChatGPT-4-turbo demonstrated improved alignment at 30-shot level.

Figure 15: 20- (A) and 30-shot (B) Dublin rent prices forecasting using ChatGPT-4-turbo (Source: Author's analysis)

**Comparison Summary**

Table 11 summarises the evaluation metrics for Chatchpt-4-turbo across all shot experiments. Similar to ChatGPT-3.5 forecasts, the zero and one-shot predictions performed poorly, with high MSE and MAE values and negative R² scores. This states that ChatGPT-4-turbo failed to capture important patterns in the data without and with one contextual example.

Notably, the performance of ChatGPT-4-turbo improved at the 5-shot and 10-shot levels. At the 5-shot level, the performance achieved the highest R² of 0.87 and lowest MSE of 2132.71. The 10-shot configuration in the opposite, performed well with the higher MAPE of 1.94% and the lowest MAE of 37.19 among all runs, indicating that the performance of ChatGPT-4-turbo was stronger within this mid-shot range.

**Table 11. Evaluation Metrics: Zero and Few-shot Prediction using ChatGPT-4-turbo**

| **Shot Level** | **MSE** | **MAE** | **R²** | **MAPE** |
| --- | --- | --- | --- | --- |
| 0-shot | 93716.78 | 302.84 | - 4.56 | 16.4% |
| 1-shot | 38325.06 | 178.66 | - 1.27 | 9.49% |
| 5-shot | 2132.71 | 38.89 | 0.87 | 2.12% |
| 10-shot | 2417.09 | 37.19 | 0.85 | 1.94% |
| 20-shot | 7651.71 | 77.74 | 0.55 | 4.18% |
| 30-shot | 4239.23 | 55.37 | 0.74 | 3.03% |

(Source: Author's analysis)

In addition, the forecasts produced by ChatGPT-4-turbo continued to lack seasonal changes, suggesting that it is not an appropriate model for temporal dependencies in time series data.

### 4.3.3 Time-LLM(GPT-2) – Evaluation Results

The forecast generated by Time-LLM(GPT-2) is illustrated in Figure 16 below.

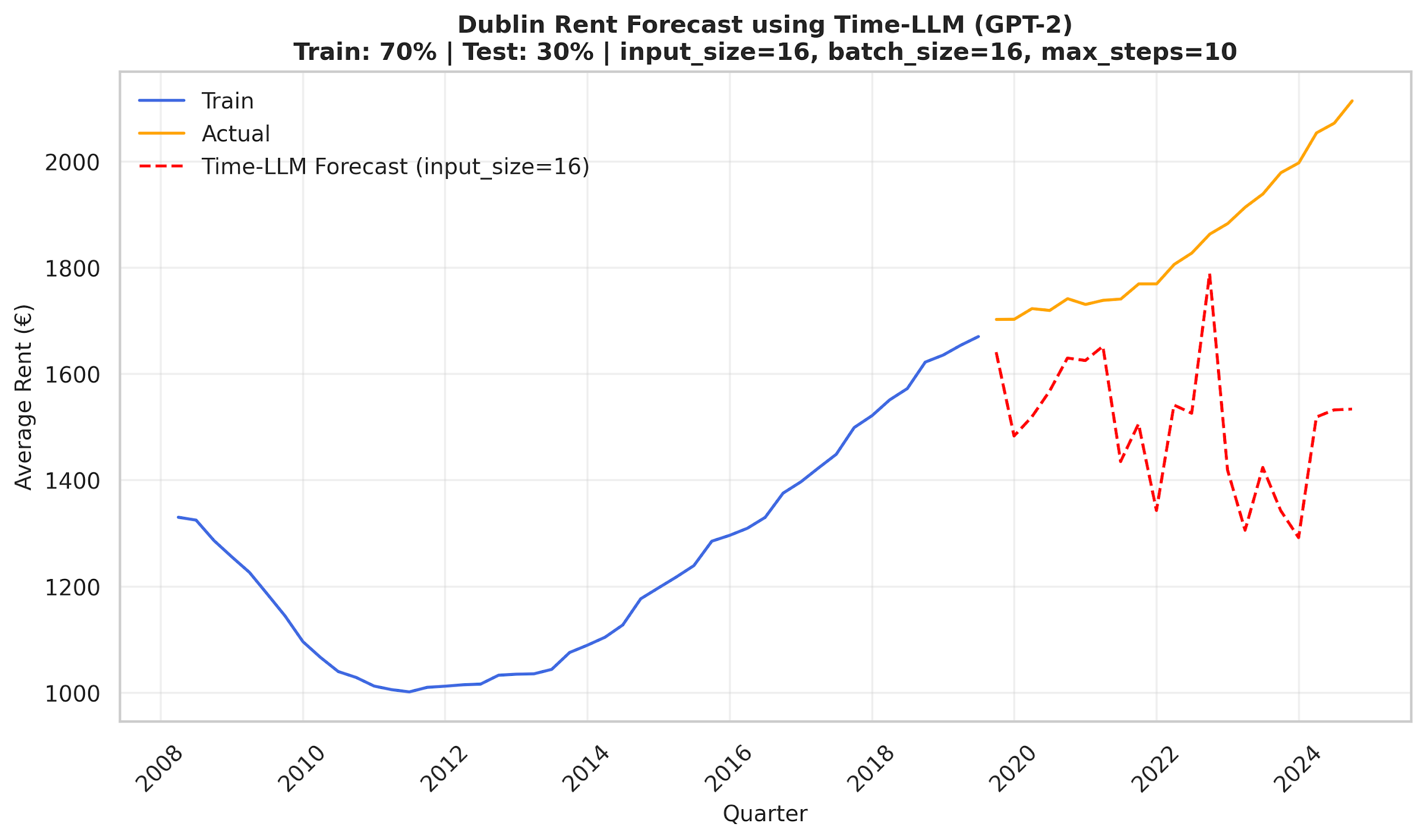


Figure 16: Dublin rent prices forecasting using Time-LLM (Source: Author's analysis)

The Time-LLM prediction seemed to be quite erratic and varies substantially from the actual price data. The plot demonstrated that the predicted values fluctuated randomly, following a downward trend across the test period. The prediction by Time-LLM (GPT-2) failed to acquire the underlying trend or seasonal progression.

Table 12 shows that the Time-LLM(GPT-2) achieved a high MSE of 158333.49 and MAE of 340.96, as well as a significantly negative R² score of -8.39. Moreover, the MAPE reached 17.92%, indicating a relatively high average forecasting error. These results confirm the poor prediction accuracy observed in Figure 16.

**Table 12. Evaluation Metrics: Time-LLM (input size=16, batch size=16)**

| **Metric** | **Value** |
| --- | --- |
| MSE | 158333.49 |
| MAE | 340.96 |
| R² | - 8.39 |
| MAPE | 17.92% |

(Source: Author's analysis)

The performance after tuning Time-LLM(GPT-2) is illustrated in Figure 17. Compared to the previous model with input size = 16, this configuration with the input size = 8 exhibited reduced volatility across the test period. The plot demonstrates that the model still has issues to extrapolate valuable long-term trends from shorter historical windows. Although the input size tuning process reduced enormous spikes, it did not sufficiently improve the model’s capacity in tracking the overall growth pattern of Dublin rent prices.

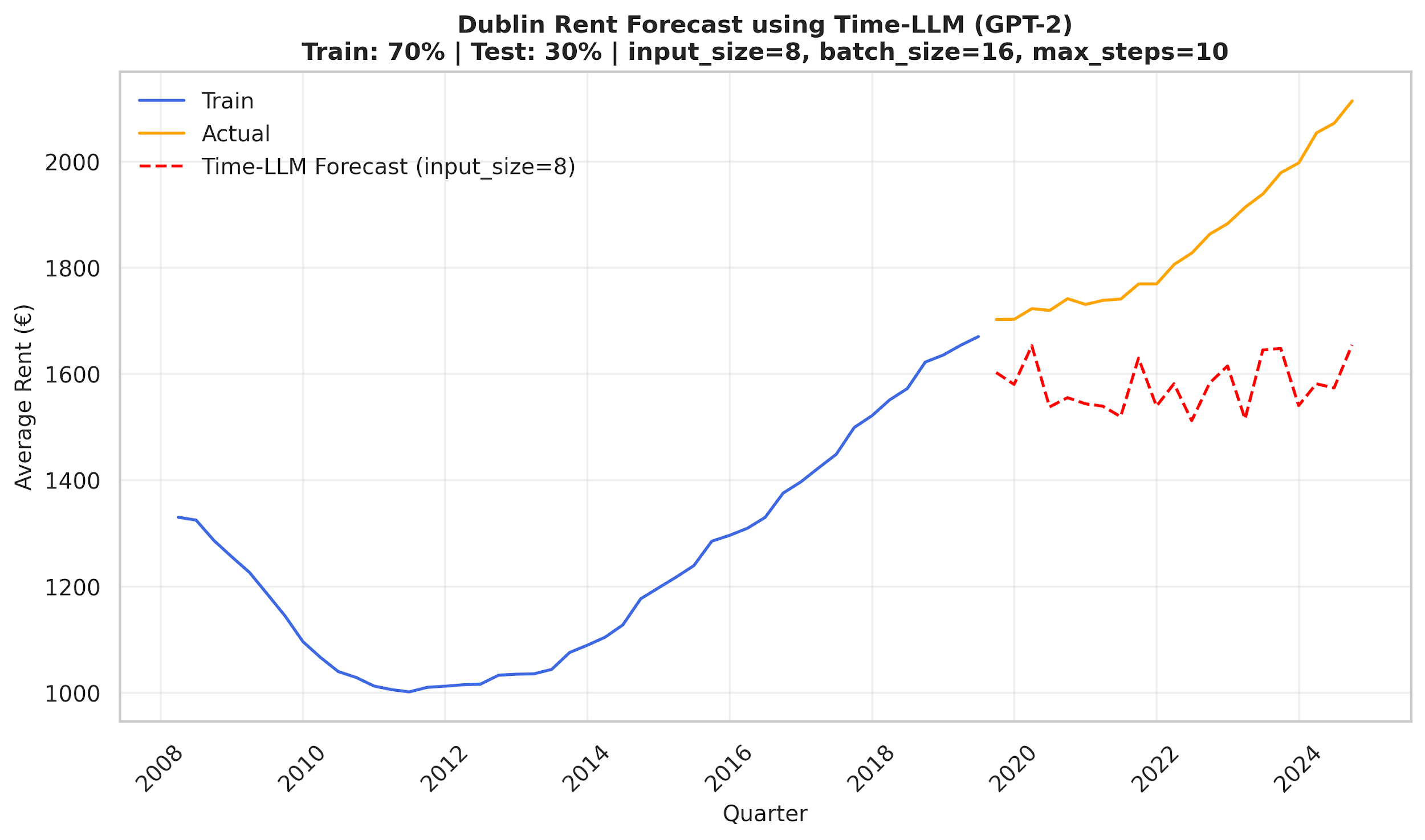


Figure 17: Dublin rent prices forecasting using Time-LLM after tuning (Source: Author's analysis)

Table 13 indicates that tuning Time-LLM(GPT-2) causes a modest improvement of the performance.The MSE dropped to 87621.73, the MAE decreased to 268.40, the MAPE improved to 14.17%. Similarly, the R² value improved, though it persists negative at -4.20, indicating continued poor predictive accuracy.

**Table 13. Evaluation Metrics: Time-LLM (input size=8, batch size=16) after tuning**

| **Metric** | **Value** |
| --- | --- |
| MSE | 87621.73 |
| MAE | 268.40 |
| R² | - 4.20 |
| MAPE | 14.17% |

(Source: Author's analysis)

**Comparison Summary**

Table 14 summarises both performance metrics of Time-LLM(GPT-2) and tuned Time- LLM(GPT-2). The forecasting performance of Time-LLM(GPT-2) continued to be weak overall, though both the MSE, MAE and MAPE values were reduced and R² was improved.

**Table 14. Evaluation Metrics: Time-LLM Comparison Summary**

| **Time-LLM** | **MSE** | **MAE** | **R²** | **MAPE** |
| --- | --- | --- | --- | --- |
| input size=16,  batch size=16 | 158333.49 | 340.96 | - 8.39 | 17.92% |
| input size=8,  batch size=16 | 87621.73 | 268.40 | - 4.20 | 14.17% |

(Source: Author's analysis)

The results with the negative R² values in both configurations demonstrated that Time- LLM(GPT-2) fails to outperform even a naive mean predictor. This further highlights the drawbacks of employing a general-purpose language model, such as GPT-2, for structured time series forecasting, especially when the model is devoid of temporal smoothing or domain-specific fine-tuning.

### 4.3.4 Chronos – Evaluation Results

To control the randomness and eliminate the nondeterministic behaviour of Chronos models each time the application is run, the torch.manual\_seed was applied and set to be 42.

* *Chronos T5-small evaluation*

Figure 18 below presents Dublin rent prices forecast using the Chronos T5-small model. Looking at the plot, the green line represents a median forecast trajectory predicted by the Chronos T5-small model, while the green shaded area indicates the 70% prediction interval, representing uncertainty over time. The forecast initially closely matched the actual prices in the test data, yet it quickly deviated, underestimating the test's upward trend. Additionally, the prediction interval grown considerably over time, meaning that uncertainty rises with the length of the forecast horizon. The Chronos T5-small model struggled to maintain long-term trend fidelity, producing an eventually declining forecast.

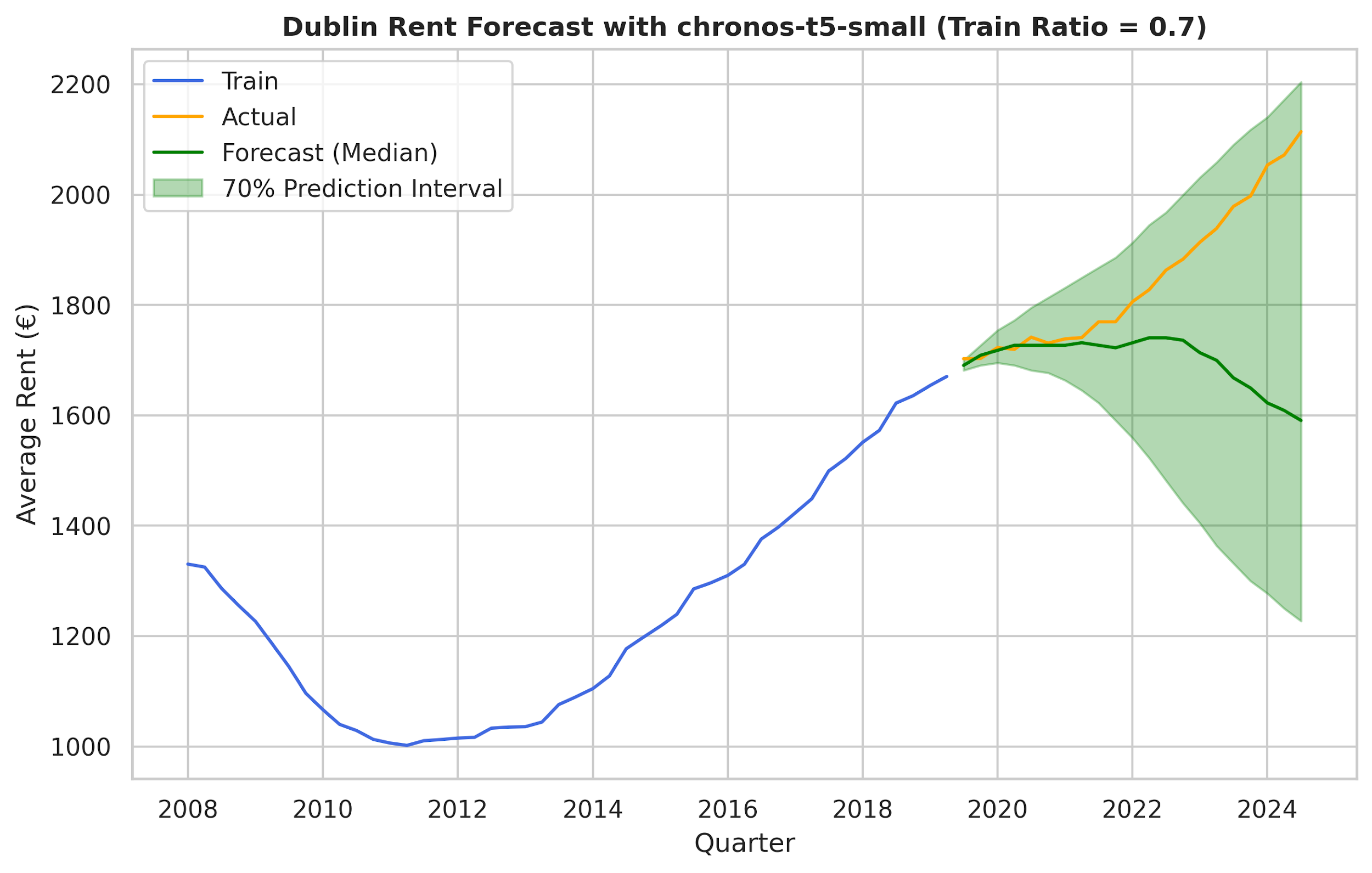


Figure 18: Dublin rent prices forecasting using Chronos (T5-small) (Source: Author's analysis)

Table 15 below shows the evaluation metrics of the Chronos T5-small model. The MSE and MAE values were 49719.94 and 147.97, respectively. The MAPE was 7.45%, corresponding to a moderate level of average prediction error. However, the R² score was negative of -1.95, which shows that the model failed to outperform a naive mean predictor.

**Table 15. Evaluation Metrics: Chronos (T5-small)**

| **Metric** | **Value** |
| --- | --- |
| MSE | 49719.94 |
| MAE | 147.97 |
| R² | - 1.95 |
| MAPE | 7.45% |

(Source: Author's analysis)

From both the plot and evaluation metrics, the Chronos T5-small model captures some short-term structure; it ultimately underfits the underlying increasing trend in the rent price data.

* *Chronos T5-base evaluation*

The forecasting results of Chronos T5-base model are illustrated in Figure 19. The median forecast followed the actual rent prices more closely in the earlier quarters of the test, similar to the T5-small model. Unlike the T5-small, the T5-base model showed better alignment with the actual test’s increasing trend, though it eventually underestimated the high values. Concerning the prediction interval, it was slightly narrower, indicating a small confidence improvement in the forecast.

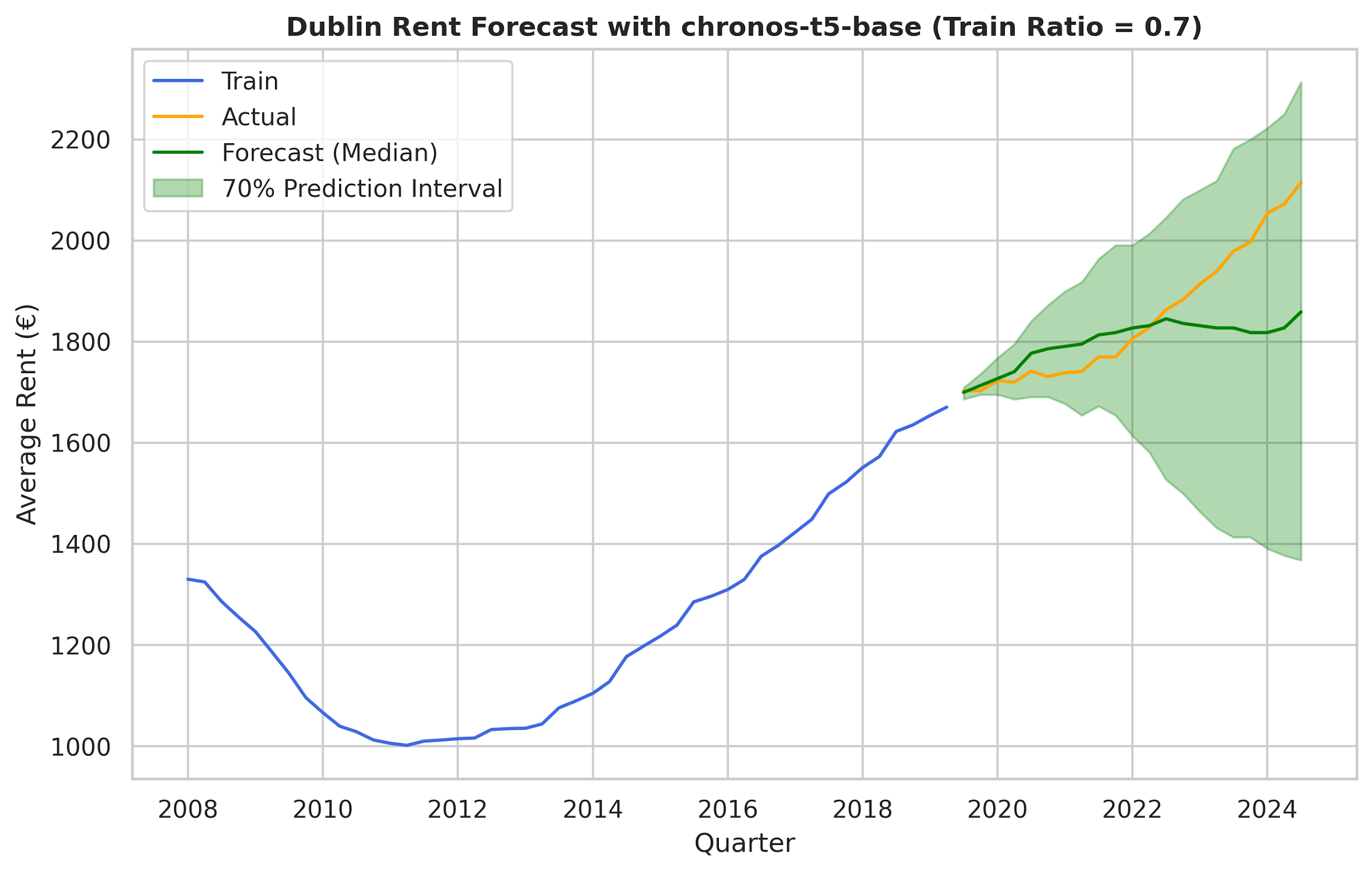


Figure 19: Dublin rent prices forecasting using Chronos (T5-base) (Source: Author's analysis)

Table 16 demonstrates that the Chronos T5-base model achieved an MSE of 13009.31 and MAE of 79.94, notably outperforming the T5-small model in terms of performance. In addition, the MAPE value decreased to 4.08%, leading to improved forecasting accuracy and stronger overall model performance. The R² attained a positive value of 0.23, meaning that the model starts to explain some of the variations in the real rent prices.

**Table 16. Evaluation Metrics: Chronos (T5-base)**

| **Metric** | **Value** |
| --- | --- |
| MSE | 13009.31 |
| MAE | 79.94 |
| R² | 0.23 |
| MAPE | 4.08% |

(Source: Author's analysis)

* *Chronos T5-large evaluation*

The Dublin rent price prediction produced by Chronos T5-large model is displayed in Figure 20. The median forecast of this model closely followed the actual trend of the test data, including the rising trajectory. The prediction interval was relatively narrow in the first quarters yet gradually widened with time to show growing uncertainty with longer-range forecasts. Compared to the smaller T5-base and T5-small versions, T5-large showed improved forecasting performance on the Dublin rent price dataset.

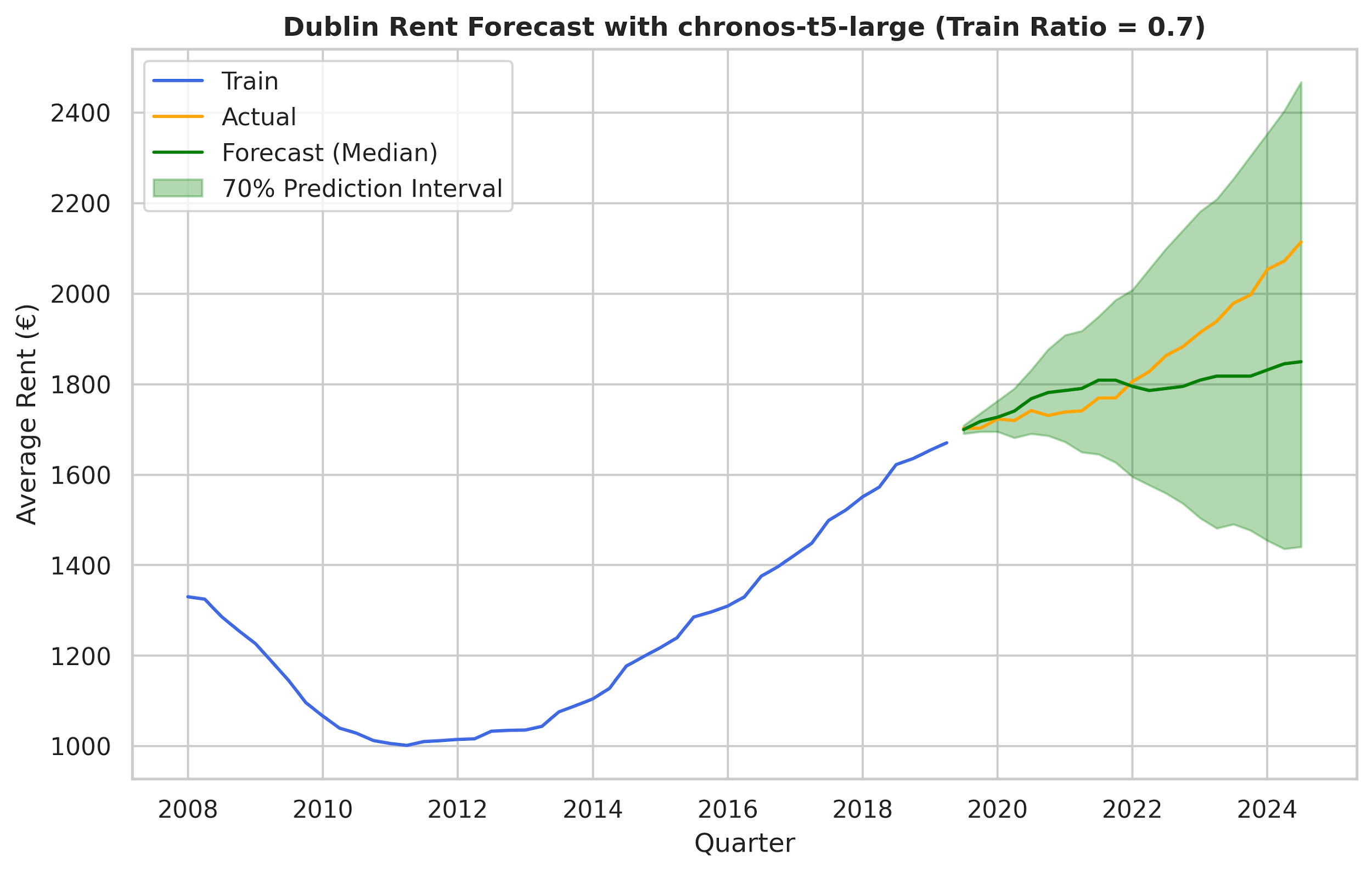


Figure 20: Dublin rent prices forecasting using Chronos (T5-large) (Source: Author's analysis)

As provided in Table 17, the Chronos T5-large model achieved the MSE = 13389.36, MAE = 85.17 and MAPE = 4.39%. According to the R² score, the T5-large model was able to explain 21% of the variance in the actual rent prices.

**Table 17. Evaluation Metrics: Chronos (T5-large)**

| **Metric** | **Value** |
| --- | --- |
| MSE | 13389.36 |
| MAE | 85.17 |
| R² | 0.21 |
| MAPE | 4.39% |

(Source: Author's analysis)

**Comparison Summary**

Forecasting accuracy across Chronos T5 versions as model size increases highlighted in Table 18. The T5-small model performed poorly, with the high MSE and MAE, and the negative R² score (-1.95). The T5-large model predicted with a marked improvement, achieving a positive R² score (0.21) and the MAE of 85.17. Lastly, the T5-base model showed the greatest improvement across all the Chronos models, recording the R² of 0.21 and MAPE of 4.08%, along with the lowest MSE and MAE. These findings demonstrate that T5-base offers the most feasible balance between predicting accuracy and model capacity.

**Table 18. Evaluation Metrics: Chronos Comparison Summary**

| **Chronos Model** | **MSE** | **MAE** | **R²** | **MAPE** |
| --- | --- | --- | --- | --- |
| T5-small | 49719.94 | 147.97 | -1.95 | 7.45% |
| T5-base | 13009.31 | 79.94 | 0.23 | 4.08% |
| T5-large | 13389.36 | 85.17 | 0.21 | 4.39% |

(Source: Author's analysis)

It was observed that the Chronos models perform particularly well in short-term forecasting scenarios, where the prediction interval is narrow (Figures 18, 19 and 20). To investigate it further, an additional experiment was implemented with the following split: 90% train and 10% test. This configuration will simulate a short-term prediction task and additionally expand the training dataset.

Figure 21 illustrates that forecasting performance of all three models T5-small, T5-base and T5-large improved, aligning with actual rent prices under the 90:10 split setup. It is visible how the prediction intervals narrowed dramatically, and the median forecasts followed the upward trend more closely.



Figure 21: Additional experiment using Chronos (T5-small (A), T5-base (B), T5-large (C)) 0.9 split to predict Dublin rent prices (Source: Author's analysis)

The evaluation results presented in Table 19 also demonstrate that all the Chronos models showed notable improvements in the forecasting performance under a 90:10 train-test split. Applying a shorter prediction horizon, both the MSE and MAE decreased remarkably across the board, and R² values improved, suggesting a strong fit to the actual rent prices in test data. Besides, the MAPE values decreased considerably, indicating improved forecast precision and higher the overall accuracy. The T5-large model achieved the remarkably low MSE of 629.98, MAE of 18.71, and a high R² of 0.86, indicating great explanatory power. It also scored the lowest MAPE of 0.91%. These results confirm the Chronos models can greatly benefit from an expanded training set and the remarkable capacity of the T5-large model to forecast in short-term scenarios when more historical data is available.

**Table 19. Evaluation Metrics: Chronos (0.9 split) Comparison Summary**

| **Chronos Model** | **MSE** | **MAE** | **R²** | **MAPE** |
| --- | --- | --- | --- | --- |
| T5-small | 2259.73 | 39.85 | 0.51 | 1.94% |
| T5-base | 3254.84 | 46.27 | 0.29 | 2.25% |
| T5-large | 629.98 | 18.71 | 0.86 | 0.91% |

(Source: Author's analysis)

### 4.3.5 Evaluation Results and Model Comparison

Table 20 illustrates the comparative performance of the best-performing configurations from each forecasting model based on MSE, MAE, R², and MAPE results.

**Table 20. Evaluation Metrics: Final Forecasting Model Comparison**

| **Forecasting Model** | **MSE** | **MAE** | **R²** | **MAPE** |
| --- | --- | --- | --- | --- |
| SARIMA | 3274.39 | 49.95 | 0.80 | 2.74% |
| Prophet | 5066.18 | 60.61 | 0.70 | 3.26% |
| 30-shot (ChatGPT-3.5) | 1848.15 | 35.91 | 0.89 | 1.96% |
| 10-shot (ChatGPT-4-turbo) | 2417.09 | 37.19 | 0.85 | 1.94% |
| Time-LLM (GhatGPT2) | 87621.73 | 268.40 | - 4.20 | 14.17% |
| Chronos T5-base | 13009.31 | 79.94 | 0.23 | 4.08% |

(Source: Author's analysis)

The results indicated that the 10-shot ChatGPT-3.5 model achieved the strongest quantitative performance, having the lowest MSE of 1848.15, lowest MAE of 35.91, highest R² of 0.89, and the lowest MAPE of 1.96%. Similarly, ChatGPT-4-turbo (10-shot) also demonstrated strong forecasting capability. However, as was previously mentioned qualitative analysis revealed that LLM-based forecasts produced overly smooth and unrealistic linear upward trends, failing to reflect seasonal fluctuations present in the actual dataset.

The Time-LLM (ChatGPT2) demonstrated the weakest performance among all the models that were evaluated, with the highest MSE of 87621.73, the largest MAE of 268.40, and the strongly negative R² of - 4.20. This performance highlighted that the model is the least effective forecasting model in this study, as it failed to capture the underlying patterns in the Dublin rental prices data.

The Chronos T5-base model, in contrast, performed rather well, outperforming Time-LLM but still lagging behind the SARIMA and Prophet models.

Regarding evaluation of the SARIMA model, it appeared to be the most accurate predicting techniques in this project. The strong R2 value of 0.80, the MSE of 3274.39, and the MAE of 49.95 showed that it could effectively capture both trend and seasonal components in the Dublin rental price data. Furthermore, the model's overall accuracy and stability was confirmed by the low average prediction error (MAPE of 2.74%).

# CONCLUSION

## 5.1 Summary of Key Findings

In this study, a comparative analysis was conducted to evaluate the forecasting accuracy of forecasting ML models against LLM-based approaches. The following LLM-based model including zero- and few-shot prediction using ChatGPT (3.5 and 4-turbo models), the Time-LLM (ChatGPT2) and the Chronos (T5-large, -base, -small) models were evaluated and compared to the classical SARIMA and Facebook Prophet model models. Despite that the ChatGPT few shot forecasting the lowest point-wise error metrics, the visual exploration of the forecast revealed that the ChatGPT failed to accurately capture the seasonality of the series, whereas the SARIMA model reproduced far more faithfully.

The Chronos model, by contrast, performed worse than the SARIMA and the Prophet models over the full test horizon, yet its first-years predictions matched the actual series much more closely than its longer-range forecast.

Lastly, the reprogramming framework Time-LLM (ChatGPT2) was the poorest forecasting model. The model was struggling to learn the basic upward trend in the Dublin rent data, despite the prompt being provided with the basic statistical values (the maximum, minimum and median values).

Based on the experimental results obtained using the quarterly Dublin rental price dataset, the LLMs underperformed both the SARIMA and Prophet model in the forecasting tasks. This research successfully achieved both the primary and secondary objectives by evaluating and comparing ML techniques and LLMs for rental price prediction in the Irish real estate market.

## 5.2 Limitations of the Study

### 5.2.1 Experimental Environment

In this research all forecasting models were developed, tested and evaluated using Python programming language and implemented in Jupyter Notebook. Although traditional forecasting models and LLM-based shot experiments were executed locally, attempts to run more complex models, such Time-LLM and Chronos, caused Jupyter kernel failures, most likely as a result of memory or execution-time limitations. Therefore, to address this challenge, the execution of Time-LLM and Chronos models was migrated to a cloud-based environment using Google Colab, which provides better memory allocation and compatibility with heavier workloads.

In regard to API authentication, it was handled using a secure.env file and the python-dotenv package to prevent credentials from being accessible in the codebase.

The hardware specifications are shown in Table 21, and the experiments were conducted on a macOS-based local machine.

**Table 21. Hardware Specifications**

| **Component** | **Specification** |
| --- | --- |
| Operating System | macOS Sequoia 15.1 |
| Processor (CPU) | 1.1 GHz Quad-Core Intel Core i5 |
| Memory (RAM) | 8 GB |
| Storage | 250 GB SSD |

(Source: Author's analysis)

### 5.2.2 Probabilistic Challenges

Throughout the experiment, one of the main limitations encountered was the presence of probabilistic issues arising from the use of the LLMs. Although the temperature parameter was set to 0 in the ChatGPT-3.5 and ChatGPT-4-turbo to stimulate deterministic outputs, non-reproducible replies occurred beyond the 5-shot level. Despite no changes in prompt structure or model parameters were applied, the higher shot levels provided different outcomes across multiple runs, whereas 0-shot, 1-shot, and 5-shot levels consistently produced similar forecasts. In contrast to deterministic forecasting, probabilistic forecasting not only provides point estimates but also captures the uncertainty inherent in predictions (Gneiting and Katzfuss, 2014). Even though both versions of ChatGPT were configured to act deterministically, the appearance of different outputs highlighted the possibility that these models may automatically act probabilistically in response to longer or complex prompts.

Similarly, the Chronos model exhibited a notable sensitivity to random seed values, which affected the forecast accuracy. This non-deterministic behaviour of the Chronos model may be amplified by the limited size of the dataset (Dodge et al., 2020; Zhou, Savova and Wang, 2025). The Chronos model generates each future step token-by-token by sampling from a probability distribution (Ansari et al., 2024). The sampling uses a pseudorandom-number generator and whenever the random seed is changed, the sequence of sampled tokens, and hence the entire prediction, changes.

## 5.3 Recommendations for Future Work

Future work could potentially be focused on expanding the dataset in order to reduce the variance that was encountered by the Chronos model and the Time-LLM.

Secondly, the new Chronos-bolt models that was released on HuggingFace can be tested on the time series prediction tasks. These new versions of Chronos-bolt can be tuned on the specific dataset trough AutoGluon-TimeSeries.

Additionally, to address the random seed sensitivity of the Chronos modes, the future research should explore seed-fixing strategies or multi-run averaging.

# References

Adedokun, S.A., 2020. *Housing Price Prediction and Classification Based on Crime Occurrence using Machine Learning Algorithms: Ireland* (Doctoral dissertation, Dublin, National College of Ireland).

Adetunji, A.B., Akande, O.N., Ajala, F.A., Oyewo, O., Akande, Y.F. and Oluwadara, G. (2022). House Price Prediction using Random Forest Machine Learning Technique. *Procedia Computer Science*, 199, pp.806–813.

doi:https://doi.org/10.1016/j.procs.2022.01.100.

Adhikari, R. and Agrawal, R.K., 2013. An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613*.

Albeladi, K., Zafar, B. and Mueen, A., 2023. Time series forecasting using LSTM and ARIMA. *International Journal of Advanced Computer Science and Applications*, *14*(1), pp.313-320.

Ansari, A.F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S.S., Arango, S.P., Kapoor, S. and Zschiegner, J., 2024. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*.

Apache (2004). *Apache License, Version 2.0*. Apache.org. Available at: https://www.apache.org/licenses/LICENSE-2.0.

Baron, M. and Karpinski, A., 2025. The Relevance of AWS Chronos: An Evaluation of Standard Methods for Time Series Forecasting with Limited Tuning. *arXiv preprint arXiv:2501.10216*.

Belete, D.M. and Huchaiah, M.D., 2022. Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *International Journal of Computers and Applications*, *44*(9), pp.875-886.

Beneduce, C., Lepri, B. and Luca, M., 2024. Large language models are zero-shot next location predictors. *arXiv preprint arXiv:2405.20962*.

Box, G.E.P. & Jenkins, G.M. (1976). *Time Series Analysis: Forecasting and Control*, Revised Edition, San Francisco: Holden Day.

Box, G.E.P., Jenkins, G.M. and Reinsel, G.C. (2008). Time Series Analysis. *Wiley Series in Probability and Statistics*. doi:https://doi.org/10.1002/9781118619193.

Box, G.E., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M., 2015. *Time series analysis: forecasting and control*. John Wiley & Sons.

Bramer, M., 2007. Avoiding overfitting of decision trees. *Principles of data mining*, pp.119-134.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S., 2020. Language models are few-shot learners. *Advances in neural information processing systems*, *33*, pp.1877-1901.

Brockwell, P.J. and Davis, R.A. eds., 2002. *Introduction to time series and forecasting*. New York, NY: Springer New York.

Bryman, A., 2016. *Social research methods*. Oxford university press.

Chamieh, I., Zesch, T. and Giebermann, K., 2024, June. Llms in short answer scoring: Limitations and promise of zero-shot and few-shot approaches. In *Proceedings of the 19th workshop on innovative use of nlp for building educational applications (bea 2024)* (pp. 309-315).

Chaturvedi, S., Rajasekar, E., Natarajan, S. and McCullen, N., 2022. A comparative assessment of SARIMA, LSTM RNN and Fb Prophet models to forecast total and peak monthly energy demand for India. *Energy Policy*, *168*, p.113097.

Chen, T. and Guestrin, C. (2016). XGBoost: a Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16*, pp.785–794. doi:https://doi.org/10.1145/2939672.2939785.

Chen, T. and Si, S., 2024. Predicting Rental Price of Lane Houses in Shanghai with Machine Learning Methods and Large Language Models. *arXiv preprint arXiv:2405.17505*.

Cheng, Q., Chen, L., Hu, Z., Tang, J., Xu, Q. and Ning, B., 2024. A novel prompting method for few-shot ner via llms. *Natural Language Processing Journal*, *8*, p.100099.

Chiarazzo, V., Caggiani, L., Marinelli, M. and Ottomanelli, M. (2014). A Neural Network based Model for Real Estate Price Estimation Considering Environmental Quality of Property Location. *Transportation Research Procedia*, 3, pp.810–817. doi:https://doi.org/10.1016/j.trpro.2014.10.067.

Chicco, D., Warrens, M.J. and Jurman, G., 2021. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *Peerj computer science*, *7*, p.e623.

Cook, R.D. (1977). Detection of Influential Observation in Linear Regression. *Technometrics*, 19(1), pp.15–18. doi:https://doi.org/10.1080/00401706.1977.10489493.

Crawford, G.W. and Fratantoni, M.C., 2003. Assessing the forecasting performance of regime‐switching, ARIMA and GARCH models of house prices. *Real Estate Economics*, *31*(2), pp.223-243.

Creative Commons (n.d.). *Attribution 4.0 International (CC BY 4.0) License*. Available at: https://creativecommons.org/licenses/by/4.0/deed.en.

Cryer, J.D. and Chan, K.-S. (2008). *Time Series Analysis*. *Springer Texts in Statistics*. New York, NY: Springer New York. doi:https://doi.org/10.1007/978-0-387-75959-3.

CSO *- central statistics office* (2019) *CSO*. Available at: https://www.cso.ie/en/ (Accessed: 13 February 2025).

CSO Copyright Policy *- CSO - Central Statistics Office*. Available at: https://www.cso.ie/en/aboutus/whoweare/copyrightpolicy/. (Accessed: 1 March 2025).

Das, K., Jiang, J. and Rao, J.N.K., 2004. Mean squared error of empirical predictor.

Davis, J., Van Bulck, L., Durieux, B.N. and Lindvall, C., 2024. The temperature feature of ChatGPT: modifying creativity for clinical research. *JMIR human factors*, *11*(1), p.e53559.

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.

De Zarzà, I., De Curtò, J., Roig, G. and Calafate, C.T. (2023). LLM Multimodal Traffic Accident Forecasting. *Sensors*, 23(22), p.9225. doi:https://doi.org/10.3390/s23229225.

Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H. and Smith, N., 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

European Union. (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation).* Official Journal of the European Union, L119, pp.1–88.

García, S., Luengo, J. and Herrera, F., 2015. *Data preprocessing in data mining* (Vol. 72, pp. 59-139). Cham, Switzerland: Springer International Publishing.

Garlapati, A., Krishna, D.R., Garlapati, K., Rahul, U. and Narayanan, G., 2021, April. Stock price prediction using Facebook Prophet and Arima models. In *2021 6th International Conference for Convergence in Technology (I2CT)* (pp. 1-7). IEEE.

Ghanad, A., 2023. An overview of quantitative research methods. *International journal of multidisciplinary research and analysis*, *6*(08), pp.3794-3803.

Gneiting, T. and Katzfuss, M., 2014. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, *1*(1), pp.125-151.

Gupta, R. and Miller, S.M., 2012. The time-series properties of house prices: A case study of the Southern California market. *The Journal of Real Estate Finance and Economics*, *44*, pp.339-361.

Gruver, N., Finzi, M., Qiu, S. and Wilson, A.G., 2023. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, *36*, pp.19622-19635.

Heidari, M. and Rafatirad, S. (2020). Bidirectional Transformer based on online Text-based information to Implement Convolutional Neural Network Model For Secure Business Investment. *2020 IEEE International Symposium on Technology and Society (ISTAS)*. doi:https://doi.org/10.1109/istas50296.2020.9462170.

Ho, T.K., 1995, August. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278-282). IEEE. doi:https://doi.org/10.1109/ICDAR.1995.598994.

Hodson, T.O., 2022. Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development Discussions*, *2022*, pp.1-10.

Hossain, R. and Timmer, D., 2021. Machine learning model optimization with hyper parameter tuning approach. *Glob. J. Comput. Sci. Technol. D Neural Artif. Intell*, *21*(2), p.31.

Hyndman, R.J. and Athanasopoulos, G., 2018. *Forecasting: principles and practice*. OTexts.

Hyndman, R.J., & Athanasopoulos, G. (2021) *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on 26 April 2025.

Jadon, S., Milczek, J.K. and Patankar, A., 2021. Challenges and approaches to time-series forecasting in data center telemetry: A survey. *arXiv preprint arXiv:2101.04224*.

Jiang, Z. and Shen, G. (2019). Prediction of House Price Based on The Back Propagation Neural Network in The Keras Deep Learning Framework. *2019 6th International Conference on Systems and Informatics (ICSAI)*. doi:https://doi.org/10.1109/icsai48974.2019.9010071.

Jin, M., Wang, S., Ma, L., Chu, Z., Zhang, J.Y., Shi, X., Chen, P.-Y., Liang, Y., Li, Y., Pan, S. and Wen, Q. (2023). Time-LLM: Time Series Forecasting by Reprogramming Large Language Models. *arXiv (Cornell University)*. doi:https://doi.org/10.48550/arxiv.2310.01728.

Jin, M., Zhang, Y., Chen, W., Zhang, K., Liang, Y., Yang, B., Wang, J., Pan, S. and Wen, Q., 2024. Position paper: What can large language models tell us about time series analysis. *arXiv e-prints*, pp.arXiv-2402.

Kaushik, S., Choudhury, A., Sheron, P.K., Dasgupta, N., Natarajan, S., Pickett, L.A. and Dutt, V., 2020. AI in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, *3*, p.4.

Kim, H., Kwon, Y. and Choi, Y. (2020). Assessing the Impact of Public Rental Housing on the Housing Prices in Proximity: Based on the Regional and Local Level of Price Prediction Models Using Long Short-Term Memory (LSTM). *Sustainability*, 12(18), p.7520. doi:https://doi.org/10.3390/su12187520.

Kim, Y., Xu, X., McDuff, D., Breazeal, C. and Park, H.W. (2024). Health-LLM: Large Language Models for Health Prediction via Wearable Sensor Data. *arXiv (Cornell University)*. doi:https://doi.org/10.48550/arxiv.2401.06866.

Kirchgässner, G., Wolters, J. and Hassler, U. (2012) *Introduction to modern Time Series analysis*. Springer Science & Business Media.

Kojima, T., Gu, S.S., Reid, M., Matsuo, Y. and Iwasawa, Y., 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, *35*, pp.22199-22213.

Lam, K.C., Yu, C.Y. and Lam, K.Y. (2008). An Artificial Neural Network and Entropy Model for Residential Property Price Forecasting in Hong Kong. *Journal of Property Research*, 25(4), pp.321–342. doi:https://doi.org/10.1080/09599910902837051.

Liashchynskyi, P. and Liashchynskyi, P., 2019. Grid search, random search, genetic algorithm: a big comparison for NAS. *arXiv preprint arXiv:1912.06059*.

Li, D.-Y., Xu, W., Zhao, H. and Chen, R.-Q. (2009). *A SVR based forecasting approach for real estate price prediction*. IEEE Xplore. doi:https://doi.org/10.1109/ICMLC.2009.5212389.

Lim, W.T., Wang, L., Wang, Y. and Chang, Q. (2016). Housing price prediction using neural networks. *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. doi:https://doi.org/10.1109/fskd.2016.7603227.

Manjula, R., Jain, S., Srivastava, S. and Rajiv Kher, P. (2017). Real estate value prediction using multivariate regression models. *IOP Conference Series: Materials Science and Engineering*, 263, p.042098. doi:https://doi.org/10.1088/1757-899x/263/4/042098.

Menke, W., 2018. *Geophysical data analysis: Discrete inverse theory*. Academic press.

Milenković, M., Švadlenka, L., Melichar, V., Bojović, N. and Avramović, Z. (2016). SARIMA modelling approach for railway passenger flow forecasting. *Transport*, pp.1–8. doi:https://doi.org/10.3846/16484142.2016.1139623.

Milunovich, G. (2020). Forecasting Australia’s Real House Price Index: A Comparison of Time Series and Machine Learning Methods. *Journal of Forecasting*. doi:https://doi.org/10.1002/for.2678.

Mirg, V., 2022. *Prediction of Property Prices of Dublin Housing Market using Ensemble Learning* (Doctoral dissertation, Dublin, National College of Ireland).

Mushtaq, R. (2011). Augmented Dickey Fuller Test. *SSRN Electronic Journal*. doi:https://doi.org/10.2139/ssrn.1911068.

​​Nelson, B.K., 1998. Time series analysis using autoregressive integrated moving average (ARIMA) models. *Academic emergency medicine*, *5*(7), pp.739-744.

Ogunsanya, M., Isichei, J. and Desai, S., 2023. Grid search hyperparameter tuning in additive manufacturing processes. *Manufacturing Letters*, *35*, pp.1031-1042.

OpenAI (2024). *OpenAI API*. Available at: https://platform.openai.com/docs/models.

Özöğür Akyüz, S., Eygi Erdogan, B., Yıldız, Ö. and Karadayı Ataş, P. (2022). A Novel Hybrid House Price Prediction Model. *Computational Economics*.

doi:https://doi.org/10.1007/s10614-022-10298-8.

Pal, A. and Prakash, P. (2017) *Practical Time Series analysis: Master Time Series Data Processing, Visualization, and Modeling using Python*. Packt Publishing Ltd.

Palma, W., 2016. *Time series analysis*. John Wiley & Sons.

Panahandeh, A., Rabiei-Dastjerdi, H., Goktas, P. and McArdle, G. (2024). Answering new urban questions: Using eXplainable AI-driven analysis to identify determinants of Airbnb price in Dublin. *Expert Systems with Applications*, 260, p.125360. doi:https://doi.org/10.1016/j.eswa.2024.125360.

Perez, E., Kiela, D. and Cho, K., 2021. True few-shot learning with language models. *Advances in neural information processing systems*, *34*, pp.11054-11070.

Pinter, G., Mosavi, A. and Felde, I. (2020). Artificial Intelligence for Modeling Real Estate Price Using Call Detail Records and Hybrid Machine Learning Approach. *Entropy*, 22(12), p.1421. doi:https://doi.org/10.3390/e22121421.

Prophet (no date). Available at: https://facebook.github.io/prophet/. (Accessed: 20 March 2025).

Pyle, D., 1999. *Data preparation for data mining.* San Francisco, Cal.: Morgan Kaufmann.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., 2019. Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), p.9.

Rawool, A.G., Rogye, D.V., Rane, S.G. and Bharadi, V.A., 2021. House price prediction using machine learning. *IRE Journals*, 4(11), pp.29-33. Available at: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.irejournals.com/formatedpaper/1702692.pdf. (Accessed 22 October 2024)

Rosen, S., 1974. Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of political economy*, *82*(1), pp.34-55. Available at: https://www.journals.uchicago.edu/doi/abs/10.1086/260169. (Accessed 27 October 2024)

Satrio, C.B.A., Darmawan, W., Nadia, B.U. and Hanafiah, N., 2021. Time series analysis and forecasting of coronavirus disease in Indonesia using ARIMA model and PROPHET. *Procedia Computer Science*, *179*, pp.524-532.

Sellam, Z.A., Distante, C., Taleb-Ahmed, A. and Mazzeo, P.L., 2025. Boosting house price estimations with Multi-Head Gated Attention. *Expert Systems with Applications*, *259*, p.125276. doi:https://doi.org/10.1016/j.eswa.2024.125276.

Shcherbakov, M.V., Brebels, A., Shcherbakova, N.L., Tyukov, A.P., Janovsky, T.A. and Kamaev, V.A.E., 2013. A survey of forecast error measures. *World applied sciences journal*, *24*(24), pp.171-176.

Taylor, S.J. and Letham, B., 2018. Forecasting at scale. *The American Statistician*, *72*(1), pp.37-45.

Tsay, R.S. (2010). *Analysis of financial time series*. Hoboken, New Jersey: John Wiley & Sons.

Tukey, J.W., 1977. *Exploratory data analysis* (Vol. 2, pp. 131-160). Reading, MA: Addison-wesley.

Vapnik, V., Golowich, S. and Smola, A., 1996. Support vector method for function approximation, regression estimation and signal processing. *Advances in neural information processing systems*, *9*.

Vinutha, H.P., Poornima, B. and Sagar, B.M., 2018. Detection of outliers using interquartile range technique from intrusion dataset. In *Information and decision sciences: Proceedings of the 6th international conference on ficta* (pp. 511-518). Springer Singapore.

Walfish, S., 2006. A review of statistical outlier methods. *Pharmaceutical technology*, *30*(11), p.82.

Wang, C. and Wu, H. (2018). A new machine learning approach to house price estimation. *New Trends in Mathematical Science*, 4(6), pp.165–171. doi:https://doi.org/10.20852/ntmsci.2018.327.

Wang, F., Zou, Y., Zhang, H. and Shi, H. (2019). *House Price Prediction Approach based on Deep Learning and ARIMA Model*. IEEE Xplore. doi:https://doi.org/10.1109/ICCSNT47585.2019.8962443.

Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y. and Zhao, W.X., 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, *18*(6), p.186345.

Wang, P.-Y., Chen, C.-T., Su, J.-W., Wang, T.-Y. and Huang, S.-H. (2021). Deep Learning Model for House Price Prediction Using Heterogeneous Data Analysis Along With Joint Self-Attention Mechanism. *IEEE Access*, 9, pp.55244–55259. doi:https://doi.org/10.1109/ACCESS.2021.3071306.

Wang, S., Ji, T., Wang, L., Sun, Y., Liu, S.C., Kumar, A. and Lu, C.T., 2024. Stocktime: A time series specialized large language model architecture for stock price prediction. *arXiv preprint arXiv:2409.08281*.

Wang, X., Wen, J., Zhang, Y. and Wang, Y. (2014). Real estate price forecasting based on SVM optimized by PSO. *Optik*, 125(3), pp.1439–1443. doi:https://doi.org/10.1016/j.ijleo.2013.09.017.

Whittle, P. (1951). *Hypothesis Testing in Time Series Analysis*.

Xue, H. and Salim, F.D., 2023. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, *36*(11), pp.6851-6864.

Yu, X., Chen, Z., Ling, Y., Dong, S., Liu, Z. and Lu, Y., 2023. Temporal data meets LLM--explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*.

Zhan, C., Wu, Z., Liu, Y., Xie, Z. and Chen, W., 2020, July. Housing prices prediction with deep learning: an application for the real estate market in Taiwan. In *2020 IEEE 18th International Conference on Industrial Informatics (INDIN)*(Vol. 1, pp. 719-724). IEEE.

Zhan, C., Liu, Y., Wu, Z., Zhao, M. and Tommy (2023). A hybrid machine learning framework for forecasting house price. *Expert Systems with Applications*, 233, pp.120981–120981. doi:https://doi.org/10.1016/j.eswa.2023.120981.

Zhang, X., Chowdhury, R.R., Gupta, R.K. and Shang, J., 2024. Large language models for time series: A survey. *arXiv preprint arXiv:2402.01801*.

Zhang, Z. (2021). Decision Trees for Objective House Price Prediction. *2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*. doi:https://doi.org/10.1109/mlbdbi54094.2021.00059.

Zhou, H., Savova, G. and Wang, L., 2025. Assessing the Macro and Micro Effects of Random Seeds on Fine-Tuning Large Language Models. *arXiv preprint arXiv:2503.07329*.

Zhou, T., Niu, P., Sun, L. and Jin, R., 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, *36*, pp.43322-43355.

# Appendices

# Appendix A: Summary of Previous Research on Real Estate Price Forecasting in Ireland

Table A1 presents briefly all the relevant research conducted in Ireland on real estate price prediction, accenting the algorithms used and key outcomes.

**Table A1. Summary of Previous works on Real Estate Prices Forecasting in Ireland**

| **Authors** | **Title** | **Algorithms** | **Results** |
| --- | --- | --- | --- |
| Adedokun (2020) | Housing Price Prediction and Classification Based on Crime Occurrence using Machine Learning Algorithms: Ireland | Generalized Linear Model, Ridge, Lasso, SVM,  Random Forest, C5.0,  K-Nearest Neighbours, Multinomial Logistic Regression | Random Forest achieved the lowest RMSE value of 0.0946 |
| Mirg (2022) | Prediction of Property Prices of Dublin Housing Market using Ensemble Learning | Multiple Linear Regression,  K Nearest Neighbour, Decision Tree Regressor,  Random Forest Regressor,  Gradient Boosting Regressor | Gradient Boosting Regression Model achieved an R-Square value of 75.22 |
| Panahandeh et al. (2023) | Answering new urban questions: Using eXplainable AI-driven analysis to identify determinants of Airbnb price in Dublin | Lasso,  Random Forest,  SVR,  XGBoost | R-2 score of XGBoost on testing data indicates 0.816 |
| Hurley and Sweeney (2024) | Irish Property Price Estimation Using a Flexible Geo-Spatial Smoothing Approach: What is the Impact of an Address? | Linear Regression,  K-Nearest Neighbours, Decision Trees, Random Forests and Generalized Additive Models (GAMs) | While Random Forests establish marginally better accuracy (0.88) in predicting mean values, they are less effective in terms of prediction interval coverage compared to Generalized Additive Models |

(Source: Author's analysis)

# Appendix B: Capstone Project Timeline

Gantt Chart representing chronological tasks based on the proposed secondary objectives shown in Figure B1. This Gantt chart breaks down the main tasks, milestones, and completion status for the capstone project "Machine Learning Techniques and Large Language Models for Property Rental Price Prediction in Ireland."

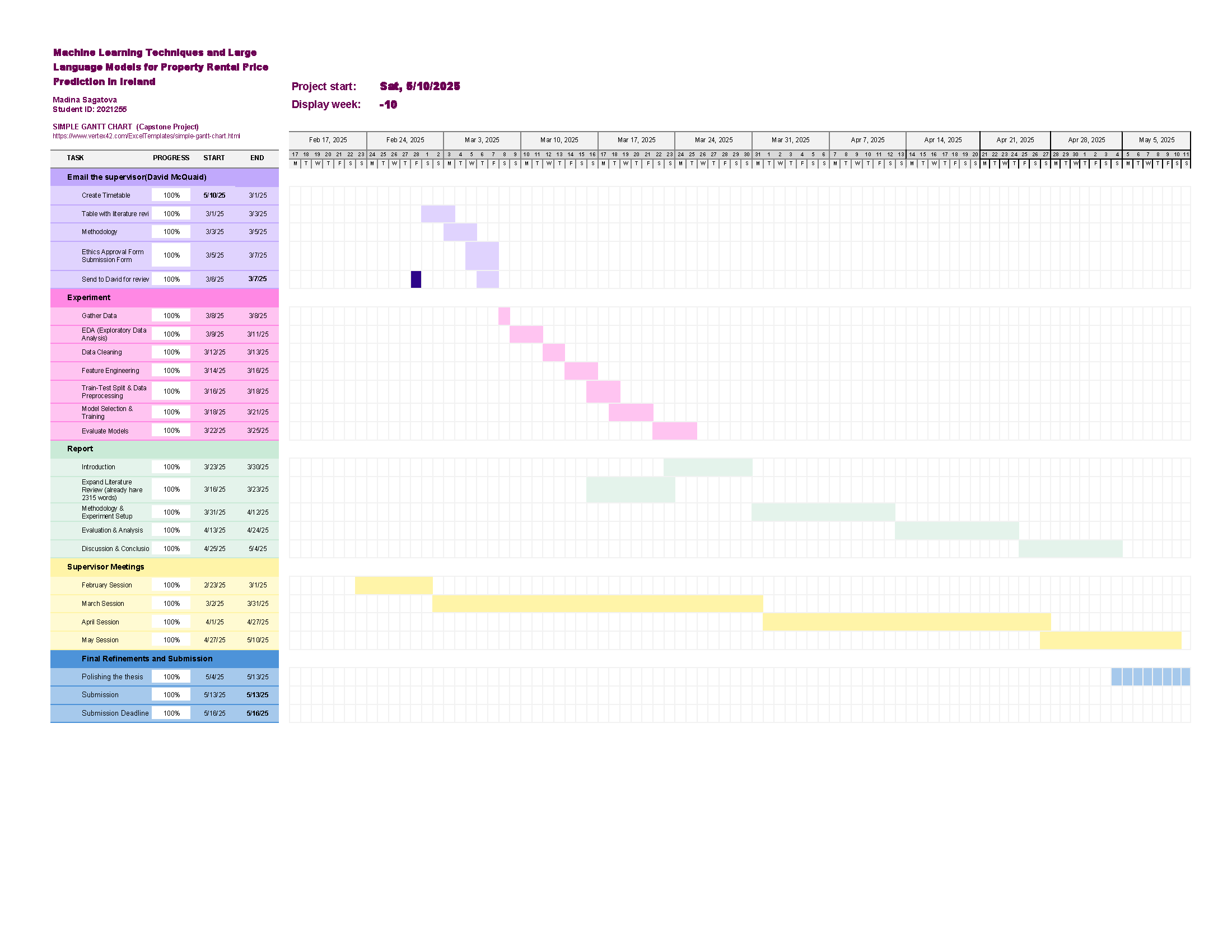


Figure B1: Capstone Project Timeline and Progress Tracker (Source: Author's analysis)

# Appendix C: : Forecasting Methodology Workflow Diagram

Figure C1 displays a workflow of the forecasting pipeline, including time series model selection based on trend and seasonality, hyperparameter optimisation approaches, and selection of LLM-based forecasting approaches. The diagram also outlines evaluation metrics and model comparison methodologies employed in this capstone project.

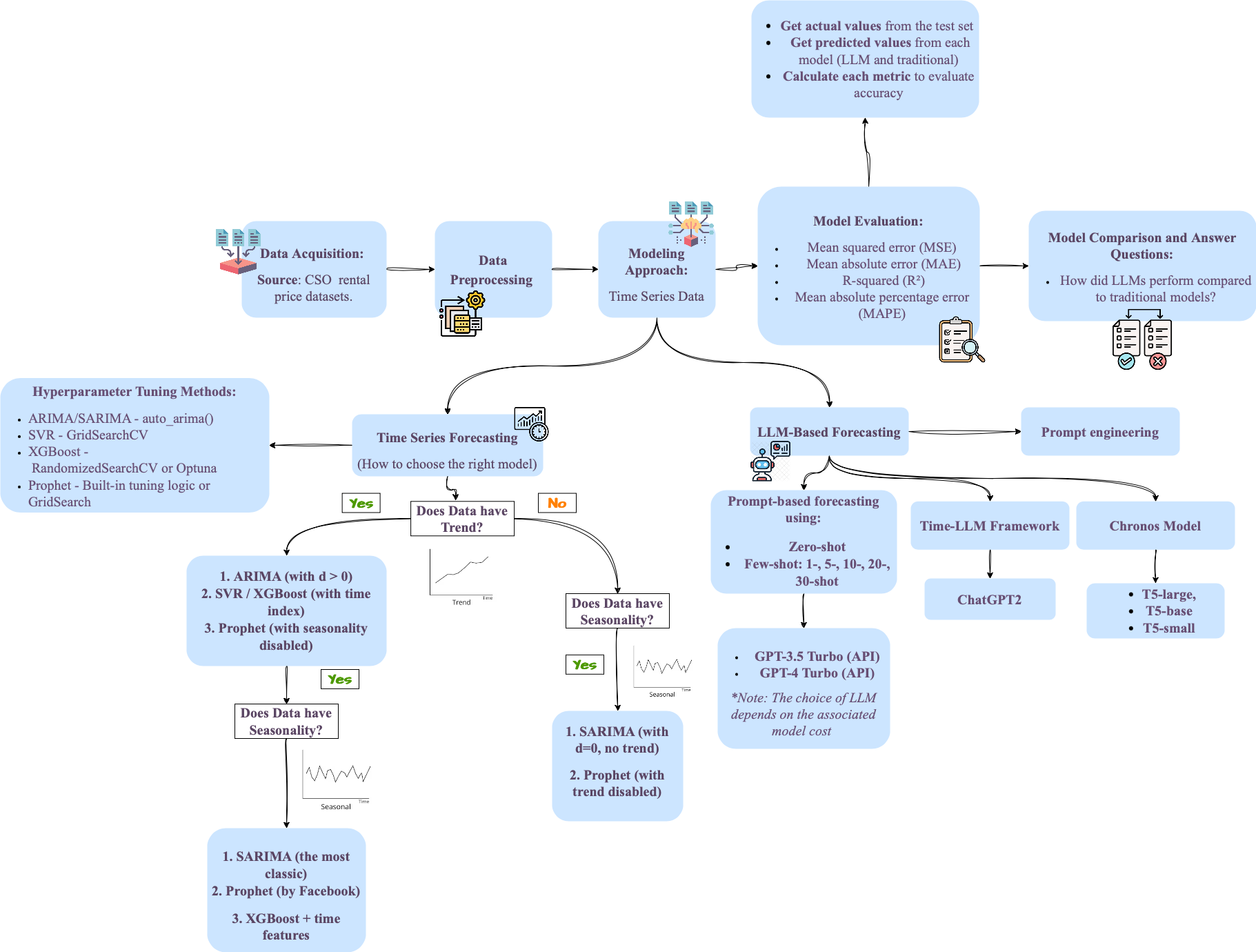


Figure C1: Forecasting Methodology Workflow Diagram (Source: Author's analysis)