A Comparison In Predicting Regional House Prices Using Advanced Machine Learning Techniques

Master Thesis

Business Analytics

by

Gerke Schaap (2640163)

First Supervisor:	Rob van der Mei
Second Supervisor:	Joost Berkhout
Host Company Supervisor:	Rudi Hendrix

Vrije Universiteit Amsterdam
Amsterdam, The Netherlands

Accenture
Amsterdam, The Netherlands

July 2024

Abstract

This study aims to predict average regional house prices in the Netherlands based on a specific selection of macroeconomic, socioeconomic, and demographic variables. The advanced machine learning models Extreme Gradient Boosting (XGBoost), Feedforward Neural Network (FNN), Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM) are compared with an Autoregressive Integrated Moving Average (ARIMA) model on time dependent data. Ablation of macroeconomic, socioeconomic, and demographic components in the data results in four (one complete, three reduced) experimental datasets for the advanced machine learning models, testing the contributions of each component. The best performing models are FNN and XGBoost. FNN performs slightly better on the complete dataset, with a MAE of 0.024 and a RMSE of 0.032, while XGBoost obtained better results on most reduced datasets. All advanced machine learning models obtained better results in forecasting regional house prices than the baseline model ARIMA. Furthermore, it is evident that socioeconomic variables had the greatest impact on predictive performance. Demographic variables also played a minor role, whereas macroeconomic variables did not contribute to the predictive performance in this case study.

Acknowledgements

I would like to express my gratitude to some individuals who supported and guided me during my thesis.

I am grateful towards my supervisors (client and buddy) from Accenture, Rudi Hendrix and Koen Lijffijt-de Vogel, who guided me through this project at Accenture helping me out with any topic or non-topic related problem I encountered during my thesis. I appreciate the freedom I was given and the support at times when I needed guidance. I would also like to extend my gratitude toward Marco de Vree for this thoughtful ideas and discussions.

Additionally, I would like to thank my first supervisor Rob van der Mei from Vrije Universiteit Amsterdam for his guidance and insightful remarks. Also, my second supervisor Joost Berkhout for his time and effort.

Lastly, I would like to express my gratitude to my fellow interns at Accenture for the inspiration and support during gatherings at Accenture. I'm also grateful to everyone else who has offered help, shared their time to discuss problems, or exchanged ideas with me. Your support has been invaluable and greatly appreciated!

Contents

1 Introduction

1.1 Description of Host Company: Accenture

Accenture is a global professional service provider being settled in over 200 cities worldwide. The company provides a range of services in consulting, digital technology, finance and strategy & operations. Accenture's services have reached 7000+ clients in more than 120 countries since it was originally founded in 1989, as Anderson Consulting. However, Accenture is a split-off from the formerly Swiss company Andersen Worldwide Société Coopérative (AWSC) after it broke ties with the parent company in 2000 after enduring conflicts. The conflict, settled by the International Chamber of Commerce, permitted to terminate all the contracts along with changing the company name. The firm was renamed as Accenture, inspired by the phrase "Accent from the future". The services provided by the company are branched off into five core businesses: Strategy and Consulting, Technology, Industry X, Song, and Operations. These core businesses are described in the following paragraphs.

Strategy and Consulting: The services provided by Strategy and Consulting support clients to innovate and reinvent their business strategies and operations to identify growth opportunities. Expertise in diverse domains are exploited and enhanced by using advanced digital technologies, data and artificial intelligence.

Technology: The technology branch of the firm focuses on implementing and producing advanced digital technologies and solutions. Offering cloud solutions such as SAP or Microsoft Azure to clients often integrated with required client-specific digital tools. Advanced technological solutions involved in software development or digital infrastructures are deployed here.

Industry X: The industry X is the latest core business integrated into the core business of Accenture. A merge of Technology and Strategy & Consulting resulted in Industry X. It connects the digital revolution of the Internet Of Things (IOT), Artificial Intelligence (AI), data analytics and robotics to traditional manufacturing processes. It is focused on integrating up-to-date digital technologies in client's engineering, infrastructure and capital projects.

Song: Accenture Song, formerly known as Accenture Interactive, is a division operating in a more creative domain of technological projects, with innovation and execution of creative ideas at its core. The services provided by Song include customer services, product and experience design; technology and experience platforms; creative, media and marketing strategy; and channel, content, and campaign orchestration.

Operations: The last core service Accenture is involved in is called Operations, focusing on the management and optimization of business processes, technology services, and outsourcing engagements. This department is involved in supply chain management and digital efficiency (cloud and AI), but also includes business process outsourcing (BDO) solutions such as finance and accounting, human resources, procurement, and marketing operations.

The project is carried out in cooperation with the area Data & AI in the technology branch in the Amsterdam office located at the Zuidas. This is a specialized area within the technology branch that takes responsibility for projects in data engineering, cloud solutions, and artificial intelligence.

1.2 Introduction To Subject

The global crisis and political decision-making caused the Dutch housing market to suffer more than other western countries [1]. After the housing market had recovered in 2014, house prices have increased approximately 7% per year [2]. However, the increase in prices is not uniformly distributed across regions, as well as the prices itself. In the northern province Groningen, house prices in the municipality Pekela were sold for 215.800 euros on average in 2023, while the city Groningen had average house prices of 349.900 euros. Compared to a larger city such as Utrecht, this was still considered a reasonable price, where average house prices were 474.406 euros [3]. The regional differences in house prices are large and appear to remain large in the future. This raises the question: How much influence does the quality of the living environment have on the price of owner-occupied homes? And are we able to improve the predictions the regional house prices by incorporating regional characteristics as the quality of a living environment?

The prediction of house prices is a difficult problem. The price of housing in the Netherlands fluctuates over time and is influenced by many different factors. Think of macroeconomic, demographic, locational, and structural factors. The complexity of the driving force behind the pricing system makes it difficult to make precise estimates of the market. Furthermore, external influences such as unforeseen economic or political changes are almost impossible to predict but could have a significant impact on the predictions of the housing market, such as the oil crisis in 1979 [4]. Moreover, determining the price is a complex task by itself, as it is driven by the willingness to pay. This (subjective) price determination by notaries or residential investors is often relied on by years of experience. While improving the accuracy of valuations and providing insight in factors influencing house prices can benefit all stakeholders in the real estate market.

In addition, influences on the real estate market also differ on a smaller regional scale. In the city Amsterdam, the most expensive neighborhood has an average appraisal value (WOZ-waarde) of 1.282.000 euros. In comparison, the relatively low-priced neighborhood The Bijlmer is more than four times cheaper, with an average house price of 280,000 euros. This illustrates the large price gap between low-priced and expensive neigbhorhoods in the Netherlands. Certain neighborhoods are more appealing than others due to the availability of jobs, facilities, and accessibility in the neighborhood [5]. In addition, the quality of the houses in a neighborhood influences the average price of housing in a neighborhood.

The housing market is relevant to a large degree of companies, policymakers and many households. For many households, it is not only their most valuable asset, but more importantly, a place to live. The living environment plays an important role in choosing a home. Someone who buys a house is not just buying a property, but also the surrounding environment. Insight into these preferences can be important for the development of new residential areas and the restructuring of existing neighborhoods. People's preferences for their living environment influence the selling price of the house. But how significant is this influence? And do different characteristics of a living environment contribute to the predictions of house prices in a neighborhood?

1.3 Problem Statement

This thesis contributes to improving the prediction of regional housing price trends using machine learning techniques based on a specific selection of macroeconomic, socioeconomic, and demographic variables. The case study is conducted in the Netherlands using a data collection of characteristics of all neighborhoods from the Dutch Institution of Statistics (CBS) [6]. Research in housing price prediction is quite a satisfied field of study, with more than 4,000 research papers published between 1960 and 2020 [7]. However, predicting house prices remains a challenging task. House price prediction is slowly adapting to more advanced and innovative methods, such as deep learning models. Currently, there are two widely used categories for forecasting housing prices. One is based on more traditional regression techniques, such as the Hedonic Pricing Method (HPM) or autoregressive models. The other is based on more advanced techniques including machine learning and deep learning techniques, such as random forest algorithms or artificial neural networks. This thesis compares advanced machine learning models to a statistical autoregressive model. Many researches aim to predict house prices using structural or locational variables only, leaving out the influence of the environment or neighborhood on the selling price [8]. The scope of this research is to predict the value of houses in a neighborhood as a place to live, implicitly reflecting the appeal of a neighborhood rather than the value of structural objects. Therefore, we specifically use a set of variables that explain macroeconomic, socioeconomic and demographic factors on a small regional level (neighborhoods). The specified sets of variables are beneficial to indicate the contribution to these regional characteristics in relation to the appeal of the neighborhood. The data that is used for this research is collected over an annual time period of 19 year from 2004 until 2022.

1.4 Research Approach

The aim of this research is to compare advanced machine learning models against ARIMA regression in predicting average regional house prices in the Netherlands based on a selection of time-dependent macroeconomic, socioeconomic, demographic, and some basic spatial variables. Four different machine learning models, each capable of handling time-dependent features, are selected. Extreme Gradient Boosting (XGBoost), Feedforward Neural Network (FNN), Recurrent Neural Network (RNN), and Long-Short-Term Memory (LSTM). The last two models, RNN and LSTM are also referred to as ”sequence models” in this thesis. The research objective is clarified by the research question and the subquestions:

Research Question:

How do advanced machine learning models compare with an ARIMA regression model using socioeconomic and demographic characteristics in terms of precision in predicting housing prices in regional housing markets in the Netherlands?

Sub-Questions:

- What are suitable baseline models to compare RNNs to?
- How does the baseline model perform in comparison to advanced machine learning models on the complete dataset?
- Do sequence models perform better than non-sequence machine learning algorithms?
- Are macroeconomic, socioeconomic and demographic factors contributing to the performance of the advanced machine learning models?

To compare the various selected machine learning models, the baseline model ARIMA has been selected. Alternatively, hedonic pricing methods are also considered as a baseline model. These models represent classical statistical techniques that have been used in previous research predicting time-dependent data, while machine learning models represent modern approaches. ARIMA emphasizes differences between the predictions based on historical prices and the historical characteristics that are (in)directly related to the average house prices. The four different machine learning models that have been chosen are all capable of handling time series data. The XGBoost and FNN models are capable of handling time as a feature. These two models are two well performing models in the literature (section 2). The other two machine learning models, RNN and LSTM, are equipped to handle time series and sequential data structures. These models are able to capture information that has been processed already, making them suitable for this experiment. The models are trained using four (one complete, three reduced) experimental datasets for the advanced machine learning models, testing the contributions of each component. The performance of the ablative datasets and the complete dataset are compared and evaluated by various performance metrics. In addition, the performance of the advanced machine learning models are compared against the baseline model ARIMA. Further analysis of the grouped features is performed using feature importance metric of XGBoost.

This paper is structured as follows. Section 2 provides information about relevant research and background information about this field of study. In Section 3, the data preparation and explanatory data analysis is provided. Sections 4 is dedicated to the description of the methods and experiments that were conducted and Section 5 summarizes and compares the results. Section 6 presents the conclusion and discusses limitations of the study and offers directions for further research.

2 Literature Review

Part of the previous section was an introduction to the problem. In this section, the characteristics of the housing market are explored in the literature, with a particular focus on the context of the housing market in the Netherlands. In addition, we address the different prediction methods investigated in the context of house price prediction, including machine learning approaches.

2.1 Dynamics of the Housing Market

The real estate market itself can be considered as a financial market with unique characteristics that differentiate itself compared to traditional bid-ask markets. It can be stated that the housing market operates as oligopolistic market. It is an illiquid market where heterogeneous housing products are sold. There exists no restriction or compliance to enter and exit the market. Although in practice, there are budget constraints that withhold consumers and producers from entering the market. The assumption that there exists perfect information availability is reasonable. Most information about the availability of houses, prices, and attributes is available and provided.

The price of a house can be decomposed by multiple factors that influence the valuation of a property. In the following paragraphs, these factors are categorized as market factors, locational factors, structural factors, and external factors. In the literature, these factors are often researched on their effects on property valuations or used in their dataset for modeling the housing prices.

As a market factor, supply and demand drive the housing market. However, demand is more prominent in the determination of prices, as the supply of houses adopts rather slowly to the market. The long-term housing supply generally increases by no more than 1.5 percent per year in the Netherlands [9]. In addition to demand and supply, there are other market factors that determine the price of a property. The functioning of the economic cycle influenced by employment rates, interest rates, and other (macro)economical factors plays an important role in the housing market. Economic growth implies higher employment rates and higher incomes, which increase the willingness to purchase a house. On the other hand, house prices may drop as a result of a recession, when the real estate market tends to slow down as buyers are more reserved. Monetary policies that tend to steer the economic cycle towards stability are driven by factors such as the interest rate [10]. Changes in interest rate directly affect the housing market, due to its direct and strong correlation with mortgage rates. Lower interest rates, meaning lower mortgage rates, can lead to increased buying activity, while higher rates can cool the market. Specifically in the Netherlands, lower interest rates will also attract real estate investors, knowing that the Dutch housing market is characterized by steady growth with occasional large fluctuations in prices [2].

Another key factor determining the price of houses is the neighborhood or place where a house is located. Each location has different characteristics in terms of accessibility, traffic conditions, work opportunities, proximity to facilities, social cohesion, and environmental surroundings, among other variables. And most machine learning models use a form of these spatial features [11]. Although these factors are influenced by individual buyer preferences, research found that higher criminal rates have negative effects on house prices, while proximity to schools increases the willingness to pay for a property [12]. Also, cities or neighborhoods with rich histories or cultural landmarks often see higher property values due to their unique appeal. Research by Been et al. [13] outlines that specific qualities of designated historic neighborhoods, such as restrictions on building height and pre-existing desirability, affect property values in New York neighborhoods. In terms of the difference between property valuations in urban areas versus rural areas, different points of view exist in the literature. On the one hand, it argued that house prices in cities are valued higher because of better accessibility and being closer to public amenities such as schools, shopping centers, and hospitals. However, rural areas offer different lifestyle benefits [14].

Although structural properties are out of the scope of this research, the characteristics of a property are essential to valuate houses. A property with more desirable attributes leads to a higher property value. Using the hedonic pricing method (Section 2.3), it can be explained which structural characteristics contribute to the valuation of a property. It has been researched that the size of the house, the number of bedrooms, bathrooms, and floors imply a higher property value, as buyers are willing to pay more for more (functional) space and higher living standards [5]. Additional amenities such as a balcony, a garage or a patio further increase the value of the property [15]. Aged houses are less worth, unless the property has a historic character [16]. For machine learning models, the structural features 'Area' or 'Size' of a house are ranked as the most important features, reaching feature importance levels between 8% - 20% [17]. In this thesis, structural features are omitted due too the lack of publicly available data of this type. Despite the possibility that generalized structural characteristics within a neighborhood could add value to the predictions, this research is carried out without structural housing characteristics.

Among the many external factors that influence house prices is a selection of relevant factors that affect house prices indirectly. Although indirect factors are more difficult to prove, evidence in the literature has been found supporting the effects of external factors. One of these factors are demographic trends. Research by Gevorgyan K. [18] found in a case study that if the population increases by 1 percent point, the house prices increase by 1.4 percent point. An increase of 1 percent point in the birth rate results in a house price increase of 4% - 5% after 20-25 years [19]. This paper argues that the strong age-dependence of investments in owner-occupied housing, combined with the limited response of other housing investors to these fluctuations, causes an increase in demand of houses, driving up the house prices. Evidence from a Chinese study [20] shows that the old age ratio (65 + years) has a positive relationship with regional house prices. In addition, the child dependency ratio has a negative relation with house prices. In underdeveloped areas, changes in the age structure of a population have less impact on housing prices.

Also, factors such as population growth, migration patterns, and socioeconomic trends tend to influence the price (demand) of housing [21]. Annual household income, years of education, and work experience are used to evaluate housing prices using hedonistic pricing methods [22]. And income has been proven to be the third most important feature in the study by Rico-Juan & de La Paz [23], who used a random forest regression technique by accumulating individual absolute Shapley values, indicating the significance of the income variable. Also, the perception of a neighborhood or the social cohesion could influence house prices. Ketkar, K. [24] discovered that white residents in New Jersey often exhibited sensitivity toward the proportion of non-whites in their neighborhoods. Further research by Wei, B. & Zhao, F. [25] highlighted that there is racial disparity in mortgage lending. Meanwhile, Richardson and his team [26] identified social class as a significant influence on property values, although other factors may also contribute.

2.2 Volume and Pricing of the Dutch House Market

Periods of up- and downswings in the real estate market are recognized globally. However, in a healthy economy, periods of rising prices overall last longer than periods of declining prices as a consequence of monetary policies. An analysis by the Dutch institution Centraal Planbureau (CPB) [2] exploited that the Netherlands has relatively long periods of rising and declining prices. The longest period of upswing lasted from 1984 until 2007, having a duration of 23 years and a growth in price of 115% over this period. Compared to surrounding European economies, these statistics are exceptionally high. The United Kingdom has had a growth period of 7 years with a price change of 60% and Germany has had the longest growth period of 5 consecutive years with a growth percentage of 7%. Due to the length of these upward trends, the growth in housing prices is often relatively predictable based on the growth in the previous year, since there is usually a high autocorrelation. Also, having a stable issuance of building permits caused a relatively low price volatility [27].

Since 1970 there have only been two periods of downward trends in house prices in the Netherlands, both caused by global crisis. The Great Financial Crisis (2008 - 2013) and the second oil crisis (1979 - 1983). During these downswings, the house prices dropped rapidly. In less than 4 years time, the house prices dropped by more than 45% [28] during the economic recession in 1979. Before the crash, house prices were increasing excessively in the Netherlands, becoming to expensive for many potential buyers. Due to the cut in Iranian oil supply, oil prices tripled in a short period of time. The rent increased globally and a higher unemployment rate in the Netherlands caused the house prices to decline. Many home-owners could not afford the housing expenses anymore and were forced to sell their property, creating an oversupply of houses on the real estate market. During this period, building permissions decreased from 29 thousand in 1979 to only 7 thousand in 1983 [2]. The great recession in 2008 caused the house prices again to drop significantly. The house prices decreased in value by more than 20% since the last peak value. Both crises had a larger decline in the real estate market compared to similar European countries such as Spain with a decrease of 28% compared to 39% in the Netherlands.
However, predicting how long a declining period will last until a turning point is reached is often difficult to predict, while these periods can have negative effects on the economy [4]. The financial crisis caused such a fluctuation in the housing prices where prices dropped more than 20% since the last peak value. This had the effect that many home owners drowned 'under water', which means that the mortgage loan was higher than the current price of the home.

Analysis of regional house prices in the Netherlands shows that there exist differences in price changes between regions [2], indicating spatial heterogeneity. Emperical evidence shows that a small economic setback in 2003 did not affect house prices in all regions. The region of Amsterdam had decreased, while the Groot-Rijnmond region continued to increase. Also, the increase of house prices in the region Zuid-Limburg and De Achterhoek were minimal in the period 2000-2008, while the prices in Noord-Friesland increased rapidly. After the financial crisis in 2013, the four largest cities: Amsterdam, The Hague, Rotterdam, and Utrecht show the largest growth in house prices.
In addition, Dutch house prices are exceptionally volatile. During a period of 33 years (1985 - 2018), the standard deviation was between 20-30 for nearly all COROP [1] regions. In comparison, the volatility of the German and Swiss house prices was 6.7 and 12.8, respectively.

Macroprudential Policies effecting the Dutch Housing Market

Since the Great Financial Crisis, financial regulations and policies have been tightened to prevent excessive risks of subtle mortgage lending standards and to maintain financial stability. After the crisis, the LTV (loan-to-value) norm was reduced from 106% to 100% [29]. A lower LTV norm reduces vulnerabilities in the housing market because buyers must build up a financial buffer for their potential purchase, but this makes it more difficult for first-time home-buyers to purchase a house. Although the LTV norm is still higher in the Netherlands compared to other

[1] A cluster of one or more adjacent municipalities in the same province, designed for regional research.

EU countries, its effects are still noticeable. Elbourne et al.[29] describe in their research on macroprudential policies in the Netherlands that the effects on the economic cycle are minimal. During a cyclical upswing, the effects on pricing and supply of houses remain subliminal, as the more prosperous households will take advantage of the decreased demand of limited households.

Impact of the COVID-19 Pandemic

The global crisis had a large impact on the housing market. However, the most recent crisis caused by the COVID-19 pandemic appears to have a subliminal impact on the house prices. Research by Kransberg, S and Rouwendaal, J [30] show that the house prices in one out of the four different cities Maastricht, Eindhoven, Utrecht and The Hague is affected by the pandemic. The city Maastricht has a decline of 6% in the housing market, 5 months after the outburst in march 2019. However, close to 2020 the house prices have restored, suggesting the decline in house prices was caused by the shock. The impact of COVID-19 on other economies is also relatively small. Households did reduce their spending and increased their saving due too uncertainties caused by the pandemic, but this did not directly impact the housing market [31]. Although, this could indirectly enhance the wealth inequalities between households leading to shifts in socioeconomic classes. Longer-term effects could lead to increased house prices or lead to more pronounced inequalities in housing access, depending on how other economic factors unfold.

2.3 Predictions And Modeling Approaches For House Pricing

Modeling approaches for house price prediction and property valuation is a research field that many researchers have tried to address. A diverse selection of models has been investigated and evaluated in a wide range of scientific papers using various data sources [8]. The most popular methods in the early stages when big data became substantially more available were Multiple Regression Analysis (Hedonic Pricing Analysis, section 2.3), Kriging, Spatial Econometrics and later Spatially Varying Coefficient Models. In recent years, more advanced machine learning and deep learning methods have been researched.

Hedonic Pricing Method

A popular technique used in pricing models is the Hedonic Pricing Method (HPM). This method is a quantitative valuation method that derives its estimates by decomposing the asset into a combination of characteristics that determine the price of the asset. Internal factors and perhaps external influences are incorporated into the regression model to estimate the price of goods. The method is an indirect valuation method derived from the consumer theory introduced by Lancaster [32] in 1966 and Rosen's [33] theoretical model in 1974.

In the academic literature, there exists a substantial amount of research performed on the exploration of the hedonic pricing method and its application to the real estate market [34][35][36]. One of the pioneering researchers to apply an early comparative form of the hedonic model was in 1967, when Ridker and Henning [37] published their research on the explanatory relationship between air pollution and residential property values. A more appropriate comparison can be made with the application of the hedonic model in the housing market by Freeman in 1979 [36], who published a foundational survey on the application of the hedonic price evaluation of property valuations, emphasizing the implicit marginal effects of how environmental attributes are capitalized in property values.

Hedonic pricing methods serve as a fundamental methodology for house price valuations and predictions. Having interpretable coefficients, the model serves as an explainable method for prediction outcomes. Therefore, the model has been applied to different attributes verifying the relevance of various variables in predicting house prices. As Geerts et al.[8] describe in their review, most research has been performed on structural property data. However, also used in hedonic price analysis are socioeconomic and demographic variables. Studies have indicated several socioeconomic factors that impact property values. Ketkar, K. [24] discovered that white residents in New Jersey often exhibited sensitivity toward the proportion of non-whites in their neighborhoods. Further research by Wei, B. & Zhao, F. [25] highlighted that there exists racial disparity in mortgage lending. Meanwhile, Richardson and his team [26] identified social class as a significant influence on property values, although other factors may also contribute.

However, the main limitation of the hedonic pricing methods is the dependency on linear variables, while potential machine learning methods are capable of handling nonlinear relationships and are able to learn from data. Machine learning methods appear to outperform hedonic pricing methods. The research by Yazdani, M [38], empirically compares the hedonic pricing method, machine learning methods, and deep learning methods, resulting in a better performance of the Random Forrest regression model compared to the hedonic pricing method. When trained on a larger dataset, deep learning models show potential for more accurate prediction.

Time Series

A statistical approach based on the mean, trend, and/or seasonality of chronologically observed data is called Time Series Analysis. There exist various methods to execute time series analysis such as moving averages, exponential smoothing, and ARIMA. The general price trend, cluster-level price trends, and other specific characteristics are time series that are modeled in a so-called Hierarchical Trend Model (HTM) for selling prices of houses [39]. The cluster-level trends are composed of varying patterns across house types, districts, and neighborhoods, allowing both spatial and temporal dependencies to be accommodated. Combining autoregressive moving average (ARMA) with OLS results in the ARMAX model [40], which is applied in a forecasting equation to estimate average neighborhood prices. This strategy involves property attributes, neighborhood attributes, spatial differences, and mortgage rates taken in different temporal lags that reduce statistical estimation problems [41]. However, since time series models are not explicitly suitable for capturing features other than their own time dependency, they have a low prevalence in house price valuation studies [8].

Other Statistical Approaches

Other statistical approaches described extensively in the literature are the methods: Kriging, Spatial Econometrics and Spatially Varying Coefficient Models. These methods rely on the spatial and geographical variation between sampled locations. The geostatistical valuation method known as kriging allows the prediction of house prices in areas that have not been sampled. The method relies on the assumption that nearby locations are more likely to have similar values than locations further apart. Research conducted by Crosby et al. [42] implemented geostatistical kriging to develop an automated valuation model for estimating residential property prices in Coventry, UK. It was determined that road travel distance and time have a stronger correlation with property prices than Euclidean distance, achieving a fit of 0.69 (R^2) in predicting real estate prices, in contrast to the Euclidean method's 0.66 (R^2). In a comparison of spatial interpolation techniques and machine learning models, both the Artificial Neural Network and Random Forrest estimator surpassed traditional statistical methods in terms of

RSME, scoring 102.0348 and 80.8789 respectively. Meanwhile, ordinary kriging and Inverse Distance Weighting achieved a RSME of 131.00 and 126.53 [43]. An extension of the existing hedonic pricing method are the Spatial Econometrics techniques. These techniques incorporate a spatially weighted average of house prices. This model assumes that neighboring house prices influence the prices base on the distance of the neighboring house. A similar modeling technique is Geographically Weighted Regression, which allows parameters to adapt over space by weighting higher coefficients for smaller distance based observations. A relevant application of these methods is studied by Osland [44], who compared these methods to the hedonic pricing method in valuating houses in Norwegian municipalities. The spatial econometric methods appeared to be better indicators of spatial variables in predicting house prices. These techniques complete the list of statistical models mostly used in the early study of house price prediction methods that are widely researched in the literature [8].

Machine Learning Approaches

In the past two decades, machine learning approaches have started to evolve in the field of predicting house prices. The advancement of computational software allows for smarter and quicker methodologies for property valuations and price predictions. In 1996, researchers made the first comparison between the hedonic pricing method and an artificial neural network (ANN), demonstrating that the neural network shows great potential in predictive powers. However, due to the black-box nature, the explainability and reliability of the model are put into question [45].

A neural network can be used to model the complex relationships between various features of a house and its market price. Each neuron in the network processes inputs-such as the size of the house, location, age, and number of rooms—by applying learned weights and a nonlinear activation function to these inputs, combining the information in complex ways to predict house prices. The network's hidden layers enable the extraction and integration of nonlinear interactions between features that might affect prices, such as the interaction between location and house size. During training, the network is fed sample data of houses with known prices, allowing the NN to adjust its weights through backpropagation. This process optimizes the network's weights to reduce the discrepancy between the predicted and actual prices, enhancing the model's precision in predicting market values from input features.

The complexity and nonlinear relation of the data sourced for valuing housing prices is a logical reason to consider using an artificial neural network model to complete this task. It has been researched that the neural network model has a constant performance with over 150+ models trained on a Chinese real estate dataset that obtained RSME statistics between 1% and 2% [11] for each model. When comparing ANN models to hedonic pricing methods, the artificial neural network outperforms the hedonic pricing method by an improvement of more than 50% on the performance metrics in a study by Selim, H. showing the potential of neural networks [46]. Also, an extended design of the ANN, the 2-layer feed-forward neural network designed using memristors as synapses, employs its functionality when it comes to predicting house prices, having results close to its actual values [47]. Overall, the performance of Neural Networks seems very promising.

However, ensembling methods also perform well in valuing properties and in some cases even outperform the ANN. The authors in [48] compared a linear regression model, a multilayer perceptron, a random forrest regression model, a support vector machine and a XGBoost regression model. The performance of the linear regression model and the XGBoost model obtained high scores on the RMSE: 0.130 and 0.112, respectively. The multilayer perceptron MLP obtained an RSME of 0.190. However, having only 2930 records in the dataset, it could be criticized that

MLP's perform better on large datasets.
A case study performed on a dataset of house prices in the Spanish city Alicante shows the performance of a selection of ensembling methods: Gradient Boosting Regressor, Extreme Gradient Boosting, and Light Gradient Boosting Machine. The overall performance of all models were strong. The results show that the Gradient Boosting Regression performs best with the best overall performance (R^2 : 0.9192, RSME: 0.1818) [49].

Less explored in scientific literature is the application of recurrent models on predicting house prices. This requires a different view and approach on the data, as a single house is not sold sequentially (each year for example). The development of house prices in a bounded area (country, city or neighborhood) over time can be modeled as a sequence model. This was the exact approach of Chen et al. [50] where they applied a Long-Term Short-Term (LSTM) model and a Recurrent Neural Network temporal dataset containing the average monthly sales of properties in 80 districts in Beijng.

Although many of these models perform well in assessing the prices of individual houses based on structural and locational features, they often overlook the effects of demographic and socioeconomic factors that contribute to the desirability of a community. Changes in demographic factors over time, such as an aging community [51], lead to a decrease in house prices in the following years, but most research do not take this into consideration when predicting house prices. For potential buyers, not only the locational and physical attributes of a house, but also the communal and social attributes of a neighborhood is what attracts potential home-owners to a new residence. In contrast to most research, predictions in this thesis are made based on the environmental characteristics and historical trends of a neighborhood.

3 Data Preparation

In this section, a description of the data will be provided, how (raw) data has been sourced and manipulated, and how a dataset has been prepared for modeling. An overview of the techniques used is given and empirical issues with the data are highlighted.

3.1 Raw Data Collection

The raw data used for this thesis are mainly collected from the Dutch Institution of Statistics (CBS) [6]. The initial datasets obtained here are from the series: "Kerncijfers Wijken en Buurten". This series of data "Kerncijfers Wijken en Buurten" is a collection of all core statistics of a neighborhood obtained for a specific year. In the data, information such as population, income, housing, access to facilities, social security, business establishments, urbanization, care, and other characteristics of the neighborhoods is collected. The institute has released a dataset each year since 1995. This thesis collected the data from the "wijken en buurten" collection [6] for the year 2004 until 2022 with annual intervals. Datasets released earlier than 2004 could not be used because the variable 'WOZ'(appraisal value), fundamental for obtaining the target variable, was not included in these datasets. The year 2023 was not included because part of this dataset was not released yet during this research. Note that this is raw data and that extensive data preparation was necessary to configure a usable dataset for machine learning. Additional datasets have also been obtained from the Dutch Institution of Statistics. The target variable 'average house price', regional income statistics, households' composition, and macroeconomic variables are collected from different datasets from the Dutch Institution of Statistics (CBS). Other macroeconomic variables have been collected from De Nederlandse Bank [52].

3.2 Data Cleaning

The collected raw data required an extensive cleaning and preparation process. First, the series of datasets from the collection of "Kerncijfers Wijken en Buurten" was cleaned. Since the data was not collected consistently for each measure or with inconsistent measures over time, the data contained a lot of variation between the variables. An overview of the differences of the variables that have been gathered in the series from 2004 until 2011 can be seen in figure 19 (Appendix 1.1). As can be observed in this figure, the methodology for obtaining and preparing the statistics, changes over time. In 2012, a major change took place in the presentation of the data, where all statistics were revised and renamed with new variable conventions by CBS researchers. Because the series of data contains an inconsistent alignment of its variables over the years, renaming and recomputing variables was a necessary step in the process to obtain valid data, which has been performed after a selection of variables has been executed. Demographic, socioeconomic, and several regional attributes that did not contain more than 35% missing data values have been selected during the initial division of relevant and nonrelevant variables. A more in-depth analysis of the variables has been performed after the collected dataset has been fully established.

Subsequently, other datasets containing data regarding the income per age group [53], household composition [54], average property values [3], and macroeconomic values [55] [56] [52] are added to the existing regional dataset "Kerncijfers Wijken en Buurten". In the dataset "Kerncijfers Wijken en Buurten" there exist three regional levels: municipalities (m), districts (d) and neighborhoods (n). The smallest regional dimension is used for predicting the house prices which is the level neighborhoods (n). The regional dimension in the datasets other than the "Kerncijfers Wijken en buurten" dataset is covered over municipalities (m) contrary to neighborhoods (n) statistics, which causes the variability of these statistics to reduce among the regional dimension of neighborhoods. This dataset now contains a large collection of 148 variables, 18,761 neighborhoods, and 187,756 observations in total. In the following chapters, we explain how the dataset will be explored on outliers and missing values. Furthermore, in the last section a final feature selection takes place to determine the features for the definite dataset.

Detecting Outliers

All variables are visualized and analysed using boxplots. Each variable is individually analysed with the underlying definition of each measure to validate if certain values should be interpreted as outliers. A selection of four characteristics has been selected with an increased likelihood of containing outliers (Figure 1).

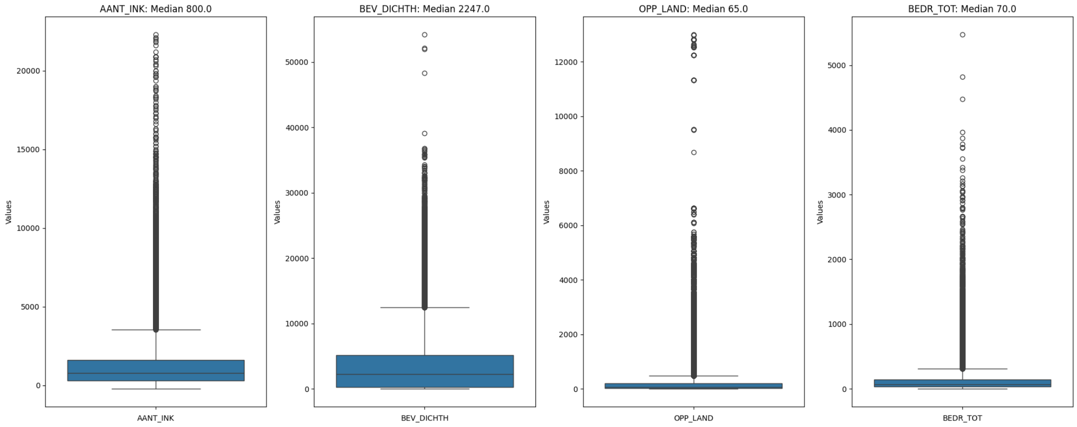


Figure 1: Boxplots of selected features with possible outliers

The features depicted in Figure 1 are Number of Income Earners (AANT_INK), Land Area (OPP_LAND), Population Density (BEV_DICHTH) and Number of Companies (BEDR_TOT). Number of Income Earners (per neighborhood) has an observed median of 800 and values reaching over 20.000. Similarly for the extreme values of the population density in a neighborhood, having a median of 2247 with extreme values reaching more than 40.000. Analysis of these extreme values show that all values belong to neighborhoods located in Amsterdam and Rotterdam, the most highly populated cities in the Netherlands. Neighborhoods with a high household density are De Jordaan or the Bijlmer in Amsterdam and Ommoord and Groot-IJselmonde in Rotterdam. These extreme values also hold for the variables 'AANT_MAN' and 'AANT_VROUW' reflecting the absolute number of man and women in a neighborhood. Also, the attribute 'AANT_HH' (Number of Households) has similar result for the extreme values of these neighborhoods.

Land Area (OPP_LAND) varies a lot due to the differences in defining the area of a neighborhood, which is defined by the municipalities. The defined area in a city is often smaller than a neighborhood in a rural area. The extreme values (> 12.000) belong to the rural areas in the municipalities of Dronten and Zeewolde, large rural areas in the Netherlands.

The total number of companies (BEDR_TOT) in a neighborhood contains some extreme values that reach over 5000 companies, while having a median of 70. Observed values can vary greatly based on the municipal land use plan of the area. Areas that have been assigned as business districts, such as the neighborhood Stadsdriehoek in Rotterdam. This business district in the center of Rotterdam contains more than 5000 companies.

In addition, neighborhoods with fewer than 200 residents in total are discarded from the dataset. Many demographic features are excluded from the data by CBS in case there are less than 50 residents in a neighborhood. This holds for attributes such as age groups, marital status, and western/nonwestern origin. Socioeconomic features, such as income, are kept confidential for neighborhoods with fewer than 200 residents. From 2019 onward, this confidentiality threshold for income data was raised to 2500, contributing to a higher rate of missing values.

Missing Values

The data contains 5.92% missing data values. Each of the variables that contain missing values is evaluated and imputed or discarded depending on the nature of the variable or the pattern of the missing values. Most missing values occurred in the "Wijken en Buurten" dataset [6] where 32 of the 54 selected features contained missing values. The number of missing values differs per attribute, ranging from 0.01% to 24.43% per attribute (Figure 2. The additionally obtained dataset including income variables contains 11 to 13 percent missing values (Appendix 1.2). The remaining data sources do not contain missing values. Four appropriate methods have been selected and modified to preserve the dependency of time and location for the specified feature. The four methods are: Mean Imputation, Forward / Backward Fill Imputation, Linear Interpolation, and Multivariate Imputation by Chained Equations (MICE). Each of these methods is explained in detail in the following paragraphs. A detailed overview of the variables that contained missing values, the number of missing values imputed for each variable, and the respective imputation methods applied in the order they were implemented is provided in Appendix 1.2.

Mean Imputation: Mean imputation stands as the initial approach for imputing missing data, where the missing values are replaced with the mean of the relevant feature. In our approach, we avoid using the general mean of a feature since it aggregates values across different years and neighborhoods. To address this issue, we employ two variations of mean imputation, each taking either the time ($i = year$) or the regional dimension ($n = neighborhood$) as a fixed value.

Methodology of Mean Imputation with Fixed Region (n):
Let $x_{i,n,a}$ denote the value of attribute a at time i for region n, where $i \in I$ represents the set of year, $n \in N$ represents the regional indices set and $a \in A$ represents the set of attributes.

Suppose that the value $\tilde{x}_{i,n,a}$ is missing. The mean imputation method imputes the missing value with the mean values for region n for all years in I for attribute a. The imputed value, denoted as $\hat{x}_{i,n,a}$, is given by:

$$\hat{x}_{i,n,a} = \frac{1}{|I_{n,a}|} \sum_{j \in I_{n,a}} x_{j,n,a}, \tag{1}$$

where:

- $I_{n,a}$ is the set of all time indices $j \in I$ where the value $x_{j,n,a}$ is not missing,
- $|I_{n,a}|$ is the number of years in the set $I_{n,a}$.

Methodology of Mean Imputation with Fixed Year (i):
Let $x_{i,n,a}$ denote the value of attribute a at year i for neighborhood n, where $i \in I$ represents the set of years, $n \in N_m$ represents the set of neighborhoods (n) within the municipality m, and $a \in A$ represents the set of attributes.

Suppose that for a specific year i and municipality m, a missing value $\tilde{x}_{i,n,a}$ exists for neighborhood $n \in m$. The mean imputation method replaces the missing value with the mean of all available values for all $n \in m$ at for a fixed year i and for a fixed attribute a. The imputed value, denoted as $\hat{x}_{i,n,a}$, is computed as follows:

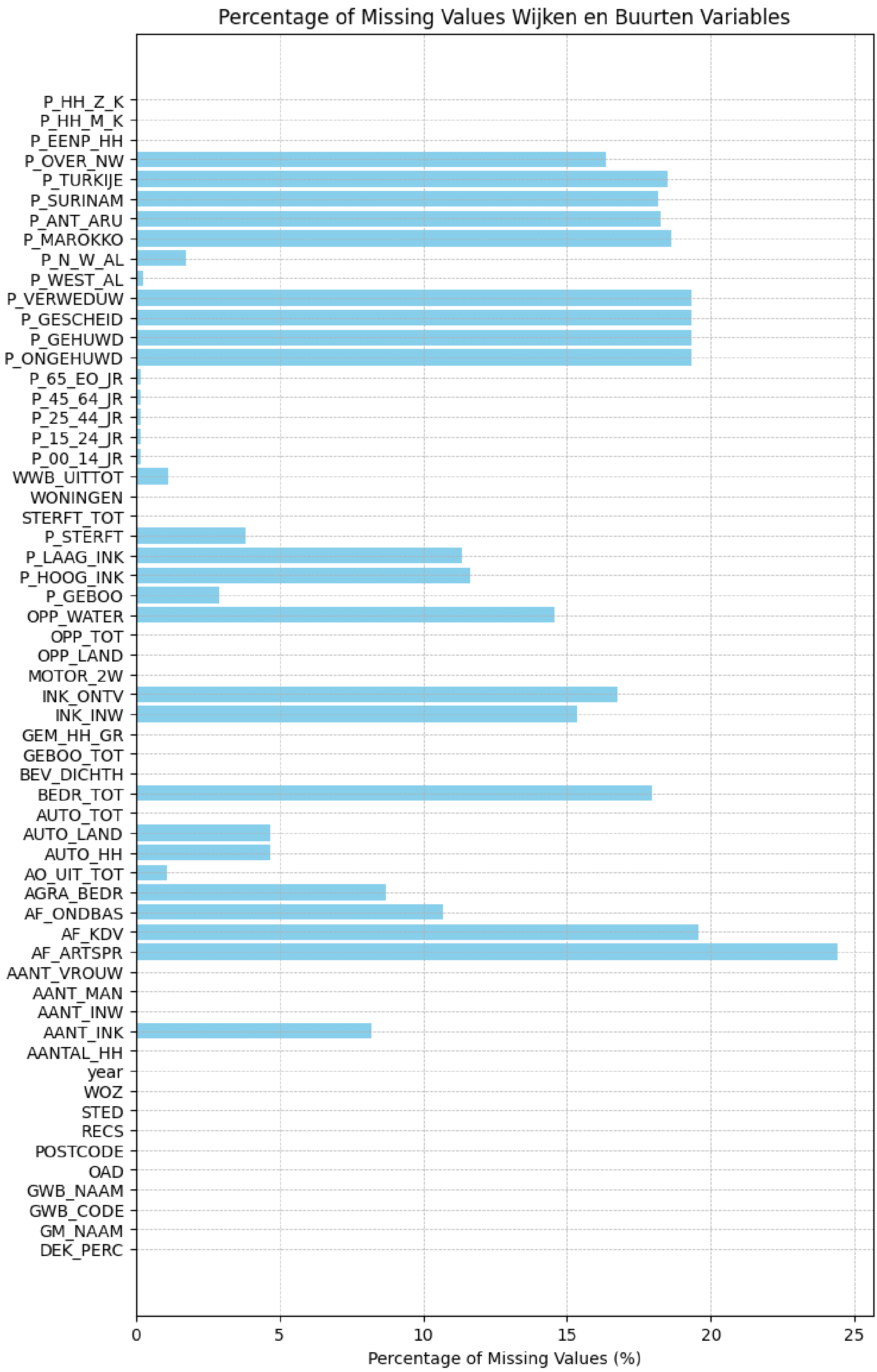


Figure 2: Percentage of missing values per variable of the Wijken en Buurten data (Only variables that have been used in the final dataset).

$$\hat{x}_{i,n,a} = \frac{1}{|N_{m,a}(i)|} \sum_{j \in N_{m,a}(i)} x_{i,j,a}, \tag{2}$$

where:

- $N_{m,a}(i)$ is the set of all neighborhoods j within municipality m for which the value $x_{i,j,a}$ is not missing at time i,
- $|N_{m,a}(i)|$ is the number of neighborhoods in the set $N_{m,a}(i)$.

Mean imputation has some drawbacks and limitations. *Loss of variability* and *Underestimation of standard errors* might occur, as mean imputation does not preserve the variance or distribution of the original data. It reduces the variability in the dataset by replacing missing values with the same constant. Mean imputation is only applied to features that have a small proportion of missing values. In case missing values have a dependency to other variables or when they are not missing at random, *Bias in estimates* might occur. This method is applied to attributes with few missing values and low variance.

Backward Fill / Forward Fill Imputation: Secondly, backward fill / forward fill imputation is applied to impute missing values. This method looks at the previous data point within the fixed region n and specified attribute a,

$$x_{i,n,a} = x_{i-1,n,a} \tag{3}$$

provided that $x_{i-1,n,a}$ is not missing. This continues backward until it finds a non-missing value or reaches the start of the data. Subsequently a forward fill is applied which follows the exact same method for the next data point within the fixed region n and specified attribute a.

$$x_{i,n,a} = x_{i+1,n,a} \tag{4}$$

The combination of backward fill followed by forward fill ensures that each missing data point is attempted to be filled by the nearest previous non-missing data point. It might occur that such a data point does not exist, because the missing entry occurs at the beginning of the data or all previous entries are also missing values. It then attempts to fill the missing value with the forward nearest non missing value. This method is applied to attributes that expect to have similar value as the previous of subsequent years, such as distance attributes (Figure 3).

Linear Interpolation: Given a dataset with observations $x_{i,n,a}$ we address missing values for a fixed region n and attribute a using linear interpolation. Specifically, for valid years $I_{\text{valid}}(n,a)$ without missing values, a linear regression is applied:

$$x_{i,n,a} \approx \beta_0 + \beta_1 i \tag{5}$$

where β_0 and β_1 are the estimated regression coefficients. For any missing data point $x_{i,n,a}$, the imputed value $\hat{x}_{i,n,a}$ is computed as:

$$\hat{x}_{i,n,a} = \beta_0 + \beta_1 i \tag{6}$$

This method is mainly applied to variables that have an increasing or decreasing linear trend as the years increase. For example, the income attributes.

Multivariate Imputation by Chained Equations (MICE): The final imputation technique, known as Multivariate Imputation by Chained Equations (MICE) [57], is a robust and flexible iterative approach for handling missing data using a series of chained equations. It produces multiple datasets based on the observed data which are then pooled to final estimates to fill in missing values, preserving the relationship between variables in the original data. The specific regression model utilized in this method is the Bayesian Ridge regression model. The model is defined as follows:

$$y = X\beta + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I) \tag{7}$$

$$\beta \sim \mathcal{N}(0, \tau^2 I) \tag{8}$$

where:

- y is the vector of observed values for attribute a,
- X is the matrix of other attributes used as predictors,
- β is the vector of regression coefficients,
- ϵ represents normally distributed errors,
- σ^2 is the variance of the error term,
- τ^2 is the variance of the prior distribution on the coefficients.

This method is not particularly capable of imputing variables with large numbers of missing values. MICE relies on the relationships and correlations between observed values in the dataset. If a significant portion of the data is missing, particularly if the missing data is concentrated in specific variables, it becomes difficult for the algorithm to accurately estimate these relationships. Therefore, it is mostly used in combination with previously mentioned techniques to fill the smaller proportion of remaining missing values.

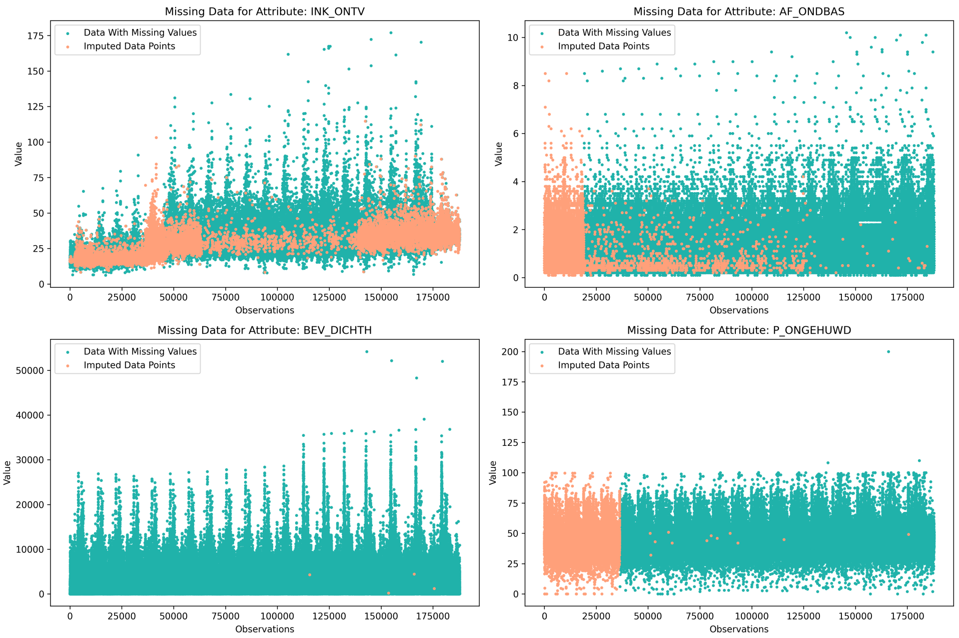


Figure 3: Imputation of the attributes: Income per Income Recipient (INK_ONTV), Distance to Schools (AF_ONDERBAS), Population Density (BEV_DICHTH), Percentage Unmarried (P_ONGEHUWD).

3.3 Feature Engineering and Selection

Target variable: Average House Price

The target variable *Average House Price* is obtained from CBS [3]. The dataset provides an overview of the average selling price paid for existing owner-occupied homes purchased by an individual during the reporting period. The average purchase price of a home reflects the average price of homes sold in a specific period in a particular region. These figures are based on the number of notarial transactions recorded each month by the Land Registry (Kadaster). Notary transactions are the price determined by the seller and buyer of the property without additional costs. Since different houses are sold each year, the characteristics of different houses are not taken into account. Therefore, it is different from the appraisal value (*Dutch: WOZ value*) or the price index. The appraisal value is used to determine the amount of taxes and levies, such as property tax. The valuation of houses to determine the WOZ-value is appraised by local authorities based on data of previous sales in that area. Average house prices should also not be misinterpreted as the average price index. The average price index reflects the development of the price and the economic growth of the real estate market [58].

Computation of Target Variable:
The derivation of the target variable specified for the regional dimension 'Neighborhoods' requires necessary computations to adjust the values to a different scale. The data on average house prices have been collected at the municipal level. However, our objective is to predict the average house prices at neighborhood level. The average prices for a neighborhood are derived by scaling the average appraisal values in a neighborhood to the average house prices in the overarching municipality for each year.

Definitions and Initial Assumptions
In the data, there exist two levels of regional dimensions: municipalities (m) and neighborhoods (n). The average house price is obtained on municipal level m for each year $i \in I$, where $I = \{2004, \ldots, 2022\}$.

Let M be the set of all municipalities and N the set of all neighborhoods. A municipality m_k consists of a subset of neighborhoods N_l.

$$M = \{m_1, m_2, \ldots, m_{7755}\} \tag{9}$$

$$N = \{n_1, n_2, \ldots, n_{14430}\} \tag{10}$$

$$m_k \subseteq N, \quad \forall m_k \in M \tag{11}$$

$$N = \cup_k^{7755} m_k \tag{12}$$

$$m_k \cap m_l = \emptyset, \quad \forall k \neq l \qquad\qquad I = 2004, \ldots, 2022 \tag{13}$$

The appraisal value (WOZ-value) and the average house price are defined as:

- WOZ Value: Given for each neighborhood n_l at year i, denoted as $\text{WOZ}_{n_l,i}$.
- Average House Price: Given for each municipality level m_k at year i, denoted as $\text{AvgPrice}_{m_k,i}$.

Mathematical Derivation
For each neighborhood n_l, an initial estimate $\theta_{n_l,i}$ of the average price of a neighborhood n_l and for year i is derived using the following computation:

$$\theta_{n_l,i} = median\left(\frac{\text{AvgPrice}_{m_k,i}}{\text{WOZ}_{n_l,i}}\right) * \text{WOZ}_{n_l,i} \tag{14}$$

However, this initial estimate $\theta_{n_l,i}$ needs to be scaled again to ensure consistency on municipal level. A rescaled estimate $T_{n_l,i}$ for each neighborhood is computed, ensuring that the mean of the estimates within each municipality m_k is similar to the actual average price $\text{AvgPrice}_{m_k,i}$ of municipality m_k.

Using the initial estimate $\theta_{n_l,i}$, the target variable $T_{n_l,i}$ is derived:

$$T_{n_l,i} = \theta_{n_l,i} + \left(\sum_{n_l \in m_k} \text{AvgPrice}_{m_k,i} - \sum_{n_l \in m_k} \theta_{n_l,i} \right) * \frac{\theta_{n_l,i}}{\sum_{n_l \in m_k} \theta_{n_l,i}} \tag{15}$$

The term $\left(\sum_{n_l \in m_k} \text{AvgPrice}_{m_k,i} - \sum_{n_l \in m_k} \theta_{n_l,i} \right)$ calculates the discrepancy between the observed average price in the municipality and the mean of the initial estimates. The correction ratio $\frac{\theta_{n_l,i}}{\sum_{n_l \in m_k} \theta_{n_l,i}}$ scales the adjustment according to the proportion of the initial estimate within the total municipality, ensuring that the correction is distributed proportionally in the neighborhoods. Relative differences within neighborhoods are maintained within the municipality while adjusting the overall estimates to the average prices observed.

The trend of the obtained target variable T (average property prices) for the total of The Netherlands can be observed in relation to the appraisal values (WOZ) in Figure 4. Both measures generally increase over time, with some fluctuations. Influenced by the effects of the financial crisis, average property prices decrease while the appraisal value shortly continues to increase, followed by a period of decreasing prices. After 2014, average house prices recover and continue to increase rapidly.

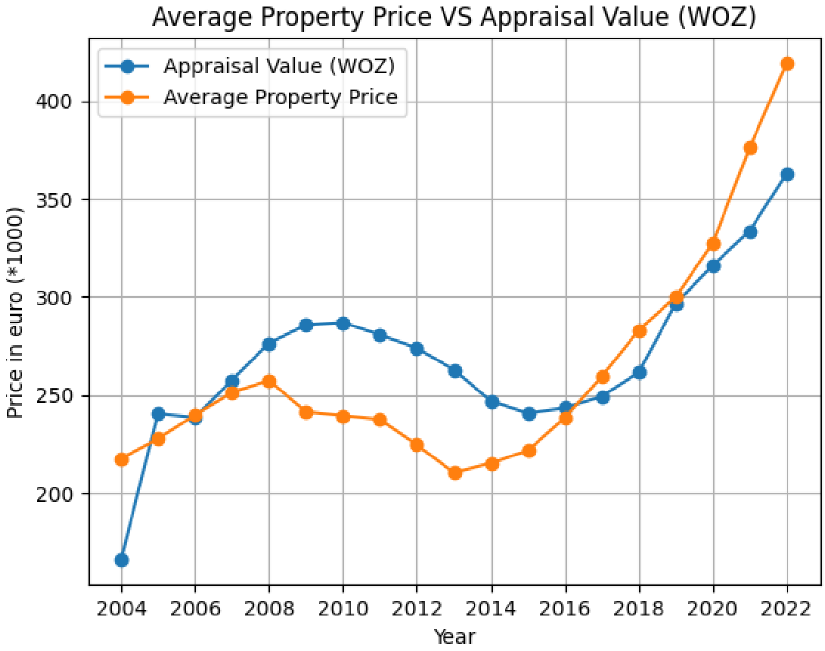


Figure 4: Average selling price of houses versus appraisal value of houses in the Netherlands from the years 2004 until 2022.

Multicollinearity

Multicollinearity is a condition in which there exists a high degree of correlation between two or more independent variables [59]. If multicollinearity involves a significant correlation between independent variables in a regression model, it can complicate the estimation and interpretation of the model coefficients. In linear regression models, the assumption of independence between the predictor variables will require removing multicollinearity. Other nonlinear machine learning techniques such as Neural Networks or Recurrent Neural Networks do not require reducing multicollinearity. In this research, none of the selected models has the assumption of independence between the predictor variables. However, it improves the efficiency of the models and reduces the feature space. Eliminating variables that exhibit a high degree of similarity to other variables simplifies the interpretation of the importance of individual variables. Therefore, it is helpful to identify variables with a high degree of multicollinearity to remove or combine these variables.

The technique we used is the Variance Inflation Factor (VIF) [60]. A VIF value is obtained for each variable by running a regression on the set of remaining variables given as input data. Iteratively deselecting the variable with the highest VIF value results in an independent subset of variables. The technique is applied as a feature reduction method. To maintain the separation among macroeconomic, socioeconomic, demographic and basic variables, the VIF method is applied solely within each subset to identify variables with high correlation. Only variables that had implicitly similar definitions in relation to the target variable were eliminated. For example, the variables "Arbeidsvolume werknemers" and "Arbeidsvolume zelfstandigen," which represent the amount of labor contributed by employees and self-employed individuals, respectively, expressed in labor years or hours worked, show a high correlation. Both variables communicate similar information concerning the target variable.

$$\mathrm{VIF}_a = \frac{1}{1 - R_a^2} \tag{16}$$

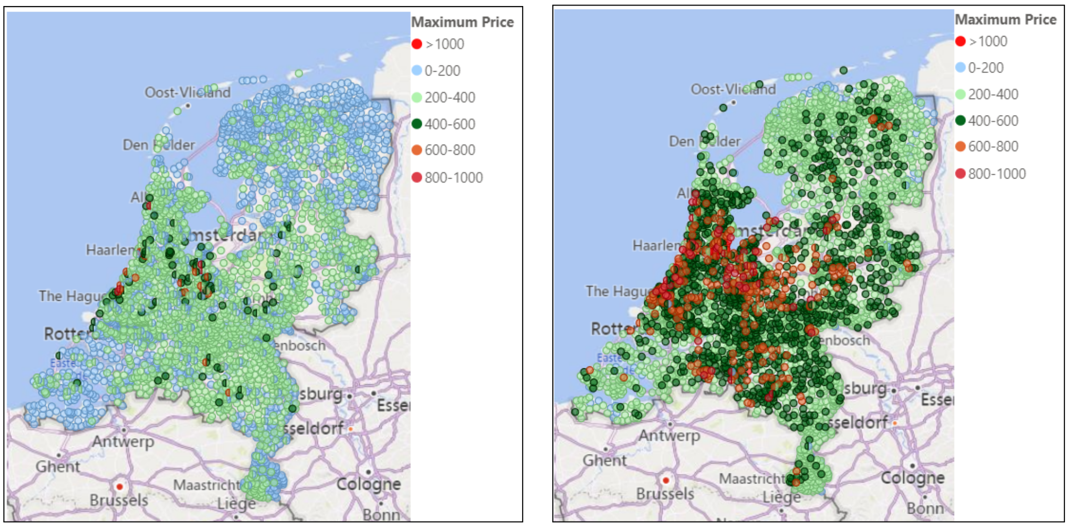
where R_a^2 is the coefficient of determination of attribute a from the regression of the i-th variable on all other variables.

3.4 Exploratory Data Analysis

In this subsection, an exploratory data analysis is performed on the data. The dimensions time and region are explored. In addition, the correlation between the features and the target variable "Average Housing Price" is analysed.

First, a heat map has been created showing the average house prices in the Dutch neighborhoods for the years 2004 and 2022 (Figure 5). It can be observed that there has been a price shift in nearly all neighborhoods over this period of time. Most neighborhoods in the Netherlands are shown in shades of blue and green for the year 2004, indicating a spread of relatively lower house prices up to 400 thousand euros. The northern part of the Netherlands, the provinces Groningen, Friesland and Drenthe and the southern province Zeeland are mostly blue shaded. In the province of Utrecht (in the center of the Netherlands) and along the coastline are more expensive neighborhoods such as Blaricum (631,923 euros) and Wassenaar (532,081 euros). The large cities Utrecht, Rotterdam and Amsterdam have a mix of expensive neighborhoods and cheaper neighborhoods.

For the year 2022, areas are shown in mostly in green, darkgreen and orange referring to a price scale between 200 to 800 thousand euros. This shows a clear trend of rising house prices across all regions. Similar to the year 2004, have the provinces Groningen, Friesland, Drenthe and Zeeland have cheaper house prices in relation to other provinces. The largest price shifts are visible in the larger cities Amsterdam, Utrecht, Rotterdam and several places along the coast, depicted by the orange dots. These cities had an increase in house prices of 174% (Amsterdam), 147% (Utrecht) and 149% (Rotterdam) compared to an average growth of 96% overall. This is also underlined by the research by Deelen et al. [61], who claimed that the cities Rotterdam, Amsterdam, Utrecht and The Hague obtained the highest price growth since the financial crisis ended in 2013.



(a) A heatmap of the Netherlands depicting the average house price (x1000) in a neighborhood in 2004

(b) A heatmap of the Netherlands depicting the average house price (x1000) in a neighborhood in 2022

Figure 5: Heatmap House Prices 2004 and 2022

The time dimension is an important factor in this research. Not all neighborhoods have existed since 2004 due to new-build projects or reclassification of neighborhoods by the municipality principal. By exploring the number of neighborhoods based on the consecutive years in which they exist, we can determine that only 3,999 out of 14,430 neighborhoods have existed for a full period of 19 years. While during this period, 10,431 neighborhoods have been been created or ceased to exist due to municipal annexation or redistricting. This research aims to predict the last two years, so the consecutive years are filtered by the neighborhoods that exist at least for the years 2021 and 2022. The number of neighborhoods and their respective length of existence in years is presented in Figure 6 where the red bars show the number of neighborhoods in relation to the length of the annual time series of neighborhoods which include at least the years 2021 and 2022. The orange bars show the number of neighborhoods for their respective length of existence in total.

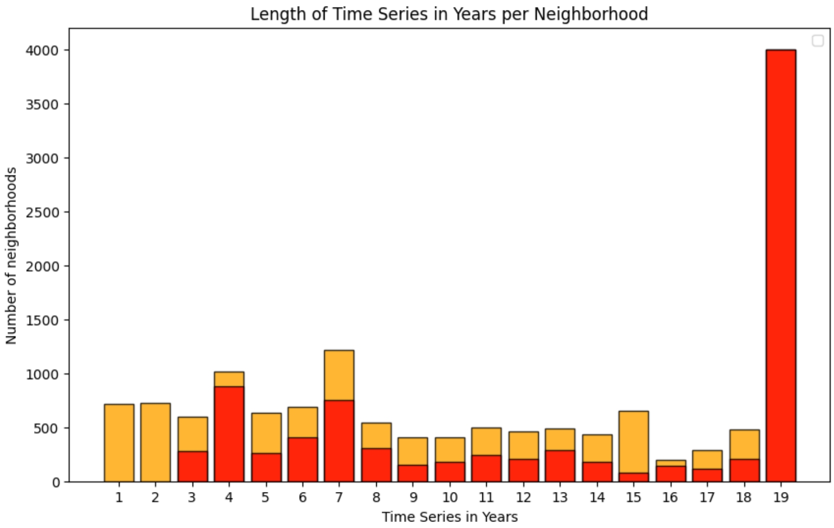
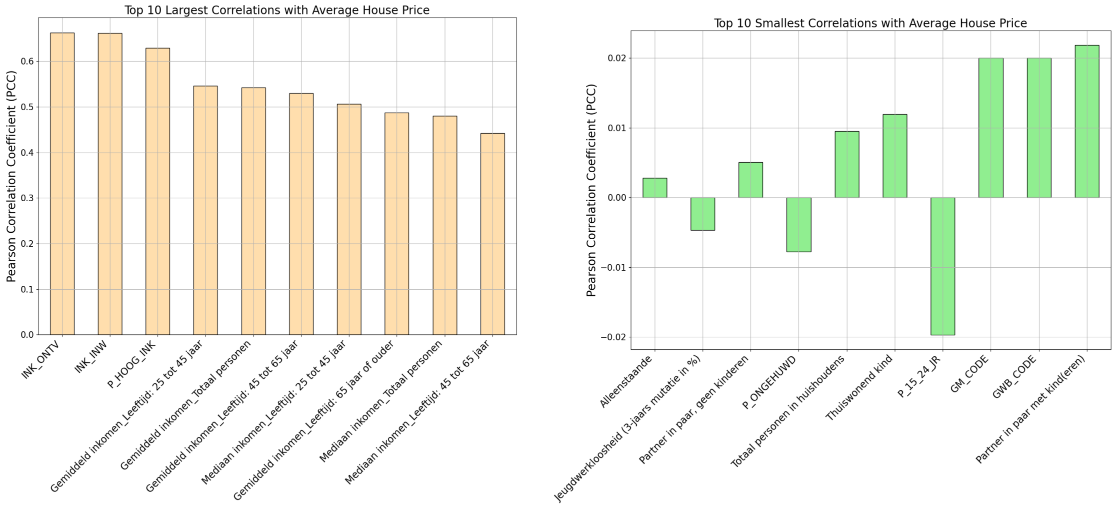


Figure 6: Length of Time Series in consecutive years for each neighborhood

In addition, correlations are explored with respect to the target variable. The identified features that are highly correlated with the ”average house price” are presented in Figure 7 (A), which shows the Pearson correlation of the ten largest (absolute) correlations. The variables shown consist completely out of income-related variables, which are all positively correlated with a Pearson correlation statistic larger than 0.5. The highest correlations corresponds to the variable ”INK_ONT” and ”INK_INW” with a correlation of 0.66 for both variables. In Figure 7 (B), the variables with the lowest correlation coefficients are depicted which are close to zero, indicating there exist no correlation. Many of the variables depicted are demographic variables that provide information about the household compositions: ”Alleenstaande”, ”Partner in paar, geen kinderen”, ”Totaal personen in huishouden”, ”Thuiswonend kind” and ”Partner in paar met kinderen”.



(a) Ten largest Pearson correlation coefficients between the target variable ’Average House Price’ and variables in the dataset

(b) Ten smallest Pearson correlation coefficients between the target variable ’Average House Price’ and variables in the dataset

Figure 7: Pearson correlation between the target variable ’Average House Price’ and features in the dataset

4 Methodology

The current chapter describes the research methods and modeling approaches of the forecasting frameworks needed to complete this study. This chapter provides a detailed examination of the experimental phases involved in the development of the methodology of the current study. There are four different machine learning models compared with the baseline model in predicting regional house prices. Each model is explained in sections 4.2, 4.4, 4.5, 4.6 and 4.7. Each model is tested and compared using four experimental datasets aiming to answer the research question. A complete data set that contains all variables and three reduced datasets with ablative subsets of socioeconomic, macroeconomic, or demographic features. The results will be provided in Section 5.

4.1 Dataset

This research examines the predictability of average house prices in a region based on discrete sets of macroeconomic, socioeconomic, demographic, and basic (spatial) variables. The prepared data (Section 3) were classified into four sets (Appendix 1.1).

- Complete Dataset
- Set A: Basic Variables + Demographic Variables + Macroeconomic Variables
- Set B: Basic Variables + Socioeconomic Variables + Macroeconomic Variables
- Set C: Basic Variables + Demographic Variables + Socioeconomic Variables

For the purpose of comparing the different machine learning models, the full dataset is used to predict the models initially. Subsequently, the sets A-C are ablated datasets, discarding either macroeconomic, socioeconomic, or demographic variables from the complete dataset. The complete dataset contains 78 features in total, while the reduced sets contain 68, 42 and 61 for sets A, B, and C, respectively. This results in four different approaches for each model, where the complete dataset and reduced datasets are tested. The predictive capability of the variable sets is evaluated using different machine learning models: XGBoost, Feedforward Neural Network (FNN), Recurrent Neural Network (RNN), and Long-Short-Term Memory (LSTM) Forecasting. Each dataset is split similarly into a training, validation and test set, where the models are trained on the historical data from the years 2004 to 2020. The models are evaluated on the validation set, which contains data of the year 2021 and tested on the data of year 2022.

Only neighborhoods that have historical data covering at least the years 2013 until 2022 are selected for time series modeling. A trade-off has been made between retaining a minimal number of time series data points and minimizing data loss. Given that a time series of 14 to 21 time points is considered a short time series [62], the minimal year 2013 has been established as the threshold year. Eliminating neighborhoods that do not reach this threshold, resulted in a decrease of 14,430 to 5256 neighborhoods. This implies a reduction of 39% observations in the dataset. The filtered dataset contains 77% of the neighborhoods with time series at full length and 15% between a sequence length of 14 and 19 yearly time points. The remaining 8% of the neighborhoods has a short time series between a sequence length of 10 and 14 time points. After splitting the data set into training, validation, and test data, the training set contains 86,874 observations, the validation set contains 5626 observations, and the test set contains 5626 observations. This dataset is used as the final dataset for each model.

Feature Scaling To equalize the contribution of numerical values between features, this research uses a feature scaling method to provide a fair contribution of each feature in the models. A single feature scaling method is applied to scale the data: Minimum-Maximum (Min-Max) scaling, also known as normalization. The technique is used to rescale the data to fit within a specific range between [0,1]. The presence of outliers can affect the scaling, leading to skewed data. Scaling is especially effective for the neural networks used in this study, where it can benefit the behavior of the model in several ways. It could lead to faster convergence of the optimal weights and biases during backpropagation, leading to a more efficient and stable learning process [63]. This reduces the risk of the vanishing or exploding gradient problem (Section 4.6). The network can adjust the weights more uniformly, preventing any single feature from disproportionately influencing the output due to its larger scale. Neural networks use activation functions (e.g. sigmoid, tanh, ReLU) that can behave differently depending on the input scale. Many activation functions, such as sigmoid and tanh, have effective ranges where they perform well.

Data Transformation Although, neither of the models evaluated during this study has the requirement to transform the data, it could be improving the predictions. The data contains valid extreme values which could affect the mean and variance of the data, leading to skewed results. Besides normalizing data, symmetrization of the data a useful method to manage skewed data. The Yeo-Johnson transformation is tested for each model whether is enhances the performance of the data. This transformation technique, a generalized Box-Cox transformation, aims to stabilize variance, reduce skewness, and alter the data to follow the normal distribution more closely. It is suitable for dealing with nonnormal data that include both positive and negative values. Unlike the more widely adapted standardization technique, where the inherent distribution of the data remains unchanged.

The Yeo-Johnson transformation is defined as follows for a given data point y and a parameter λ:

$$Y(\lambda) = \begin{cases} \frac{(y+1)^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \text{ and } y \geq 0 \\ \log(y+1) & \text{if } \lambda = 0 \text{ and } y \geq 0 \\ -\frac{(-y+1)^{2-\lambda}-1}{2-\lambda} & \text{if } \lambda \neq 2 \text{ and } y < 0 \\ -\log(-y+1) & \text{if } \lambda = 2 \text{ and } y < 0 \end{cases} \tag{17}$$

The 'PowerTransformer' class in Python automatically finds the optimal λ that maximizes the log-likelihood of the transformed data assuming a normal distribution.

4.2 Primary Baseline Model: ARIMA

The objective is to predict the average price of homes for the set of N neighborhoods using the ARIMA time series model. The AutoRegressive Integrated Moving Average (ARIMA) model, introduced by Box et al. [64] in 1970, is utilized for this purpose. The model is particularly useful for data with trends and seasonal variations. By modeling the dependencies in time series, it provides a clear baseline model for prediction based solely on time dependency of the target variable. Leaving out cross-sectional variables outlines the contribution of the cross-sectional attributes in machine learning models in addition to time series features. By employing this experimental setup, a useful comparison can be made between the performance of complex machine learning models and a simplistic statistical model with minimal input requirements.

The model consists of three statistical components: autoregression (AR), integrated (I), and moving average (MA). The mathematical formulation of the ARIMA model is expressed as ARIMA(p, d, q), where p denotes the order of the autoregressive part, d the degree of first differencing involved, and q the order of the moving average part. The derivation of each of the components result in the full model:

Autoregression: The autoregression component captures the dependency between lagged observations over time and the current observation. The current differenced observation $Y'_{i,n}$ is expressed by the successive lagged observations $Y'_{i-j,n}$ where $j \in 0, \ldots, p$, $i \in I$ and $n \in N$ (Equation 18). In case $p = 0$, then the AR component is excluded in the formula.

$$\phi(B)y'_{i,n} = \phi_1 y'_{i-1,n} + \phi_2 y'_{i-2,n} + \ldots + \phi_p y'_{i-p,n} \tag{18}$$

where $\phi_1, \phi_2, \ldots, \phi_p$ are the parameters of the AR part, p is the order of the AR process and B is the backshift operator, $B^k y_{i,n} = y_{i-k,n}$ for differencing the observations.

Differencing: Differencing the time series data is necessary to meet the assumptions of stationarity for the ARIMA model. Differencing, which is mathematically expressed as $\delta y_{i,n} = y_{i,n} - y_{i-1,n}$, removes the trend and seasonal component of the time series data. The number of times a differencing operator is applied is expressed by d.

The differenced series $y'_{i,n}$ is defined as:

$$y'_{i,n} = (1 - B)^d y_{i,n} \tag{19}$$

Moving Average: The moving average is a statistical method in time series analysis to smooth out short-term fluctuations and express long-term trends or cycles. A linear regression of the current observation $y'_{i,n}$ against the white noise of the consecutive q observations previous to $y'_{i,n}$. The MA part is written as:

$$\theta(B)\epsilon_{i,n} = \theta_1 \epsilon_{i-1,n} + \theta_2 \epsilon_{i-2,n} + \ldots + \theta_q \epsilon_{i-q,n} \tag{20}$$

where $\epsilon_{i,n}$ is the white noise error terms, $\theta_1, \theta_2, \ldots, \theta_q$ are the parameters of the MA part, and q is the order of the MA process.
Combining these components, the ARIMA(p,d,q) model can be written as:

$$(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p)(1 - B)^d y_{i,n} = (1 + \theta_1 B + \theta_2 B^2 + \ldots + \theta_q B^q)\epsilon_i \tag{21}$$

The model is trained on historical data from the years 2004 to 2020 and used to predict house prices for the years 2021 and 2022. For each region, the time series data of the target variable 'Average House Price' is selected and filtered. The data is normalized using min-max scaling.

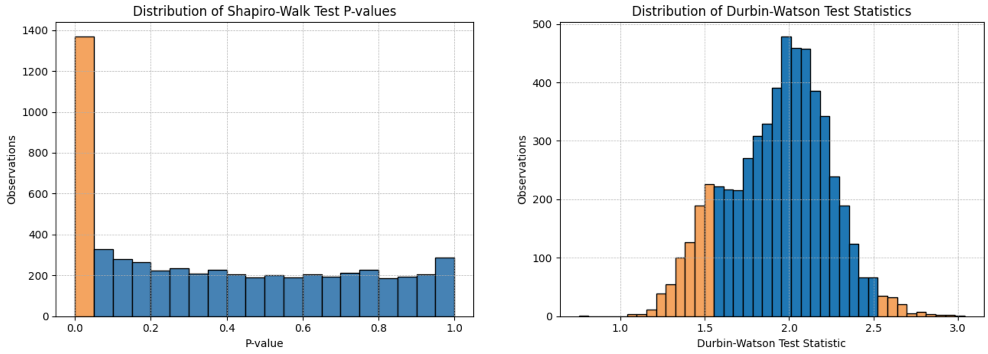
In Python, the 'auto_arima' [65] function models each region n in the refined dataset using the ARIMA(p,d,q) model by automating the selection of the best p, d, and q parameters. The function automatically determines and sets the differencing parameter d to obtain stationarity in the time series. Initially, it tests for stationarity with the default setting $d = 0$ using the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. Until stationarity is reached, d is increased by 1 and being tested. The function performs a stepwise search to find the parameters p and q that yield the lowest Akaike Information Criterion (AIC). The search space for p and q are both set to a range of $0, 1, 2, 3, 4, 5$. The optimal model is then applied to the training data to generate forecasts for 2021 and 2022. Once the time series for all neighborhoods n are modeled, it outputs the performance metrics (Section 4.8).

The ARIMA models assume stationarity, normality of the errors, and no autocorrelation. The stationarity is reached using the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test in the ’auto_arima’ model in Python. The normality of errors is tested using the Shapiro-Wilk test on each ARIMA model using the following hypothesis:

- Null Hypothesis (H0): The sample comes from a normally distributed population.
- Alternative Hypothesis (H1): The sample does not come from a normally distributed population.

If the p-value is less than 0.05, the null hypothesis is rejected, indicating that the residuals are not normally distributed. The results are shown in Figure 8. It can be observed that 24 percent of the ARIMA models have a p-value less than the significance level of 0.05, rejecting the null hypothesis.

Testing for the existence of autocorrelation after modeling ARIMA is performed using the Durbin-Watson test. The test statistic is calculated for each model. The test statistic ranges between 0 and 4, where 0 means a strong positive correlation, 4 means a strong negative correlation, and an approximate value of 2 indicates no autocorrelation in the residuals. The distibution of the Durbin-Watson test statistic can be observed in Figure 8. Here, 88 percent of the test statistic falls within a range between 1.5 and 2.5.



(a) Distribution of Shapiro-Walk Test P-Values (b) Distribution of Durbin-Watson Test Statistics

Figure 8: The distribution of p-values of the Shapiro-Walk test (a) indicating normally distributed residuals and the distribution of the Durbin-Watson test indicating autocorrelation of the ARIMA in-sample residuals.

4.3 Alternative Baseline Model: Hedonic Pricing Method

Alternatively, hedonic pricing methods (Section 2.3) are also considered as a baseline model. Its ability to provide detailed information on the contribution of each attribute in determining the price is a valuable property. The hedonic pricing method underlines the differences with the ARIMA-model, which captures time dependency but leaves out cross-sectional data. However, there exists a nonlinear relationship between the features and the prediction variable, making it not particularly well suited for statistical models such as the Ordinary Least Squares (OLS) and Generalized Least Squares model (GLS) that inherently requires a linear relationship between the independent variables and the dependent variable. Nevertheless, the hedonic baseline model was developed to provide an additional point of reference for evaluating the primary baseline model. By documenting the existence and consideration of the second baseline model, we ensure transparency in our modeling process and provide a comprehensive understanding of our approach and decision-making rationale.

Generalized Least Squares model (GLS) is considered most suitable for the specific nature of our dataset, having the ability to model observations with a form of heteroscedasticity and correlated variables. While also managing to deal with the computational limitations that occur when modeling with a large dataset. Therefore, the generalized least squares model has been evaluated as an alternative of the ARIMA model. The ARIMA model has been selected as the primary baseline model. Its simplicity and easy interpretation make it a suitable baseline model while the hedonic pricing method is more complex. Also, the interpretation of the explanatory variables in the hedonic pricing model become less reliable when the linearity assumption is not met.

Since the dataset contains time series data, we apply a general least-squares model with an autoregressive component (GLSAR). The residuals are modeled as an autoregressive process incorporated in the error terms.

The mathematical representation of the GLS model can be described as follows:

$$Y = X\beta + \epsilon$$

Where:

Y is a vector of observations.
X is a matrix of explanatory variables.
β is a vector of coefficients to be estimated.
ϵ is a vector of error terms with $\epsilon \sim N(0, \Sigma)$.

The GLS model assumes: $\epsilon \sim N(0, \Sigma)$

Where Σ is a known positive definite covariance matrix. The GLSAR model incorporates an autoregressive structure in the error terms:

$$\epsilon_t = \rho\epsilon_{t-1} + \delta_t, \quad \delta_t \sim N(0, \sigma^2)$$

Where ρ represents the autoregressive coefficient and δ is the innovation term, assumed to be white noise with zero mean and constant variance.

The estimation in GLSAR involves estimating both the regression coefficients β and the autoregressive coefficient ρ. This is often done iteratively using the estimator:

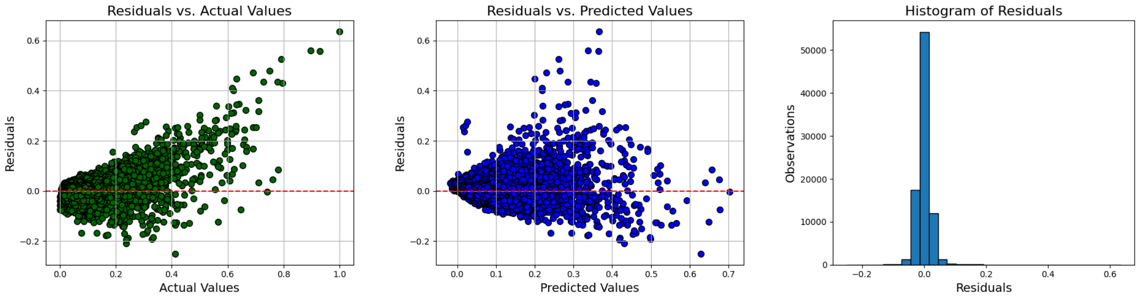
$$\hat{\beta}_{GLS} = (X'\Sigma^{-1}X)^{-1}X'\Sigma^{-1}Y$$

The weight matrix Σ^{-1} is adjusted in order to obtain a more constant variance of the error. The weight matrix is adjusted by the inverse of the predicted variance.

$$\Sigma_i^{-1} = \frac{1}{\hat{\sigma}_i^2}$$

The explanatory variables in the hedonic pricing method consist of the complete feature space (78 features) and the response variable ”Average Regional Price”. The model parameters are estimated on the training set and evaluated on the validation and test set. These results are evaluated and compared against the ARIMA baseline model only.

The GLSAR model assumes linearity, independence of errors, and normality of errors. These assumptions are evaluated using the residuals depicted in Figure 9. In plot (a) and plot (b) it can be observed that there exist and increase in the spread of the data, indicating heteroscedasticity. The residuals in plot (a) show a slightly increasing trend, showing more outliers towards the larger actual observations. This might be caused by nonlinearity of the data but it cannot be concluded. The obtained Durbin-Watson statistic is 2.25 indicating that the GLSAR model has efficiently excluded the autocorrelation. The last assumption is validated by plot (c) in Figure 9. The errors are normally distributed around zero.



(a) Residuals vs. Actual Observations

(b) Residuals vs. Predicted Observations

(c) Histogram of Residuals

Figure 9: The two subplots (a) and (b) show a scatter plot of the residuals plotted against the actual and predicted observations using GLSAR. The distribution of the residuals is depicted in a histogram in subplot (c).

4.4 Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting, or XGBoost [66], is a machine learning algorithm that employs the boosting ensemble technique. It sequentially optimizes decision trees to correct previous trees' errors, minimizing the objective function which contains a loss function and a regularization term.
This approach ensures an enhancement in computational efficiency, by employing a sparsity-aware algorithm, which is optimized for handling sparse data and missing values. It also supports parallel and distributed computing, enabling faster training on large datasets. Its ability to handle both linear and nonlinear relationships, and effective in-built regularization techniques, make XGBoost a robust tool handling data efficiently. Having a transparent design, the model is capable to explain its predictions and provide an overview of the dominant features in the feature space.

Each dataset is trained separately in the model to achieve the best results for each dataset. The model does not make any assumptions on the data, other than having only numerical input values. Since the model is capable of handling outliers well, the data do not necessarily have to be transformed. However, it can improve the performance. A Yeo-Johnson transformation is, therefore, tested to validate whether it improves the performance. The datasets are normalized using min-max scaler, followed by a train, validation and test split.

Hyperparameter tuning takes place using grid search. This technique searches all possible hyperparameter combination in the grid and evaluates all combinations by a chosen objective function. The optimal combination is then returned. The setup for each gridsearch is provided in Table 1. There exist many hyperparameters for the XGBoost model, but only a selection of parameters is optimized, as these intent to make the most impact in the performance of the model.

Hyperparameters:

1. *Booster:* The boosting algorithm to use. Options include Gradient-Boosted Trees and DART.
2. *Learning rate*: The step size used at each iteration to update the weights.
3. *Max tree depth*: The maximum depth allowed for each tree.
4. *Maximum number of trees*: The maximum number of trees to include in the ensemble (number of boosting rounds).
5. *Subsample ratio*: The fraction of the training data used to grow each tree.
6. *Columns subsample ratio for trees*: The proportion of features randomly sampled to build each tree.
7. *Columns subsample ratio for splits/levels*: The proportion of features randomly sampled for each split in a tree.
8. *Early Stopping*: A method to terminate training early if there is no improvement, to avoid overfitting.
9. *Early stopping rounds*: The number of rounds without improvement before stopping training.
10. *Gamma*: The minimum reduction in loss required to make a further split.
11. *Minimum child weight*: The minimum sum of instance weights needed in a child node.
12. *L1 regularization*: The L1 penalty term added to the objective function for regularization.
13. *L2 regularization*: The L2 penalty term added to the objective function for regularization.
14. *Max delta step*: The maximum step size limit for weight optimization.
15. *Tree method*: The algorithm used to find the best split. Options include Exact, Approx, Histogram, and Automatic, with Automatic selecting based on heuristics and dataset characteristics.

The remaining hyperparameters are considered less influential on the outcome of the model and are therefore fixed to reduce training time. The hyperparameter *subsample ratio* is set to 1. Setting the hyperparameter, for example, to 0.5 means that XGBoost would randomly sample half of the training data prior to growing trees, preventing the model form overfitting. However, our initial assumption is that the model is not overfitting, so the parameter is set to 1. For similar reasons, the parameters *L1* and *L2 regularization* is left out. In case overfitting takes place, it will be manually adjusted after gridsearch has been performed. The hyperparameter *early_stopping_rounds* is not initiated. Gridsearch is performed on various boosting rounds, searching for the optimal number of boosting round. Interference of the early stopping parameter is not necessary, since the optimal model is analysed after gridsearch and further optimized

Parameter	Values
n_estimators	200, 500, 700, 800, 1500
learning_rate	0.01, 0.05, 0.1
max_depth	3, 5, 7, 9
gamma	0, 0.1, 0.2
colsample_bytree	0.3, 0.5, 0.7
colsample_bylevel	0.3, 0.5, 0.7, 1
min_child_weight	1, 2, 3, 5, 10, 15, 20, 30

Table 1: Grid Search Options for XGBoost Model with Tree Booster

manually. The *max_delta_step* parameter is set to zero, which means that it leaves the output of the leaves unconstrained. Usually, adjusting this parameter is unnecessary. Lastly, the *tree_method* parameter is set to 'auto', which means that it automatically selects the most appropriate method. Based on the constraints of the dataset, the model chooses between the methods Exact, Approx, Histogram. Since our dataset contains many observations, it is likely that the model automatically chooses the greedy 'Approximate' or 'Histogram' methods which use less computational resources and efficiently find the suboptimal splits.

Parameter	Fixed Values
booster	gbtree
subsample	1
early_stopping_rounds	Not Initiated
max_delta_step	0
tree_method	auto

Table 2: XGBoost fixed (hyper)parameters

Extreme gradient boosting is not only a suitable method because of its robustness and efficient data handling, but one of the strengths of XGBoost is its ability to provide insight into the importance of features, which helps to understand the impact of each feature on the model predictions. Feature importance can be measured using three different methods: gain, cover, and frequency.

- Gain: The gain is measured by the contribution of a feature to minimizing the loss or improving the accuracy by the branches it is on. It represents the average gain over all splits.
- Cover: This metric counts the number of times a feature is used to split the data across all trees, weighted by the number of observations that are affected by the split. It reflects how frequently a feature is used in the model.
- Frequency: The measure counts the number of times a feature is used to split the data across all trees. It can be considered as the raw count of splits involving the feature.

The results can be observed in section 5.5.

4.5 Feedforward Artificial Neural Network (FNN)

The feedforward artificial neural network also known as the multilayer perceptron (MLP) is a deep learning model characterized by the neuron-based framework that is often compared to the structure of human brains. The framework consisting of multiple layers with neurons is designed to pass information forward from the input layer to the output layer via a set of one or more hidden layers. The in- and output layers are dependent on the data and prediction task of the model. The two-dimensional input data are determined by the batch size and the number of features: (*num_features*, *batch_size*). And depending on the origin of the prediction number of output nodes is configured, which is singular for a regression task. The number of hidden layers and the size of the hidden layers is unrestricted and often empirically determined. Increasing the number of layers and the number of neurons increases the complexity of the model, resulting in a higher risk of overfitting the training data. Regularization methods such as L1 or L2 regularization can reduce the risk of overfitting by structurally adding penalties based on the magnitude of the weights. Similarly, the function 'dropout' can be used to randomly set weights to zero, disrupting the models' weight optimization from getting stuck in a local minimum.

Hyperparameters:

1. *Activation Function*: A function that is applied to each neuron in a layer. Activation functions include ReLU (Rectified Linear Unit), Sigmoid, Linear and Tanh.

2. *Learning Rate (lr)*: A hyperparameter that controls the step size during the optimization process. It determines how much the weights are updated during training.

3. *Loss Function*: The function that measures the difference between the predicted output and the actual target. Common loss functions include Mean Squared Error (MSE) or Mean Absolute Error (MAE) for regression.

4. *Optimizer*: The algorithm used to adjust the weights and biases to minimize the loss function. Common optimizers include (Stochastic) Gradient Descent and Adam.

5. *Epochs*: The number of times the entire training dataset is passed forward through the neural network.

6. *Batch Size*: The number of training examples used in one iteration of updating the weights. It determines how many samples are processed before the model's parameters are updated.

7. *Dropout Rate*: A regularization technique where a proportion of neurons are randomly set to zero during training to prevent overfitting. The dropout rate specifies the fraction of neurons to drop.

8. *Regularization*: A L1 or L2 regularization adding penalties to the loss function adjusting the weights.

The model structure is configured empirically, with assistence of the Random Search algorithm that randomly validates constrained choices of hyperparameters and returns the best model (Table 3).

Determining the architecture of the feedforward neural network is a complex process. There exists no fine outline on establishing the optimal architecture of the network. In general, deeper and complex models tend to acquire more information from the data, but this also increases the risk of overfitting the data. The number of layers and the corresponding neurons per layer were determined experimentally. Incrementally increasing the complexity of the model, an optimal structure was achieved for each model. The models are configured and trained on four distinct datasets, which are transformed using the Yeo-Johnson transformation. The Yeo-Johnson transformation is applied to the training data, reducing the impact of extreme values in the data, and fitted to the validation and test data. Subsequently, the data is normalized. Experimentally, a initial configuration of the number of layers and number of nodes is found by fixing the remaining parameters. The loss function MSE is chosen, because it gives more weight to larger errors than smaller ones, which means it penalizes outliers more severely. Additionally, the Ridge regularizer which performs well in combination with a MSE loss function, making the combined objective function smooth and easier to optimize. Then, step-by-step, the complexity of the model is increased, by adding more layers or increasing the number of nodes in a layer. The predictions and evaluation metrics are observed in the loss curve to intuitively determine if the complexity of the model should be increased. After an initial set-up of the number of layers and neurons per layer is established, the Random Search algorithm is utilized to optimize the hyperparameters (Table 3). Again, a final empirical update on the model is performed after random search to finalize the model.

For the importance of comparing the different models equally, the dataset used for the other models is used similarly to train and make predictions using the feed-forward neural network. The reduced and full versions of the dataset are trained separately on the individually trained models. Although the model has no specific assumptions on the data, it generally performs better on a normalized dataset. So, the data are normalized using a min-max scaler. In addition, the Yeo-Johnson transformation will be employed experimentally. The train-validation-test split results in a training set of 86,874 instances, a validation set of 5626 instances, and a test set of 5626 instances.

Parameter	Search Space
Activation	['linear', 'relu', 'sigmoid', 'tanh']
Regularization	[0.01, 0.001, 0.0005]
Dropout	[0.01, 0.001, 0.0005]
Optimizer	['sgd', 'adam']
Learning Rate	[0.005, 0.001, 0.0005]

Table 3: Hyperparameter Search Space for Neural Networks

4.6 Recurrent Neural Network (RNN)

Building further on the functional architecture of neural networks, Jeffrey Elman introduced a recurrent neural network structure with backpropation [67]. Unlike traditional feedforward neural networks, RNNs have connections that form directed cycles within the hidden layers, which allows to maintain a hidden state that can capture information about previous inputs. This makes them particularly well-suited for tasks where the order of data is important, such as time series analysis and natural language processing.

The model structure that is most appropriate for time series analysis in this study is a sequence-to-one architecture. The model receives a sequence of data $(x_0, ..x_T)$ as input and returns a single regression value as output $\hat{y}$ (Figure 10). Each input sequence consists of a single region with sequentially ordered features. To handle input data with different sequence lengths (10 - 19 time points), a technique called data padding is used to extend shorter sequences to the same length as longer sequences by adding zeros prior to the shorter sequences.

Training recurrent neural networks is considerably harder than training a multilayer perceptron due too its 'deep' nature. When training a deep neural network, the gradients updating the weights have the tendency to either vanish or explode. This phenomena is called the vanishing or exploding gradient problem. The vanishing gradient problem occurs when the gradients become close to zero during backpropagation, causing the weights to remain unchanged [68]. This holds that the weights closer to the input layer are not updated anymore. A similar problem occurs when the gradients become too large, causing the weights to 'explode' [69]. The dominance of highly weighted nodes in the model becomes unproportionally large, ultimately disrupting the balance of the network irreversibly. Measures are taken to reduce the risk of the vanishing or

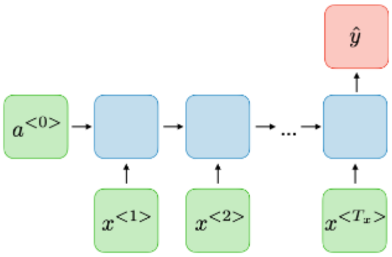


Figure 10: Sequence-to-one architecture of a recurrent neural network

exploding gradient problem. A xavier weight initialization is applied drawing the initial weights of the network from a uniform distribution preserving the constant variance of the activations. Since the variance remains stable, the gradients maintain a stable flow throughout the network. Additionally, gradient clipping is an effective measure to prevent exploding gradients by setting a maximum gradient value.

The model is built using empirical strategies and by applying the random search method. The number of layers and hidden nodes in each layer is determined experimentally using the adaptive optimizer Adam, with 30 epochs and a batch size varying between 32 and 4096. Orthogonal weight initialization is used to initialize the recurrent kernel weights, with Xavier initialized regular weights. Evaluating the performance metric and plots of the predictions lead to the final number of layers and hidden nodes. Hyperparameter tuning was executed using a random search algorithm (Table 4). Except for the hyperparameter gradient clipping, has the recurrent neural network exactly the same hyperparameters as the feedforward neural network (Section 4.5).

The recurrent neural network requires sequential input data of equal length. Since the sequences vary in length in out data, ranging from 10 up to 19 time points, the technique 'Padding' is applied. This updates the length of each time series to the maximum length of the time series by appending zeros to the sequences. Additionally, the data are transformed to be more Gaussian distributed using the Yeo-Johnson transformation, stabilizing the variance and reducing the inconsistency of the data which migitates extreme initial gradient behavior during backpropagation. Furthermore, the data is normalized using the Min-Max scaler and split into train, validation, and test data.

Parameter	**Search Space**
Activation	['elu', 'relu', 'sigmoid', 'tanh']
Regularization	[0.01, 0.001, 0.0005]
Dropout	[0.01, 0.001, 0.0005]
Learning Rate	[0.005, 0.001, 0.0005]

Table 4: Hyperparameter Search Space for RNN and LSTM

4.7 Long Short-Term Memory (LSTM)

Challenging the issues of vanishing and exploding gradient, the model Long Short-Term Memory (LSTM) was introduced by Sepp Hochreiter and Jurgen Schmidhuber in 1997 [70]. A sequence model similar to the recurrent neural network by Elman (Section 4.6) having the ability to memorize time dependencies. Nevertheless, the architecture of the model differs, featuring a cell unit that serves as long-term memory storage and a hidden state that functions as a memory controller. These two elements are connected in a gated circuit where the information is controlled using input, output, and forget gates (Figure 11). It shows the processing of information at time step t. During each time step, the vectors x_t, h_{t-1}, and c_{t-1} are input to the LSTM cell. Using the three gates, the LSTM cell essentially makes three decisions that control the output to the next cell at t+1. The first decision made by the model is whether the previous state (c_{t-1}) should be retained in the current cell state (c_t), which is decided at the forget gate. Secondly, the input gate and candidate memory cell determine whether new information should be stored in the cell state c_t. Finally, the output gate decides which part of the cell state c_t should be forwarded to the next hidden state h_t. Advances of this model compared to Elman's recurrent neural network is that the fully connected layers at time t can run in parallel, effectively saving more time. In addition, the flow of information through the different gates controls the flow of long- and short-term information, conquering the problem of vanishing and exploding gradients.

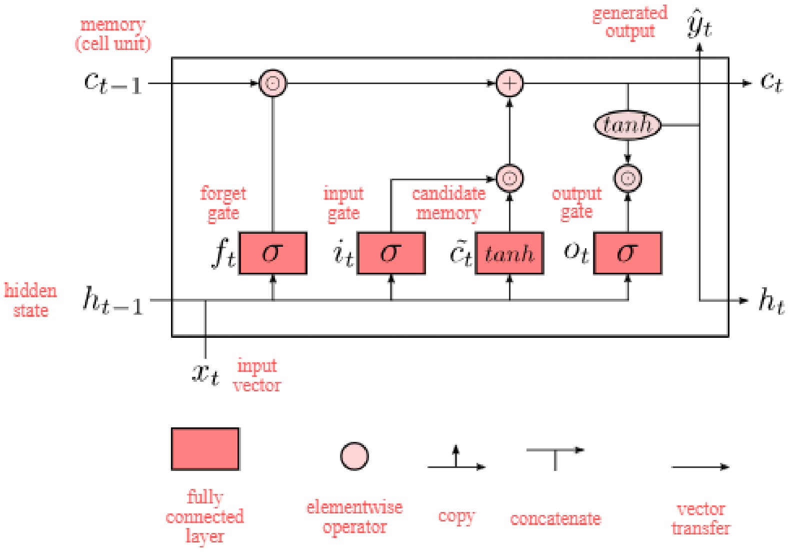


Figure 11: LSTM architecture

An equal approach to constructing the recurrent neural network is applied to develop the LSTM model. Empirical techniques are used to determine the number of layers and neurons per layer. Using the adaptive optimizer Adam, having 20 epochs and a batch size of 16, layers and neurons are experimentally determined by incrementally increasing the complexity of the models. The random search algorithm is implemented to identify the most well-fitting hyperparameters. Orthogonal weight initialization is used to initialize the recurrent kernel weights, whereas regular weights are initialized using the uniform Xavier initialization. The biases are initialized with zeros. Sigmoid activation is set as the recurrent activation function, and the output activation is set as the hyperbolic tangent activation function.

An identical data preparation took place for the LSTM as for the RNN in the previous paragraph. The padded sequences resulted in four datasets with 5626 sequences of 19 time points. These sequences are normalized and transformed by applying the Yeo-Johnson transformation. The first 17 points represent the training data and the remaining data points are used as validation and test data.

4.8 Evaluation Metrics

Adopting measures of criteria on the performance is necessary to evaluate the selected models. As this research satisfies a regression task, the performance metrics are based on the distance between the observed values y and the predicted values $\hat{y}$. The evaluation metrics used in this study are mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE) and the coefficient of determination R^2. To compare the performance of the different models equally, all evaluation measures are computed on normalized datasets. The mean squared error is chosen as the loss function for all machine learning models to penalize outliers in the dataset. (Note: n in the following formulas is the number of observations, i is a single observation.)

Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{22}$$

This metric calculates the average absolute difference between the actual values (y_i) and the predicted values ($\hat{y}_i$).

Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{23}$$

This metric calculates the average of the squared differences between the actual values (y_i) and the predicted values ($\hat{y}_i$). It is more sensitive to outliers, giving larger errors a higher penalty than more smaller ones.

Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{24}$$

RMSE is the square root of MSE and provides an error metric in the same units as the target variable. It is also sensitive to outliers.

Mean Biased Error (MBE)

$$\text{MBE} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i) \tag{25}$$

MBE calculates the mean difference between the predicted and and actual values. It indicates whether the model tends to overestimate (MBE > 0) or underestimate (MBE < 0) the actual values.

Mean Average Precision Error (MAPE)

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{26}$$

The mean absolute percentage error (MAPE) is a measure that is used to evaluate the accuracy of a forecast. It is expressed as a percentage and provides an indication of how much the predictions deviate from the actual values. MAPE ranges from 0 to infinity. 0% indicates perfect accuracy, where values greater than 100% indicate a poor prediction with large errors.

R-squared (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{27}$$

R^2 measures the proportion of the variance in the dependent variable that is predictable from the independent variables and is also known for its "Goodness of Fit" measure.

5 Results

In this study, we conducted a comparative analysis of a variety of machine learning methods that were compared to a statistical baseline model. The models included are ARIMA, XGBoost, Feedforward Neural Network, Recurrent Neural Network, and a Long-Short-Term Memory model. Four different datasets are compared for each machine learning model. A complete dataset and three reduced datasets that do not include demographic, socioeconomic, or macroeconomic variables. The results of the baseline model ARIMA and alternative baseline model GLSAR are provided in Section 5.1 and Section 5.2, respectively. In Section 5.3, configurations of the four machine learning approaches are presented and evaluated. The results for the complete dataset are discussed in Section 5.4. Additionally, the reduced datasets are discussed in Section 5.4. In the final Section 5.5, we discuss the importance of individual features obtained from the Extreme Gradient Boosting algorithm.

5.1 Primary Baseline Model: ARIMA

In this section, we present the results of our ARIMA modeling approach applied to predict regional house prices in various neighborhoods throughout the Netherlands. The model parameters were estimated using historical data only for the target variable and evaluated on out-of-sample data of years 2021 (validation set) and 2022 (test set). The data contains sequences of consecutive years of at least ten observations with a maximum of 19 observations. Each neighborhood is predicted independently using ARIMA.

The results of all independently predicted neighborhoods are evaluated using the evaluation metrics described in Section 4.8. The results for year 2021 indicate a good fit of the model (Table 5). The mean absolute error, mean squared error, and root mean squared error values are relatively low. It can be observed that the RMSE is larger than the MAE indicating a larger impact of outliers on the goodness of fit. The error metrics for year 2022 are almost twice as large for the MSE and RMSE, indicating the predictions are less accurate for a larger time window. The goodness of fit and the accuracy of the forecast of the ARIMA models is evaluated using the MAPE and R^2 metrics. On the validation year, the mean average precision error and the R^2 have a result of 0.19 and 0.81, respectively. For year 2022, the MAPE and the R^2 have a result of 0.31 and 0.53, indicating a weak fit of the model.

Year	MAE	MSE	RMSE	MBE	MAPE	R2
2021	0.024	0.0009	0.030	-0.02	0.19	0.81
2022	0.045	0.0026	0.051	-0.04	0.31	0.53

Table 5: Performance Metrics on out-of-sample data

The modeling of ARIMA models in the various neighborhoods is evaluated using the Akaike Information Criteria (AIC) and Bayes Information Criteria (BIC). These evaluation metric interpret the goodness-of-fit against the complexity of the models. Low values indicate a better fit. The 'autoarima' function in Python applied a stepwise search for parameters p and q to optimize the AIC metric. On average, the ARIMA models obtained an AIC score of -98.5 and BIC score of -95.9. Both negative values indicate a good fit on average. As can be observed in Table 6, the maximum values obtained for both the AIC and the BIC score are negative, indicating a good fit of all models.

The shortest time series in the data that have a length of 10 consecutive years is analysed to provide information on whether these sequences affect the performance of ARIMA. In Appendix

1.5, the performance metrics are provided for 195 regions that have a training sequence of 8 consecutive years. It can be observed that these minimal time series obtain good results with a MAE and RMSE of 0.0467 and 0.0504 on the test data. In also explains the variance well with a R^2 value of 0.76. The proper results on these extremely short time series can be explained by the rather stable increasing trend of the time series data during these observed 10 time points.

	Average	Median	Min	Max	Observations
AIC	-98.50	-103.648	-148.471	-10.319	5626
BIC	-95.903	-100.672	-145.971	-6.987	5626

Table 6: AIC and BIC Statistics

5.2 Alternative Baseline Model: Hedonic Pricing Method

The results of the hedonic pricing model, Generalized Least Squares with Auto Regression (GLSAR), are provided in this section. The model parameters were estimated using the complete dataset of 78 features, 5626 neighborhoods, and 17 consecutive years. The model has been evaluated on the out-of-sample data of years 2021 (validation set) and 2022 (test set).

The results of the hedonic pricing model are evaluated using the performance metrics described in Section 4.8. The results for the year 2021 indicate a good fit of the model (Table 7). The error-metrics, mean absolute error, mean squared error and root mean squared error values, obtained relatively low values. These values rapidly increase for the year 2022, where the error-metrics are almost twice as large. The negative mean biased error (MBE) shows that the model is underfitting the actual observations. The mean precision error of 15.14% and 25.23% for the years 2021 and 2022 indicates not a good fit of the model. This can also be observed in Figure 12. The AIC and BIC statistics obtained on the in-sample data is -42,950 and -42,870, respectively.

In comparison to the primary baseline model ARIMA, the hedonic pricing method performs similarly well for the out-of-sample set of year 2021. The MAE, MSE RMSE and R^2 metrics are close to each other. The MAPE of the hedonic pricing model (0.15) is lower than the obtained MAPE of ARIMA (0.19). For the out-of-sample set of year 2022, the hedonic pricing model performs worse than the ARIMA model. The MSE and RMSE differ significantly, indicating that the GLSAR model finds it harder to predict extreme values. Additionally, the goodness of fit metric (R^2) is lower with an obtained value of 0.25 compared to the R^2 value of 0.53 for ARIMA.

Year	MAE	MSE	RMSE	MBE	MAPE	R2
2021	0.020	0.0010	0.032	-0.01	0.15	0.79
2022	0.043	0.0041	0.064	-0.04	0.25	0.25

Table 7: Performance Metrics on out-of-sample data

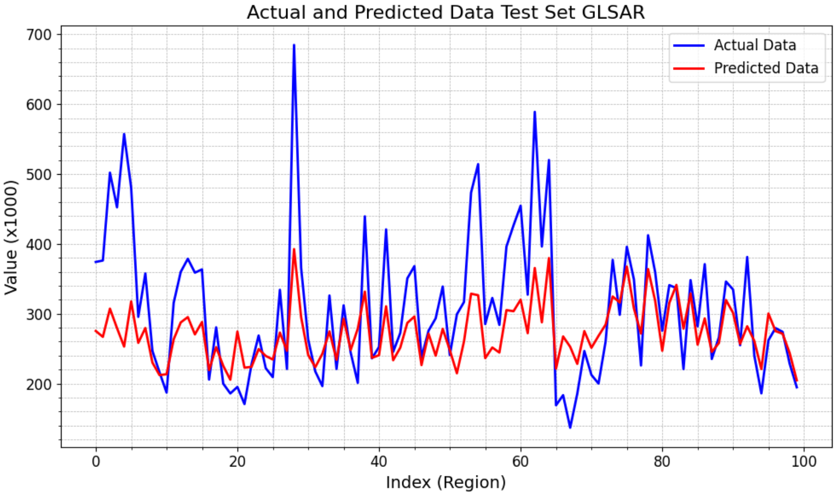


Figure 12: The predicted versus actual values for the year 2022 for the hedonic pricing model (GLSAR).

5.3 Machine Learning Models

The complete dataset including demographic, macroeconomic, socioeconomic and basic spatial variables of each neighborhood result in 78 features. The reduced datasets A (without socioeconomic variables), B (without demographic variables) and C (without macroeconomic variables) have 61, 45 and 68 features, respectively. The four machine learning models are trained on these datasets and compared using the selected evaluation metrics. In the following paragraphs, the training process of each model is described and the model architecture is provided. The results of each model are compared including the results of the baseline model.

Modeling

Extreme Gradient Boosting The XGBoost model was trained in Python on four distinct datasets consisting of 98,126 instances. To ensure consistency, each dataset was normalized using the Min-Max scaler. The scaler was fitted to the training data and subsequently applied to the validation and test datasets, preventing data leakage. A grid search was used to fine-tune the model's hyperparameters, optimizing its performance across a range of settings described in the methodology 4.4. This resulted in the final selection of the hyperparameters described in table 8. The default booster setting is 'Tree Booster'. Other fixed hyperparameters include the 'Max Delta Step', 'Subsample' and regularization parameters 'Lambda' for L2 regularization and 'Alpha' for L1 regualization. The maximum delta step is set to zero, as its impact on the outcome is negligible. The parameter 'Subsample' is set to 1, meaning it will sample the train data with a ratio of 1 each boosting round. As the regularization parameters alpha and lambda are preventing the model from overfitting, these parameters are initially set to default settings (Alpha = 0, Lambda = 1) during training to ensure a good fit on the training data. It is observed that the models are not overfitting on the training data, so these parameters are not adjusted.

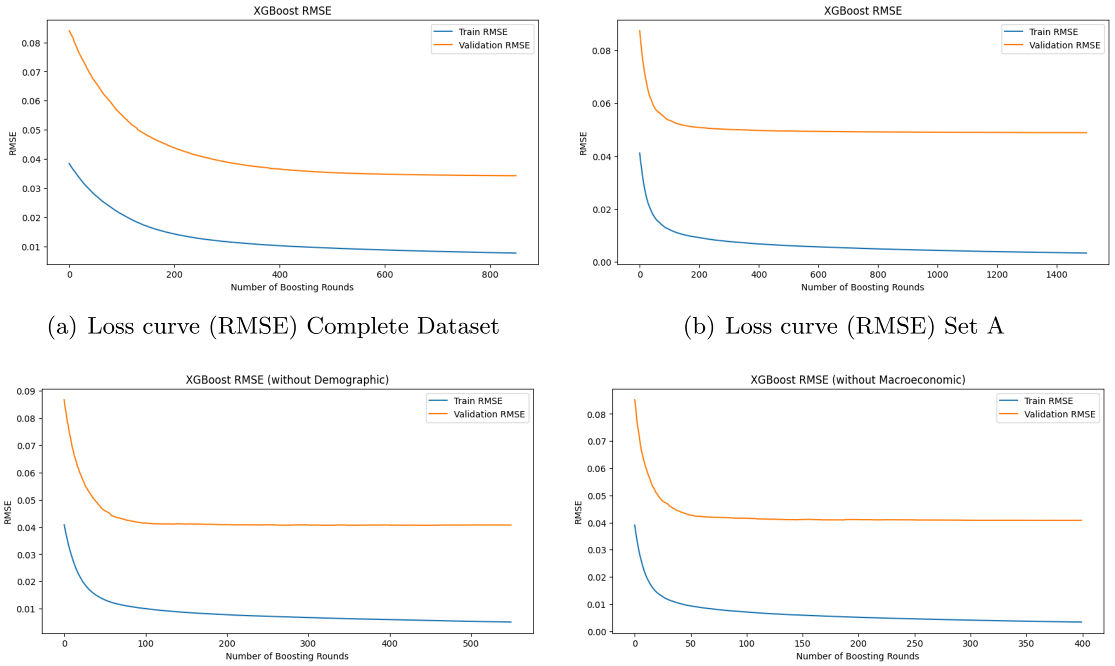
The final configurations of each distinct model are shown in Table 8 depict a variety of different settings for each parameter. It can be observed that the model belonging to dataset A has the most complex model having 1500 boosting round and high values for maximum tree dept, column sample ratios and minimum child weight. This could be explained by the absence of socioeconomic variables such as average income, which are important splits in the tree booster. Furthermore, the learning rate is increased for the reduced datasets. A lower learning rate in the complete dataset helps in slowly converging to a more accurate model, while a higher learning rate in reduced datasets may be used to compensate for the missing information and achieve faster convergence. Logically, the number of boost rounds decreases as the learning rate increases. Similarly, the maximum dept of the reduced models increase, indicating a need for deeper trees to capture the complexities of the reduced datasets. The column ratios are harder to interpret, with an inconsistent variation between the reduced datasets and the complete dataset. However, it is observed that set A has a col ratio level of 1, implying all columns are used for each split, compensating for the lack of socioeconomic data. To compromise overfitting of the training data, the minimum child weights for set A and C configured at high values of 15 and 20. The more complex model trained on set A is more prone to overfitting. Set C contains noisier data without macroeconomic variables, explaining the need for a more conservative model.

	Complete	**Set A**	**Set B**	**Set C**
Booster	Tree	Tree	Tree	Tree
Boosting Rounds	850	1500	550	400
Learning Rate	0.01	0.05	0.05	0.1
Max Tree Depth	7	9	9	9
Col Ratio Tree	0.3	0.3	0.5	0.5
Col Ratio level	0.5	1	0.7	0.3
Gamma	0	0	0	0
Minimum Child Weight	5	15	2	20

Table 8: Parameter settings of four Extreme Gradient Boosting models on the complete dataset and the reduced datasets: A (without socioeconomic variables) , B (without demographic variables) and C (without macroeconomic variables).

For each model, a smooth loss curve can be observed in figure 13 of the loss function RMSE on the training and validation data. Both the training and validation curves of each model start at a higher value around 0.04 and 0.09, and consistently decrease as the number of boosting rounds increases. The models trained on datasets A-C all converge quickly to a more stable value within 200 boosting rounds, and even under 100 boosting rounds for the model on dataset C, indicating rapid learning. The model on the complete dataset has a lower learning rate, demanding more boosting rounds to converge. A similar pattern is reflected on the corresponding training data of each model reaching nearly zero, indicating a good fit on the training data. The loss curves of models slowly decrease in validation loss until the loss increases again and the model starts to overfit. The models trained on the complete dataset and reduced sets B and C converge to a approximate RMSE of around 0.04, indicating a similarly good fit with respect to each dataset. The model trained on set A has a larger gap between the training and validation data indicating a slightly worse fit on the model on the validation set, also reflected by the results in Table 13. This could be explained by the lack of socioeconomic variables, decreasing the performance of the model.

Feedforward Neural Network Each model consists of an input layer, two hidden layers and an output layer with a single neuron. The input layer is determined by the batch size and the respective number of features for the given input data of each model. The number of hidden layers and their sizes reflect the complexity of the data. Increasing the number of layers led to equal or worse performances due too overfitting. The complete and reduced dataset C have more complex layer configurations (1000/50), indicating the need to capture more complexity in the data. In contrast, the models on reduced datasets A and B have smaller hidden layers (20/20 and 20/10), likely because these reduced datasets have less variance in the data. The



(a) Loss curve (RMSE) Complete Dataset

(b) Loss curve (RMSE) Set A

(c) Loss curve (RMSE) Set B

(d) Loss curve (RMSE) Set C

Figure 13: The training and validation loss curves of four Extreme Gradient Boosting algorithms, trained on the complete and reduced datasets.

choice of activation function has some small influence on the performance where the linear activation function turn out to be most effective for most layers. Applying the linear activation function obtained better results as it generated more variance in the output. The linear activation function is unbounded and produces values on the real line, compared to sigmoid and tanh activation functions. The learning rate, batch size and number of epochs all influence each other. A higher learning rate often leads to a faster decrease of the training loss, implying less epochs are needed. However, this is dependent on the batch size where large batch sizes generalize backpropagations more, while smaller batch sizes are more accurate. The Adam optimizer is initialized with a default learning rate of 0.001, but it adopts its learning rate dynamically based on historical gradients. The larger batch size of the complete dataset and reduced dataset A and B are chosen to create a more smoother convergence whilst keeping a similar performance as smaller batch sizes. The smaller batch size of set C has a better gradient update than larger batch sizes while still remaining a smooth convergence. The very small regularization rate in the reduced model on set C is applied for a slight regularization to control overfitting without affecting learning, while the other models do not need regularization, likely due too their larger batch size. Therefore, the dropout parameter is not activated as well.

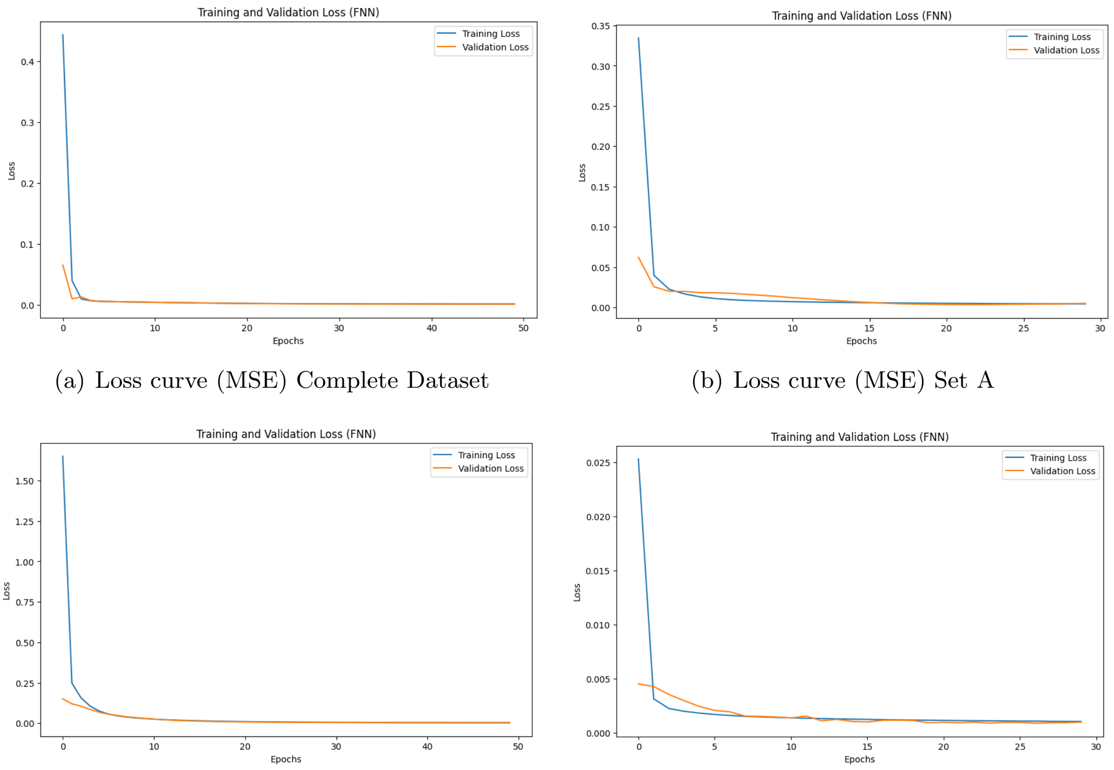
	Complete	**Set A**	**Set B**	**Set C**
Input Layer	78	61	45	68
Hidden Layers	1000/50	20/20	20/10	1000/50
Activation	Linear/Sigmoid	Linear/Linear	Linear/Linear	ReLu/Tanh
Regularizor	-	-	-	L2
Regularizer Rate	-	-	-	0.000001
Optimizer	Adam	Adam	Adam	Adam
Lr	0.001	0.001	0.001	0.0001
Epochs	50	50	30	30
Batch	4096	4096	4096	1024

Table 9: Feedforward Neural Network configuration for each model trained on a reduced dataset without socioeconomic, demographic or macroeconomic variables.

The loss curves presented in Figure 14 display the training and validation loss of four Feedforward Neural Network models trained on the complete dataset and the three reduced datasets. All loss curves show a similar pattern with a larger loss on the training data that decreases rapidly in the first few epochs and then stabilizes towards the last number of epochs. The rapid decrease in the training loss can be explained by a relatively high learning rate and the large batch size of each model. The training loss of the model on the complete dataset and the reduced dataset C starts at a relatively small MSE compared to the less complex models on dataset A and B. This can be partly explained by the complexity of the models, which is simpler for the models A and B, but the random weights initialization of the the model plays a important role too. Both the training and validation loss curves decrease rapidly and then flatten out after less than 10 epochs. The gap between the training data and the validation data of each model is minimal indicating a generalized and good fit of the model on the validation data. There is a slightly larger gap between the validation loss and training loss of the model trained on set A, which reflects the difficulty of training the dataset without socioeconomic variables.

Recurrent Neural Network Building the optimal recurrent neural network is a process that relies on a series of decisions. Each model's architecture and hyperparameters were specifically customized to optimize performance on their respective datasets. The datasets require slightly different preprocessing steps than the datasets used to train XGBoost and the fully connected neural network. The RNN and LSTM sequence models demand sequential input data, as mentioned in Section 4.6.

The first layer consists of an input layer of size (*batch size* * 17, *number of features*), followed by two recurrent layers for each model. Increasing the number of layers reduces the performance of the models, likely caused by the vanishing and exploding gradient problem. As can be observed in Table 10, the models on the complete dataset and the reduced dataset A perform well on a small number of neurons in the hidden layers. During training, this also led to good results for reduced sets B and C, indicating that a lower complexity of the model generally performs well on these datasets. However, a larger number of neurons in the hidden layers for models B and C are chosen because these structures were able to capture more complex patterns. Simultaneously, its architecture was caught overfitting the training data, so measurements were taken to make these models more robust. Both models on set B and set C utilize a Ridge regularization with parameters 0.00005 and 0.00001, respectively. Also, the batch size of sets B and C are increased stabilizing the gradient updates and leading to faster convergence. Set C stabilizes faster, requiring only 35 epochs, while the number of epochs of the model on set B requires a larger number of epochs (100). This is caused by the larger number of neurons, the regularization parameter, and the addition of a dropout parameter with 0.0005 to regularize



(a) Loss curve (MSE) Complete Dataset

(b) Loss curve (MSE) Set A

(c) Loss curve (MSE) Set B

(d) Loss curve (MSE) Set C

Figure 14: The training and validation loss curves of four Feedforward Neural Network algorithms, trained on the complete and reduced datasets.

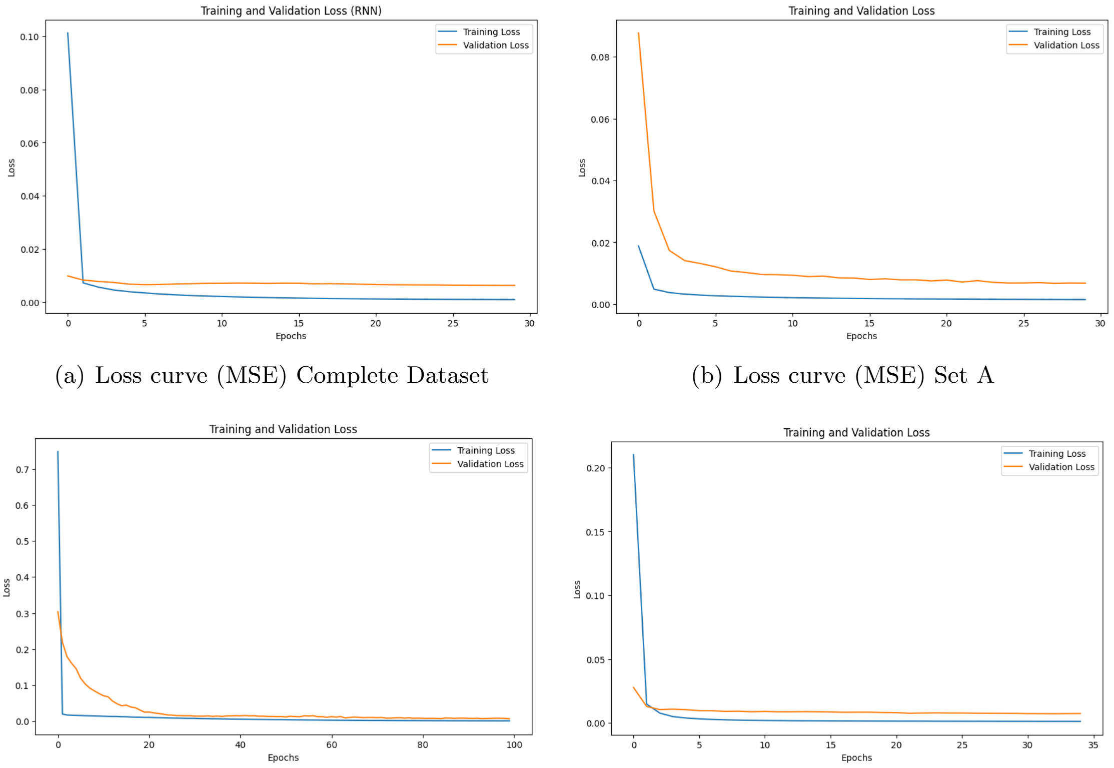
the model. The batch size for the complete model and Set A models remains at 16, suitable for their relatively simpler structures. The activation functions are obtained from the random search algorithm, where the Rectified Linear Unit (ReLu) and the hyperbolic tangent activation serve the purpose to introduce non-linearities in the network, while the sigmoid functions appear to stabilize the flow of data to the output node. The Exponential Linear Unit (ELU) is similar to the Rectified Linear Unit (ReLu), except that is handles negative values exponentially. The weights are initialized using the Xavier initialization, providing uniformly distributed weights. The model is compiled with the Adam optimizer initialized with the default learning rate of 0.001, using the mean squared error (MSE) as the loss function and the mean absolute error (MAE) as a performance metric. The optimizer is bounded using gradient clipping with a norm of 1, tackling the problem of exploding gradients.

	Complete	**Set A**	**Set B**	**Set C**
Input Layer	78	61	45	68
Hidden Layers	20/10	20/10	80/80	200/20
Activation	ReLu/Sigmoid	ReLu/Sigmoid	elu/elu	Tanh/Sigmoid
Regularizor	-	-	L2(0.00005)	L2(0.00001)
Optimizer	Adam	Adam	Adam	Adam
Lr	0.001	0.001	0.005	0.001
Epochs	30	30	100	35
Batch	16	16	128	64

Table 10: Recurrent Neural Network configuration for each model trained on the complete dataset and a reduced dataset without socioeconomic (Set A), demographic (Set B) or macroeconomic variables (Set C).

The learning curves of each model can be observed in Figure 15, where four graphs present the loss curves of the mean squared error (MSE) of the validation and training data. The loss curve of plot (A) on the complete dataset converges quickly and stabilizes after approximately 15 epochs, indicating effective learning without overfitting the training data. The validation gap between the training and the loss curve is narrow, showing that the model closely fits the training and validation data. The validation loss curve in plot (B) starts at a higher loss but then decreases to a stable and low loss of approximately 0.01. Having a similar configuration of the RNN architecture as on the complete data implies that the missing socioeconomic variables result in less effective learning of the model. The training losses in plots (C) and (D) have a higher initial training loss which is likely caused by the more complex models. Both training curves rapidly decrease and converge to a value near zero. The validation loss in plot (C) needs more epochs to converge but stabilizes after 20 epochs. Overall, the loss curves indicate well-trained models that perform consistently on both training and validation datasets, with no significant overfitting.

Long Short-Term Memory The LSTM model has an experimental setup similar to the neural networks mentioned above. The Long Short-Term memory model is a deep neural network that has a complex structure of itself. Therefore, the approach to establish the framework of each model starts with a simplistic model with a single layer and a minimum of 10 neurons. A small batch size of 16 was chosen, providing more accurate gradient updates and faster convergence of the weights and biases. Incrementally increasing the complexity of the models in combination with random search led to the following LSTM frameworks showed in table 11. The models range from a single deep layer to more complex architectures having 2 or 3 deep layers. A deeper network seems to performs better with larger number of input features. The number of neurons are small for each layer, ranging from 30 to 10. The Tanh activation function is used consistently for the activation of the output gate, while the Sigmoid activation is used for the activation of the forget gate as discussed in section 4.7. Each model tended to overfit the training data, so the Ridge (L2) regularization parameter was activated with a parameter of 0.00001 for each hidden layer in the network, except for the model of dataset A. Additionally, overfitting of the training data was prevented by decreasing the learning rate to 0.0005 or 0.0001 and decreasing the epochs to 60 to stop training before the model starts overfitting. The model on the complete dataset increased the number of epochs to 100 to obtain better results. The models on the reduced datasets A-C remained consistent at 60 epochs. Increasing the batch size larger than 16 sequences led to smoother learning curves, but also increased the validation loss leading to worse performances.



(a) Loss curve (MSE) Complete Dataset

(b) Loss curve (MSE) Set A

(c) Loss curve (MSE) Set B

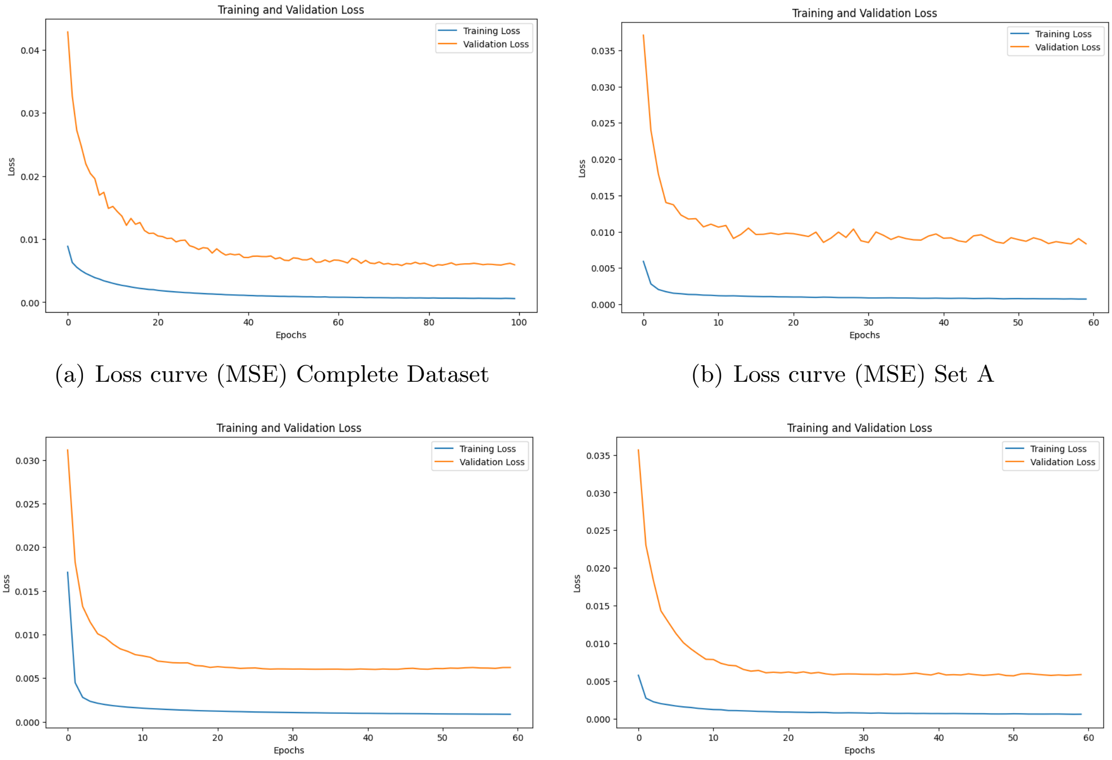
(d) Loss curve (MSE) Set C

Figure 15: The training and validation loss curves of four Recurrent Neural Network algorithms, trained on the complete and reduced datasets.

	Complete	**Set A**	**Set B**	**Set C**
Input Layer	78	61	45	68
Hidden Layers	20/20/20	30/30	30	30/10
Regularizor	L2(0.00001)	-	L2(0.00001)	L2(0.00001)
Optimizer	Adam	Adam	Adam	Adam
Lr	0.0005	0.0005	0.0001	0.0005
Epochs	100	60	60	60
Batch	16	16	16	16

Table 11: Long Short-Term Memory configuration for each model trained on the complete and reduced datasets without socioeconomic (Set A), demographic (Set B) or macroeconomic variables (Set C).

In Figure 16 the loss curves of the different LSTM models are presented, where the training and validation loss (MSE) is plotted against the number of epochs. It is noticeable that the loss curves (A) and (B) are less smooth and appear to be 'bumpy'. This is caused by the small batch size, which causes the gradients to update with more noise during backpropagation. In combination with larger learning rate, the loss curve of the validation data can become less smooth. The smaller batch size is also the reason that the loss curves all start with a relatively low MSE on both the training and validation loss. The loss curves all learn a bit slower due too the smaller learning rate overall. The model on the complete dataset learn slower than the other models, which is likely caused by the extra layer. Similarly, the learning curve of the reduced set B (C) learns faster having a single layer and a decreased learning rate. Overall, training the LSTM models require a smaller learning rate and a larger epoch size to obtain better results, compared to the recurrent neural network and the feedforward neural network.



(a) Loss curve (MSE) Complete Dataset

(b) Loss curve (MSE) Set A

(c) Loss curve (MSE) Set B

(d) Loss curve (MSE) Set C

Figure 16: The training and validation loss curves of four Long Short-Term Memory models, trained on the complete and reduced datasets

5.4 Comparative Results

In this section, the performance metrics and the final predictions of the models will be presented. First, the results on the complete dataset are provided. Then a comparison between the four experimental datasets is presented.

Complete Dataset

To evaluate the effectiveness of models in predicting average regional house prices, it is necessary to calculate statistical parameters to assess the model's ability to predict, as well as to compare the individual models to make it clear which nonlinear model is the most accurate in forecasting. Table 12 presents the calculated performance metrics on the "goodness of fit" in relation to the regressions of the ARIMA, XGBoost, FNN, RNN, and LSTM models.

The FNN model had the best performance on the test data based on all metrics, which are MAE, MSE, RMSE, MBE, MAPE, and R^2. The low MAE value indicates that the FNN model has high accuracy in predicting the output variables. Having a low RMSE tells us that the more extreme values are also predicted well, but this value being higher than the MAE means the error is larger due too more extreme values. The MBE of 0.0031 shows that the predictions of the models do not tend to overestimate or underestimate the actual values. The low MAPE of 10.42% and the high R^2 results indicate a good fit of the predicted data compared to the actual data and explaining high variance, which can be observed in Figure 17(B). The performance of the XGBoost model is second best, having significantly better results than the ARIMA model on the test data. It obtained lower RMSE and MAE values than the RNN and LSTM models, which perform nearly equally well on the complete dataset. However, the sequence models a better MAPE value, which means that the sequence models have more accurate prediction on average. All models outperform ARIMA with every performance metric on the test data. The baseline model ARIMA is outperformed on the validation data by the XGBoost and FNN model, but not by the RNN and LSTM models (Table 15). The XGBoost model performs best on the validation data, where it obtained the lowest MAE, MSE and RMSE. The FNN performs second best with larger the smallest MAPE value, but higher MAE, MSE and RMSE values. This suggests that the FNN model is more robust and regularized compared to the XGBoost model and ARIMA model. The RNN and LSTM models are also robust, having small deviations between the test and validation performances.

Figure 17 shows the ability of the models to predict the target value (vertical axis) for a subsample of 100 observations of different neighborhoods (horizontal axis). As can be seen, the four machine learning models generally agree well with the observed data. The model FNN (B) had the lowest estimation error with respect to the number of observed samples of the models listed. XGBoost (A), RNN (C) and LSTM (D) are underestimating the actual data with MBE values of -0.02, -0.03 and -0.02, respectively. It can be observed that these models have larger errors with more extreme values. Finally, some extreme values that have been observed in the predictions were caused by valid observations in the neighborhoods: NDSM Wharf, Bergwijkpark and Wageningen Campus. These neighborhoods have been renovated, and therefore prices increased rapidly in these places, which appears to be more difficult for the models to predict accurately.

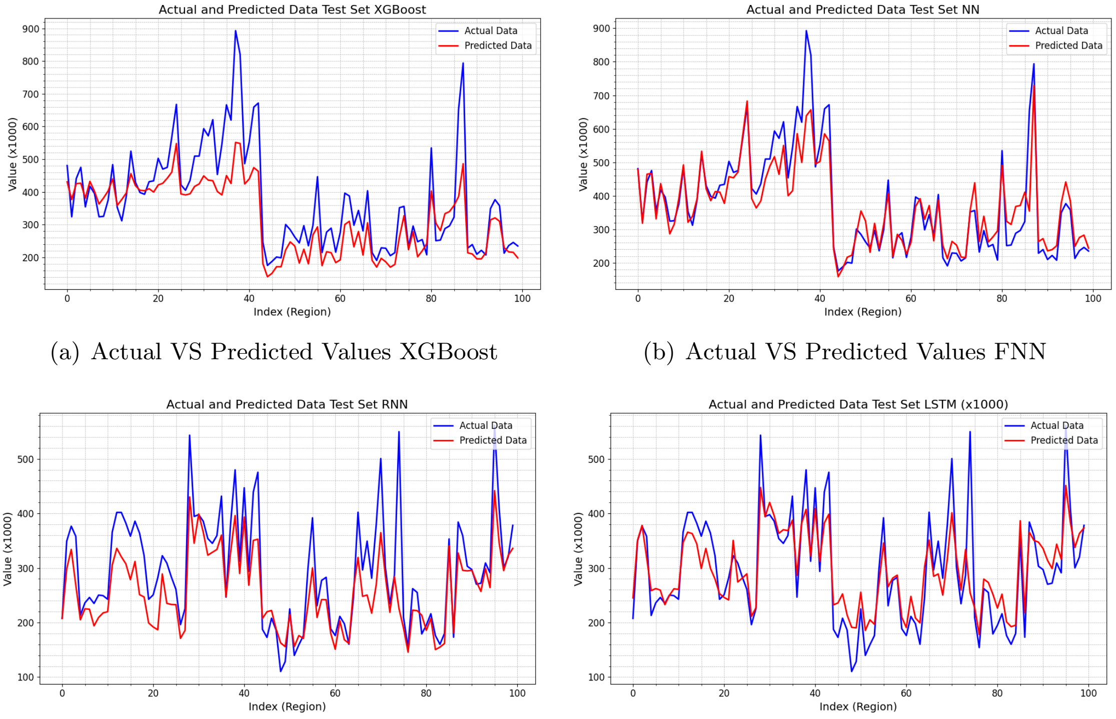
	MAE	**MSE**	**RMSE**	**MBE**	**MAPE**	**R2**
ARIMA	0.045	0.0026	0.051	-0.04	0.31	0.54
XGBoost	0.025	0.0014	0.037	-0.02	0.16	0.68
FNN	**0.024**	**0.0010**	**0.032**	**0.00**	**0.10**	**0.81**
RNN	0.031	0.0022	0.047	-0.03	0.10	0.65
LSTM	0.032	0.0023	0.048	-0.02	0.09	0.63

Table 12: Performance Metrics Test Data on the Complete Dataset

Comparison Reduced Datasets

In this section, a comparison of the reduced datasets and the complete dataset applied to the different machine learning models is provided. Each of the models XGBoost, FNN, RNN, and LSTM are employed to predict the average regional house price on the reduced datasets. Three datasets are used where macroeconomic, socioeconomic, or demographic variables are deselected from the feature space (Appendix 1.3). In this way, the effects of these variables on predicting the average regional house price is explored and compared.
In Table 13, the results of the four nonlinear regression models XGBoost, FNN, RNN and LSTM are presented for the complete dataset in comparison to the reduced models where a subset of the features are deselected from the complete dataset. The reduced datasets, set A (without socioeconomic variables), set B (without demographic variables), and set C (without macroeconomic variables), are compared against the complete dataset to reflect the importance of either of these missing variable sets in predicting the average property prices in a neighborhood.

The FNN performed the best on the reduced dataset C (without macroeconomic variables), outperforming all other models with the lowest MAE, MSE, RMSE, MAPE measures and the highest R^2 value of 0.84. Obtaining a MAE of 0.023 and a RMSE of 0.029 indicates an accurate



(a) Actual VS Predicted Values XGBoost

(b) Actual VS Predicted Values FNN

(c) Actual VS Predicted Values RNN

(d) Actual VS Predicted Values LSTM

Figure 17: The predictions of each machine learning model on the complete dataset compared against the actual values for a sample of 100 observations of the test set.

prediction with a small gap of 0.006 between the values, indicating fewer 'large' errors. The mean average precision error of 3% indicates a good fit to the model, with a minimal underestimation. However, its performance is nearly equal to the performance metric of the FNN on the complete dataset, with suptle differences that are negligible.

With respect to the complete dataset, set C obtained slightly better results for all models. Although the differences in performance are minor. This shows that macroeconomic variables do not contribute significantly to the predictions of average house prices. The set without demographic variables (set A) obtained worse results on the performance metrics than the complete data and the reduced set C. However, the results are still better than the performance metrics of the ARIMA model. XGBoost and FNN obtained the best results, with XGBoost having the smallest terms on absolute errors and deviations. FNN performs best on MAPE and R^2, suggesting better performance in terms of relative errors and general trend capture.

Low results are obtained on the reduced dataset without socioeconomic variable. The models obtained large absolute and squared errors. In addition, the MAPE is higher for each model and the R^2 value remains below 0.50. The lack of socioeconomic variables results in a decrease in performance compared to the complete dataset. Additionally, the performances of ARIMA is better than the performance of the FNN, RNN, and LSTM models.

The performance metrics are derived on normalized data. This makes it harder to interpret the accuracy on the observed house prices, as the performance metric is not in the same units as the data. Therefore, the results of the two best performing models XGBoost and FNN on dataset C are rescaled to their original scale. Rescaling the actual and predicted values leads to changes for the MAE, MSE, RMSE and MBE, which rely on the absolute differences of the residuals. MAPE and R^2 remain unchanged. In Appendix 1.4 the results are provided for the validation and test data. XGBoost obtained a MAE and RMSE of 63.84 and 92.19, respectively. The feedforward neural network achieved a MAE of 50.48 and a RMSE of 76.36. This implies that on average, the predictions are off by 50.48 (in thousands) from the actual values. Additionally, the MBE of both XGBoost and FNN are -30.86 and -56.14, respectively. This implies a systematic underestimation bias in the predictions of the respective units compared to the actual observations.

	MAE	**MSE**	**RMSE**	**MBE**	**MAPE**	**R2**
Complete						
XGBoost	0.025	0.0014	0.037	-0.02	0.16	0.68
FNN	**0.024**	**0.0010**	**0.032**	**0.00**	**0.10**	**0.81**
RNN	0.031	0.0022	0.047	-0.03	0.10	0.65
LSTM	0.032	0.0023	0.048	-0.02	0.09	0.63
Set A (Without socioeconomic)						
XGBoost	**0.042**	**0.0024**	**0.049**	**-0.04**	0.31	**0.44**
FNN	0.065	0.0060	0.077	-0.09	0.41	-0.11
RNN	0.047	0.0045	0.067	-0.05	**0.28**	0.30
LSTM	0.053	0.0047	0.069	-0.06	0.32	0.26
Set B (Without demographic)						
XGBoost	**0.030**	**0.0017**	**0.041**	-0.02	0.20	0.61
FNN	0.033	0.0019	0.044	**-0.01**	**0.15**	**0.64**
RNN	0.031	0.0027	0.052	-0.02	0.22	0.56
LSTM	0.034	0.0025	0.050	0.02	0.16	0.60
Set C (Without macroeconomic)						
XGBoost	0.024	0.0012	0.035	-0.02	0.14	0.72
FNN	**0.023**	**0.0009**	**0.029**	**-0.01**	**0.10**	**0.84**
RNN	0.027	0.0020	0.045	-0.02	0.15	0.61
LSTM	0.032	0.0020	0.044	0.02	0.11	0.69

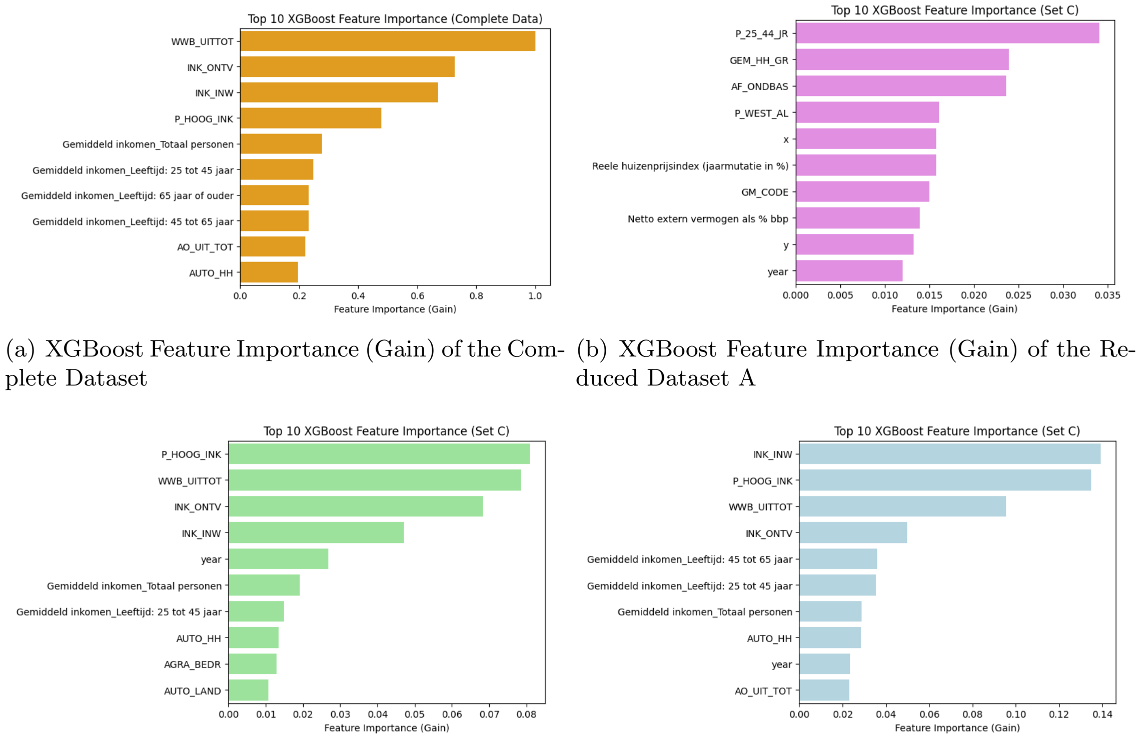
Table 13: Model Performance Comparison

5.5 Feature Importance

Feature importance provides insights in the relative importance of the individual features to reach a final output of the model. There exist three ways to measure the feature importance: Frequency, Gain and Cover (Section 4.4). For each (sub-)set of the data, the top 10 features with the highest feature importance 'gain' is presented in Figure 18. The feature importance measure 'gain' is considered most important in this research as it explains the importance of a feature in reducing the prediction error, referring to its informative ability. The feature importance measure 'Weight' is useful for understanding how frequently a feature is used to make splits in the trees. However, it does not provide information about the effectiveness of those splits in terms of improving model performance, making weight a less informative measure of feature importance compared to gain. The measure 'Coverage' indicates how many samples are affected by the splits involving the feature. It provides information on how broadly a feature is used in the model, but it doen not necessarily indicate the quality or importance of those splits. Therefore, coverage and weight are left out of scope for this research.

The socioeconomic variables have the highest feature importance results in the complete model and the reduced models including socioeconomic variables. The variable 'WWB_UITTOT', which means the number of resident in a neighborhood with a social assistance benefit, has the highest information gain. This variable which has a negative correlation with the target variable, indicating the a lower average house price in neighborhoods with higher number of social assistence benefits. This variable also appears high on the ranking with the reduced datasets B and C. The income related variables 'INK_ONTV', 'INK_INW' AND 'P_HOOG_INK', referring to the average income ('INK_ONTV', 'INK_INW') and percentage of wealthy residents in a neighborhood ('P_HOOG_INK'), are most important. In addition, the income variables related to age are contribute considerably to the models as well. The features 'Gemiddeld inkomen_Leeftijd:' are ranked in the middle of the top 10 features importance of the complete model and reduced datasets B and C.

Each of the feature depicted in feature importance figure 18 on the complete dataset are identified as socioeconomic variables. The reduced datasets B and C are mostly explained by socioeconomic variables except for the variables 'year', 'AGRA_BEDR' (number of agricultural businesses) and 'AUTO_LAND' (average automobile per km^2), which are considered as basic spatial variables.
The reduced dataset B (without socioeconomic variables) highest feature importance variables differ enormously compared to the other datasets. The highest gain is obtained by the feature 'P_25_44_JR', which is the percentage of inhabitants with ages between 25 and 44. This variable has a negative correlation of -0.29 with the target variable indicating that a larger percentage of 25-44 year old residents implies a lower average house price. Other important demographic feature in reduced dataset A are 'GEM_HH_GR' (average household growth) and 'P_WEST_AL' (percentage western origin). Furthermore, a mixture of macroeconomic such as 'Reele huizenprijsindex' (Real House Price Index) and 'Netto extern vermogen als % bbp' (Net External Assets as % of GDP) and basic spatial variables such as 'year' and geographical coordinates 'x' and 'y' contribute most to the model.



(a) XGBoost Feature Importance (Gain) of the Complete Dataset

(b) XGBoost Feature Importance (Gain) of the Reduced Dataset A

(c) XGBoost Feature Importance (Gain) of the Reduced Dataset B

(d) XGBoost Feature Importance (Gain) of the Reduced Dataset C

Figure 18: The feature importance (gain) for the extreme gradient boosting model obtained on the full dataset and the reduced datsets without socioeconomic (A), demographic (B) and macroeconomic (C) variables.

6 Discussion

In this chapter, we discuss the experimental observations and conclude how we solved our research problem that was raised in the introduction (Section 1.4). Furthermore, limitations and future research are described.

The purpose of this research was to compare advanced machine learning models with ARIMA modeling in predicting average regional house prices in neighborhoods in the Netherlands based on the socioeconomic and demographic characteristics of a region. To accomplish this, four machine learning models were exploited and validated against the performance of ARIMA. The dataset used for this experiment contained diverse attributes that described socioeconomic and demographic characteristics of a neighborhood, with a time dependence over a period from 2004 to 2022. The effects of these regional characteristics and the contributions of macroeconomic variables were also researched. The performance of the sequence models compared to non-sequential constructed models is also discussed.

6.1 Conclusion

From the results obtained in Section 5, we will interpret the findings of our study to answer the primary research question by addressing each of the subquestions posed. The answers to the subquestions will form a comprehensive narrative that leads to a conclusion that answers the research question.

What are suitable baseline models to compare RNNs to?

ARIMA is a traditional statistical method widely used in various researches for time series forecasting. The different components autoregression (AR), differencing (I), and moving average (MA) can capture the trend and seasonality in univariate time series data, making predictions by regressing on lagged data points. It represents a classical statistical technique that has been used in previous research predicting time-dependent data, while machine learning models represent modern approaches. It emphasizes the value of implementing a complex machine learning model such as a neural network (with a black-box design) in comparison to a more simplistic and interpretable statistical approach. Additionally, it emphasizes differences between predictions based solely on historical prices and the historical characteristics of a neighborhood that are (in)directly related to the average house prices. Both the ARIMA and RNN models inherently make predictions on time dependent data, which makes it an appropriate baseline model to compare to RNNs.

A hedonic pricing model would alternatively suit as a proper statistical baseline model. The employed hedonic model, Generalized Least Squares AutoRegression model (GLSAR), shows that a hedonic pricing model serves the purpose of fitting to multivariate time series data. However, the complex nonlinear nature of the data makes the interpretation of the model less reliable, as the model inherently assumes a linear relationship between the dependent variable and the independent variables.

Both models could serve as a baseline model. The ARIMA model has been chosen as the primary baseline model due to its simplistic and interpretable nature. The hedonic pricing method is presented as an alternative baseline model.

How does the baseline model perform in comparison to advanced machine learning models on the complete dataset?

In comparing ARIMA modeling with the advanced machine learning models (XGBoost, FNN, RNN, LSTM), it can be concluded that the ARIMA models achieved poor results on the test data. Overall, the ARIMA models performed worse across all performance metrics (Table 12). It resulted in relatively large errors, leading to a high MAE and RMSE of 0.045, and 0.051, respectively. Compared to the artificial neural network, which had a MAE of 0.024 and a RMSE of 0.032, this is a significant difference. Although, the out-of-sample predictions on the validation data are more comparable with the advanced machine learning models. The MAE and RMSE obtained are better than the FNN, RNN and LSTM models. It is also able to explain a high variance of the data with a R^2 measure of 0.81.
By evaluating the performance metrics, it can be concluded that the ARIMA models are underestimating the actual observations, having a negative MBE of -0.04. This can be explained by the rapid increase of house prices in the years 2021 and 2022, which are the validation and test years, respectively. The increase in prices in those year is not in relation to the general trend obtained from recent historical observations, where the impact of the financial crisis plays a big role. The upswings in the Dutch housing market last longer than the downward trends [2], leaving out an effective consistent seasonal component. This makes it harder for ARIMA to forecast the turning points. It can be concluded that the ARIMA prediction was not capable of fully capturing the increased rising trend after a period of up-and downswings.

Do sequence models perform better than non-sequence machine learning algorithms?

In comparing the non-sequence models to the RNN and the LSTM, can we conclude that the non-sequence models are better models in predicting the average regional house price on the time dependent dataset. The XGBoost and FNN models perform consistently better on the complete and diverse reduced datasets. They obtain the best results overall (Table 13). The FNNs have better performance metrics on the complete dataset and the reduced dataset C (without macroeconomic variables). The differences on MAE and RMSE are minimal, but the differences in relative error terms and variance explainability are significantly better for the feedforward neural networks, reaching a MAPE of 0.10 and a R^2 of 0.84. XGBoost obtained better results on the MAE, MSE and RMSE of the reduced models set A (without socioeconomic variables) and set B (without demographic variables). The performance on set B results in a MAPE of 0.15 and a R^2 of 0.64 for the FNN model, reaching a better 'goodness-of-fit' on the reduced data than XGBoost.

The recurrent neural network (RNN) and the long-short term memory model (LSTM) are outperformed by the other machine learning models on most performance metrics. The mean squared error, used as the loss term during training for RNN and LSTM, is nearly twice as large as the results of XGBoost and FNN. The MBE of the RNN is negative for all datasets, which means that the model consistently underestimates the actual values. When LSTM and RNN are compared, it cannot be concluded that either of the models performs better. Performance metrics on the complete dataset are slightly better for the recurrent neural network, but the differences are negligible. The performances on the reduced dataset vary slightly across the dataset with no significantly better results for either of the models. Although, it can be concluded that the RNN consistently reaches a lower MAE on all dataset.

There are several possible reasons why LSTM and RNN do not perform better than XGBoost and FNN. RNNs rely on the time dependent nature of the data. If the time dependency in the data is not strong, it cannot be captured well by the RNNs and LSTMs. Some features in the data remain constant over time, such as spatial features. This could negatively influence the time dependency in the data. Additionally, the method of providing input data into the sequence models differs from the XGBoost and FNN models, which can affect the performance. LSTMs and RNNs require a sequence of instances as input data, while XGBoost and FNN models are presented single instances. If a sequence representation does not effectively capture the underlying patterns, it affects the performance.

Are macroeconomic, socioeconomic and demographic factors contributing to the performance of the advanced machine learning models?

The contributions of the macroeconomic, socioeconomic, and demographic features are compared in relation to the complete dataset for the models XGBoost, FNN, RNN, and LSTM. Each group of features is left out of the dataset, resulting in three reduced datasets: set A (without socioeconomic variables), set B (without demographic variables), and set C (without macroeconomic variables). The results of the models on the different datasets are compared and evaluated in the results 5.4. Furthermore, the feature importance property of XGBoost is explored, providing information on the relevant gain that features exploited to reduce the residual error (Section 5.5). From these results, the following conclusions can be drawn:

1. Macroeconomic features do no contribute to the prediction of regional house prices
2. Demographic features are relevant in predicting regional house prices
3. Socioeconomic features contribute most in predicting regional house prices

The results in Table 13 show that the performance of each model on the set of variables without macroeconomic attributes has obtained better results. This illustrates that the slight differences in the performance compared to the models on the complete are likely caused by the lack of macroeconomic variables. Although the configuration of the models plays an important role in the outputs, the consistent difference between four model performances is a strong indication that the macroeconomic variables macroeconomic do not serve their purpose in the model. Furthermore, macroeconomic variables are not high-valued contributors in the feature importance regarding the complete dataset. The main reason for the lacking contributions can be given by the fact that the data is constant for all regions, thus only fluctuating over time.

Demographic characteristics are relevant in modeling the average regional house prices in the Netherlands. Leaving out demographic features increases the absolute and squared error terms for the different models. In addition, the 'goodness-of-fit' indicators R^2 and MAPE decreased compared to the performance on the entire dataset. Further analysis of the feature importance shows that the variables "Population density", "Percentage divorced", and "percentage of 25 until 44 year old residents" are relevant in predicting the average house prices. Conversely, many demographic features have a low feature importance, indicating there is an inequality in the contribution of the demographic variables relative to the performance of the model.

The most important contributors to the data are the socioeconomic variables. The reduced dataset A achieved the worst performance measures with almost twice as large errors as the models in the complete dataset. It is noticeable that each model severely underestimates the actual values with negative mean bias errors ranging from -0.04 to -0.09. This is presumably due to the lack of data related to wealth and income in regional aspects. Households with higher income levels generally indicate greater purchasing power, allowing individuals to afford more expensive homes. Leaving out the income variables and supporting variables that indicate wealth leads to overcompensation of spatial and demographic variables that are not directly related to more expensive houses. Incorporating the results from the feature importance (Section 5.5), it can be stated that the socioeconomic variables are the most relevant for extreme gradient boosting. Having the highest ten features ranked in increasing the predictive powers in the XGBoost model. Features such as social securities and income-related characteristics are the main contributors in the complete dataset, but also in the reduced datasets B and C. These variables reflect the economic status of residents in a neighborhood. neighborhoods with higher average income levels often have higher average house prices because these households can afford more expensive houses in these areas. The effect of social benefits on the average price of a house in a region can have multiple causes. On the one hand, it reflects the economic status of the neighborhood. But it can also influence social cohesion and social dynamics. Potential buyers might view the area as less desirable, negatively influencing demand and consequently reducing house prices.

6.2 Limitations

Despite a carefully constructed dataset and thoughtful experimental setup, this study is subject to several limitations that should be acknowledged. These limitations are discussed here to accurately interpret the implications and significance of the research.

A relevant trade-off that had to be made impacting the results of this study was considered when handling missing values. In the raw data collected from Centraal Bureau van de Statistiek (CBS), this research uncovered several issues with missing values. Several variables that were relevant for this research had too many missing values, which were discarded during data collection. Examples of variables that have been excluded are "Employed persons by sector", "Buy or rental property", and "House build before or after 2000". The data for these variables were incomplete because they were not collected during certain years. An example of some missing variables is provided in Figure 19 in Appendix 1.1. Not only were variables excluded from the data with missing annual values. Unfortunately, some variables had to be excluded because of repetitive missing values for some neighborhoods. Often in combination with missing values for a certain year. For example, the variable "Not actively working", which represented the number of people that are not participating in the laboring market. However, many variables are included in the dataset that contain missing values, which have been imputed using several techniques (Section 3.2). Often imputation methods assume that the data values are missing at random. However, for some variables, the data is purposely kept confidential by the institution CBS. This implies that not all variables are missing at random and could interfere with the underlying validity of the data. This problem occurs for income-related variables where 11.8% age related income variables and 15.4% of the household related income variables.

Furthermore, another limitation is the effect of short time series on prediction outcomes. The data, collected from 2004 until 2022, provides a yearly time series of 19 data points at maximum. When this is split into training, validation, and test data, only 17 (yearly) data points or less are available for training. This small number of time points could affect the performance of the time-dependent algorithms RNN, LSTM and ARIMA, which are less likely to catch the long-term time dependency of the data. It is not easy to determine the minimal number of time-dependent observations necessary for fitting these models. For ARIMA, it depends on the number of model parameters to be estimated and the amount of randomness in the data, which requires larger samples sizes for the parameters to increase. The chances are greater that ARIMA is overfitting the in-sample data with less noise and smaller sample sizes. The chosen two-step out-of-sample forecast provides a logical distribution between the in- and out-of-sample predictions. And this is also in line with the test and validation set of the sequence models. Training RNNs and LSTMs is less problematic when trained on smaller sequences, as they are capable of capturing relevant information from multiple features so that they are not only dependent on the time dependency.

In addition, a point of discussion is addressed to the minimal length of the time dimension, which is set to a minimum of ten for this research. A trade-off has been made to include as many neighborhoods as possible versus the validity of time series and sequential modeling. Recurrent deep neural networks perform better with a larger number of observations, while shorter time series reduce in performance when it fails to capture the time dependency. In contrast, XGBoost and FNN models are not disturbed by neighborhoods with short time dependencies, implying that they can use all observations in the dataset. However, this questions the validity in comparing the performances as the sequence models (LSTM and RNN) and ARIMA would then have a smaller number of observations. The results of the smallest time sequence of 10 observations is provided in the results for ARIMA, which is the most sensible model to short time series. This is to validate that the minimal time series length of ten is valid for this research by verifying that the sequences of 10 do not perform significantly worse than longer time series. The general trend of the house prices during this period has an increasing trend without a turning point, making ARIMA predictions easier. Therefore, it cannot be validated that a length of ten time points is valid in general because of this result.

Lastly, the target variable "Average regional house price" is derived by an approximation of the average house price in a neighborhood based on the average appraisal value in the neighborhood scaled by the average house price ratio of the corresponding municipality. This methodology was chosen because it is more robust against occasional sales of expensive or cheap houses in a neighborhood, which could lead to bias in the average prices of houses in a neighborhood. The underlying appraisal values ensure an average price of houses in the neighborhood, with a data-driven adjustment to the market value of the house prices. However, a drawback of this methodology is that it generalizes house prices towards the mean of a larger area of observations, reducing the individual differences in sales prices between neighborhoods. By exploiting this methodology, assumptions cannot be made on characteristics of houses reflected by the target variable. This research only considers regional characteristics and national economical statistics.

6.3 Future Research

This section aims to identify potential directions for future research that can further explore and expand upon the findings of this study. The results show the potential of machine learning models in the context of predicting house prices, even though the results showed that the these models do not translate to a direct applicable use in the real world. Despite the comprehensive analysis conducted, this study encountered certain limitations that present opportunities for further research.

One such limitation this research encountered was the limitation in availability of data. Expanding the feature space with more relevant features in the related field of macroeconomic, socioeconomic or demographic features could lead to a improved accuracy of the models. For example, the socioeconomic variables could be expanded with educational level, employed or not or the social class. Also, data about the number of houses sold in a region or more details about the type of houses could improve the models performances. This will make it easier to predict fast emerging neighborhoods such as the NDSM-Wharf. Additionally, increasing the data with more time points would increase the performance of the time dependent models. This would also imply more observations which is beneficial for deep neural networks.

Another potential direction of further research is to obtain a deeper understanding of the impact of the features on the neural networks. This research focuses on the feature importance of the extreme gradient boosting model. Expanding the scope of explainable artificial intelligence toward the deep learning models could be performed using several techniques. For example by obtaining Shapley values or exploiting neural networks with integrated gradients.

Bibliography

[1] P. Boelhouwer, "The housing market in the netherlands as a driver for social inequalities: Proposals for reform," *International Journal of Housing Policy*, vol. 20, no. 3, pp. 447–456, 2020. DOI: 10.1080/19491247.2019.1663056.

[2] A. Deelen, K. Van der Wiel, J. Olsen, R. Van der Drift, L. Zhang, and B. Vogt, *Beweging op de woningmarkt: prijzen en volumes* (CPB Notitie). Centraal Planbureau, 2020. [Online]. Available: https://www.cpb.nl/sites/default/files/omnidownload/CPB-Notitie-mrt2020-Beweging-op-de-woningmarkt-prijzen-en-volumes.pdf.

[3] CBS, *Bestaande koopwoningen, gemiddelde verkoopprijzen*, https://www.cbs.nl/nl-nl/cijfers/detail/83625NED, Accessed: December 29, 2023, 2023.

[4] M. Piazzesi and M. Schneider, "Chapter 19 - housing and macroeconomics," in ser. Handbook of Macroeconomics, J. B. Taylor and H. Uhlig, Eds., vol. 2, Elsevier, 2016, pp. 1547–1640. DOI: https://doi.org/10.1016/bs.hesmac.2016.06.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574004816300167.

[5] C. K. Wing and T. Chin, "A critical review of literature on the hedonic price model," *International Journal for Housing Science and Its Applications*, vol. 27, pp. 145–165, Jun. 2003.

[6] CBS, *Kerncijfers wijken en buurten*, https://www.cbs.nl/nl-nl/reeksen/publicatie/kerncijfers-wijken-en-buurten, Accessed: December 29, 2023, 2023.

[7] N. Li and R. Li, "A bibliometric analysis of six decades of academic research on housing prices," *International Journal of Housing Markets and Analysis*, vol. 17, no. 2, pp. 307–328, 2024. DOI: 10.1108/IJHMA-05-2022-0080. [Online]. Available: https://doi.org/10.1108/IJHMA-05-2022-0080.

[8] M. Geerts, S. vanden Broucke, and J. De Weerdt, "A survey of methods and input data types for house price prediction," *ISPRS International Journal of Geo-Information*, vol. 12, no. 5, 2023, ISSN: 2220-9964. DOI: 10.3390/ijgi12050200. [Online]. Available: https://www.mdpi.com/2220-9964/12/5/200.

[9] J. Verbruggen, H. Kranendonk, M. Van Leuvensteijn, and M. Toet, *Welke factoren bepalen de ontwikkeling van de huizenprijs in Nederland?* (CPB document). Centraal Planbureau, 2005, ISBN: 9789058332110. [Online]. Available: https://books.google.nl/books?id=QRjTR2GYH5UC.

[10] F. Fuders, "The effect of interest on the money supply, demand and growth," in *How to Fulfil the UN Sustainability Goals: Rethinking the Role and Concept of Money in the Light of Sustainability*. Cham: Springer International Publishing, 2023, pp. 59–96, ISBN: 978-3-031-37768-6. DOI: 10.1007/978-3-031-37768-6_5. [Online]. Available: https://doi.org/10.1007/978-3-031-37768-6_5.

[11] X. Xu and Y. Zhang, "House price forecasting with neural networks," *Intelligent Systems with Applications*, vol. 12, p. 200 052, 2021.

[12] S. Machin and S. Gibbons, “Valuing school quality, better transport, and lower crime: Evidence from house prices,” *Oxford Review of Economic Policy*, vol. 24, pp. 99–119, Feb. 2008. DOI: `10.1093/oxrep/grn008`.

[13] V. Been, I. G. Ellen, M. Gedal, E. Glaeser, and B. J. McCabe, “Preserving history or hindering growth? the heterogeneous effects of historic districts on local housing markets in new york city,” National Bureau of Economic Research, Working Paper 20446, Sep. 2014. DOI: `10.3386/w20446`. [Online]. Available: `http://www.nber.org/papers/w20446`.

[14] J. Shin, G. Newman, and Y. Park, “Urban versus rural disparities in amenity proximity and housing price: The case of integrated urban–rural city, sejong, south korea,” *Journal of Housing and the Built Environment*, vol. 39, Jan. 2024. DOI: `10.1007/s10901-023-10098-y`.

[15] G. Garrod and K. Willis, “Valuing goods’ characteristics: An application of the hedonic price method to environmental attributes,” *Journal of Environmental Management*, vol. 34, no. 1, pp. 59–76, 1992, ISSN: 0301-4797. DOI: `https://doi.org/10.1016/S0301-4797(05)80110-0`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0301479705801100`.

[16] J. M. Clapp and C. Giaccotto, “Residential hedonic models: A rational expectations approach to age effects,” *Journal of Urban Economics*, vol. 44, no. 3, pp. 415–437, 1998, ISSN: 0094-1190. DOI: `https://doi.org/10.1006/juec.1997.2076`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0094119097920763`.

[17] C. Xue, Y. Ju, S. Li, Q. Zhou, and Q. Liu, “Research on accurate house price analysis by using gis technology and transport accessibility: A case study of xi’an, china,” *Symmetry*, vol. 12, no. 8, p. 1329, 2020.

[18] K. Gevorgyan, “Do demographic changes affect house prices?” *Journal of Demographic Economics*, vol. 85, no. 4, pp. 305–320, 2019. DOI: `10.1017/dem.2019.9`.

[19] M. Francke and M. Korevaar, “Baby booms and asset booms: Demographic change and the housing market,” *SSRN*, 2022. DOI: `10.2139/ssrn.3368036`. [Online]. Available: `https://ssrn.com/abstract=3368036`.

[20] Y. Wang and T. Kinugasa, “The relationship between demographic change and house price: Chinese evidence,” *International Journal of Economic Policy Studies*, vol. 16, no. 1, pp. 43–65, 2022. [Online]. Available: `https://EconPapers.repec.org/RePEc:spr:ijoeps:v:16:y:2022:i:1:d:10.1007_s42495-021-00068-z`.

[21] J. Yang, “Factors influencing housing consumption of urban residents in china,” in *Proceedings of the 2022 2nd International Conference on Business Administration and Data Science (BADS 2022)*, Atlantis Press, 2022, pp. 1065–1070, ISBN: 978-94-6463-102-9. DOI: `10.2991/978-94-6463-102-9_111`. [Online]. Available: `https://doi.org/10.2991/978-94-6463-102-9_111`.

[22] G. M. Izón, M. S. Hand, D. W. Mccollum, J. A. Thacher, and R. P. Berrens, “Proximity to natural amenities: A seemingly unrelated hedonic regression model with spatial durbin and spatial error processes,” *Growth and change*, vol. 47, no. 4, pp. 461–480, 2016.

[23] J. R. Rico-Juan and P. Taltavull de La Paz, “Machine learning with explainability or spatial hedonics tools? an analysis of the asking prices in the housing market in alicante, spain,” *Expert Systems with Applications*, vol. 171, p. 114 590, 2021, ISSN: 0957-4174. DOI: `https://doi.org/10.1016/j.eswa.2021.114590`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0957417421000312`.

[24] K. Ketkar, “Hazardous waste sites and property values in the state of new jersey,” *Applied Economics*, vol. 24, pp. 647–659, 1992.

[25] B. Wei and F. Zhao, “Racial disparities in mortgage lending: New evidence based on processing time,” *Federal Reserve Bank of Atlanta Working Paper*, vol. 2022, no. 1, 2022. [Online]. Available: %5E1%5E.

[26] H. W. Richardson, J. Vipond, and R. A. Furbey, “Determinants of urban house prices,” *Urban Studies*, vol. 11, pp. 189–199, 1974. [Online]. Available: %5E1%5E.

[27] B. Öztürk, D. van Dijk, F. van Hoenselaar, and S. Burgers, “The relation between supply constraints and house price dynamics in the netherlands,” 2018.

[28] R. Bhageloe-Datadin, *De huidige crisis vergeleken met die in de jaren tachtig*. Centraal Bureau voor de Statistiek, 2013.

[29] A. Elbourne, B. Soederhuizen, and R. Teulings, *De effecten van macroprudentieel beleid op de woningmarkt* (CPB Notitie). Centraal Planbureau, 2020. [Online]. Available: https://www.cpb.nl/sites/default/files/omnidownload/CPB-Notitie-apr2020-De-effecten-van-macroprudentieel-beleid-op-de-woningmarkt.pdf.

[30] J. Rouwendal and S. Kransberg, “De impact van covid-19 op de huizenprijzen in maastricht,” English, *Real Estate Research Quarterly*, vol. 20, no. 4, pp. 1–11, Dec. 2021, ISSN: 1877-9700.

[31] N. Balemi, R. Füss, and A. Weigand, “Covid-19’s impact on real estate markets: Review and outlook,” *Finance Markets and Portfolio Management*, vol. 35, pp. 495–513, 2021. DOI: 10.1007/s11408-021-00384-6.

[32] K. J. Lancaster, “A new approach to consumer theory,” *Journal of Political Economy*, vol. 74, no. 2, pp. 132–157, 1966, ISSN: 00223808, 1537534X. [Online]. Available: http://www.jstor.org/stable/1828835 (visited on 03/08/2024).

[33] S. Rosen, “A theory of life earnings,” *Journal of Political Economy*, vol. 84, no. 4, S45–S67, 1976, ISSN: 00223808, 1537534X. [Online]. Available: http://www.jstor.org/stable/1831102 (visited on 03/08/2024).

[34] M. J. Ball, “Recent empirical work on the determinants of relative house prices,” *Urban Studies*, vol. 10, no. 2, pp. 213–233, 1973, ISSN: 00420980, 1360063X. [Online]. Available: http://www.jstor.org/stable/43080758 (visited on 03/13/2024).

[35] K. W. Chau, F. F. Ng, and E. C. T. Hung, “Developer’s good will as significant influence on apartment unit prices,” English, *The Appraisal Journal*, vol. 69, no. 1, pp. 26–30, Jan. 2001, Copyright - Copyright Appraisal Institute Jan 2001; CODEN - APPJA5. [Online]. Available: http://vu-nl.idm.oclc.org/login?url=https://www.proquest.com/scholarly-journals/developers-good-will-as-significant-influence-on/docview/199962416/se-2.

[36] A. M. Freeman, “Hedonic prices, property values and measuring environmental benefits: A survey of the issues,” *The Scandinavian Journal of Economics*, vol. 81, no. 2, pp. 154–173, 1979, ISSN: 03470520, 14679442. [Online]. Available: http://www.jstor.org/stable/3439957 (visited on 03/13/2024).

[37] R. G. Ridker and J. A. Henning, “The determinants of residential property values with special reference to air pollution,” *The Review of Economics and Statistics*, vol. 49, no. 2, pp. 246–257, 1967, ISSN: 00346535, 15309142. [Online]. Available: http://www.jstor.org/stable/1928231 (visited on 03/13/2024).

[38] M. Yazdani, *Machine learning, deep learning, and hedonic methods for real estate price prediction*, 2021. arXiv: 2110.07151 [econ.EM].

[39] M. K. Francke and G. A. Vos, “The hierarchical trend model for property valuation and local price indices,” *The Journal of Real Estate Finance and Economics*, vol. 28, pp. 179–208, 2004.

[40] E. Hannan, "The identification and parameterization of armax and state space forms," *Econometrica: Journal of the Econometric Society*, pp. 713–723, 1976.

[41] M. Kaboudan and A. Sarkar, "Forecasting prices of single family homes using gis-defined neighborhoods," *Journal of Geographical Systems*, vol. 10, pp. 23–45, 2008.

[42] H. Crosby, T. Damoulas, A. Caton, P. Davis, J. P. de Albuquerque, and S. A. Jarvis, "Road distance and travel time for an improved house price kriging predictor," *Geo-spatial Information Science*, vol. 21, no. 3, pp. 185–194, 2018. DOI: 10.1080/10095020.2018.1503775. eprint: https://doi.org/10.1080/10095020.2018.1503775. [Online]. Available: https://doi.org/10.1080/10095020.2018.1503775.

[43] J. Kim, Y. Lee, M.-H. Lee, and S.-Y. Hong, "A comparative study of machine learning and spatial interpolation methods for predicting house prices," *Sustainability*, vol. 14, no. 15, 2022, ISSN: 2071-1050. DOI: 10.3390/su14159056. [Online]. Available: https://www.mdpi.com/2071-1050/14/15/9056.

[44] L. Osland, "An application of spatial econometrics in relation to hedonic house price modeling," *Journal of Real Estate Research*, vol. 32, pp. 289–320, Jul. 2010. DOI: 10.1080/10835547.2010.12091282.

[45] W. McCluskey, M. McCord, P. Davis, M. Haran, and D. McIlhatton, "Prediction accuracy in mass appraisal: a comparison of modern approaches," *Journal of Property Research*, vol. 30, no. 4, pp. 239–265, Dec. 2013. DOI: 10.1080/09599916.2013.781.

[46] H. Selim, "Determinants of house prices in turkey: Hedonic regression versus artificial neural network," *Expert systems with Applications*, vol. 36, no. 2, pp. 2843–2852, 2009.

[47] J. J. Wang, S. G. Hu, X. T. Zhan, *et al.*, "Predicting house price with a memristor-based artificial neural network," *IEEE Access*, vol. 6, pp. 16 523–16 528, 2018. DOI: 10.1109/ACCESS.2018.2814065.

[48] H. Sharma, H. Harsora, and B. Ogunleye, "An optimal house price prediction algorithm: Xgboost," *Analytics*, vol. 3, no. 1, pp. 30–45, 2024, ISSN: 2813-2203. DOI: 10.3390/analytics3010003. [Online]. Available: https://www.mdpi.com/2813-2203/3/1/3.

[49] R.-T. Mora-Garcia, M.-F. Cespedes-Lopez, and V. R. Perez-Sanchez, "Housing price prediction using machine learning algorithms in covid-19 times," *Land*, vol. 11, no. 11, 2022, ISSN: 2073-445X. DOI: 10.3390/land11112100. [Online]. Available: https://www.mdpi.com/2073-445X/11/11/2100.

[50] X. Chen, L. Wei, and J. Xu, "House price prediction using lstm," Sep. 2017.

[51] N. Hiller and O. W. Lerbs, "Aging and urban house prices," *Regional Science and Urban Economics*, vol. 60, pp. 276–291, 2016, ISSN: 0166-0462. DOI: https://doi.org/10.1016/j.regsciurbeco.2016.07.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166046216301107.

[52] DNB, *Dnb macroeconomische stabiliteitsindicatoren*, Accessed: January 24, 2024, 2024. [Online]. Available: https://www.dnb.nl/statistieken/data-zoeken/#/details/financi-le-stabiliteitsindicatoren-jaar/dataset/46b2e851-c439-40f8-820c-c88c22a1ade9/resource/7a4faeb6-a43e-4db1-8c8f-3958ca3a4ead.

[53] CBS, *Cbs inkomen huishoudens*, Accessed: January 24, 2024, 2024. [Online]. Available: https://opendata.cbs.nl/statline/portal.html?_la=nl&_catalog=CBS&tableId=83932NED.

[54] CBS, *Cbs bevolkingsposities*, Accessed: January 23, 2024, 2024. [Online]. Available: https://opendata.cbs.nl/statline/portal.html?_la=nl&_catalog=CBS&tableId=83504NED&_theme=272.

[55] CBS, *Cbs kerngegevens nationale rekeningen*, Accessed: January 23, 2024, 2024. [Online]. Available: https://opendata.cbs.nl/statline/#/CBS/nl/dataset/84097NED/table?ts=1708523401915.

[56] CBS, *Cbs arbeidsinkomensquote*, Accessed: January 24, 2024, 2024. [Online]. Available: https://opendata.cbs.nl/statline/#/CBS/nl/dataset/84178NED/table?ts=1708523597593.

[57] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: What is it and how does it work?" *International Journal of Methods in Psychiatric Research*, vol. 20, no. 1, pp. 40–49, 2011. DOI: 10.1002/mpr.329.

[58] E. Van der Wal and W. Tammiga, *Waarom de gemiddelde koopsom geen huizenprijsindicator is.* CBS, 2008.

[59] P. Das, "Analysis of collinear data: Multicollinearity," in *Econometrics in Theory and Practice: Analysis of Cross Section, Time Series and Panel Data with Stata 15.1*. Singapore: Springer Singapore, 2019, pp. 137–151, ISBN: 978-981-32-9019-8. DOI: 10.1007/978-981-32-9019-8_5. [Online]. Available: https://doi.org/10.1007/978-981-32-9019-8_5.

[60] R. Salmerón, C. García, and J. García, *Overcoming the inconsistences of the variance inflation factor: A redefined vif and a test to detect statistical troubling multicollinearity*, 2020. arXiv: 2005.02245 [stat.ME].

[61] T. Michielsen, S. Groot, and J. Veenstra, *Het bouwproces van nieuwe woningen* (CPB document). Centraal Planbureau, 2019. [Online]. Available: https://www.cpb.nl/sites/default/files/omnidownload/cpb%20boek%20woningmarkt%20-%20boek%2033.pdf.

[62] M. A. Cruz Najera, M. Treviño-Berrones, M. Ponce, *et al.*, "Short time series forecasting: Recommended methods and techniques," *Symmetry*, vol. 14, p. 1231, Jun. 2022. DOI: 10.3390/sym14061231.

[63] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, ch. 4. [Online]. Available: http://www.deeplearningbook.org.

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

[65] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for r," *Journal of Statistical Software*, vol. 27, no. 3, pp. 1–22, 2008. DOI: 10.18637/jss.v027.i03. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v027i03.

[66] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, ACM, Aug. 2016. DOI: 10.1145/2939672.2939785. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785.

[67] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990, ISSN: 0364-0213. DOI: https://doi.org/10.1016/0364-0213(90)90002-E. [Online]. Available: https://www.sciencedirect.com/science/article/pii/036402139090002E.

[68] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994. DOI: 10.1109/72.279181.

[69] R. Pascanu, T. Mikolov, and Y. Bengio, *On the difficulty of training recurrent neural networks*, 2013. arXiv: 1211.5063 [cs.LG].

[70] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, Dec. 1997. DOI: `10.1162/neco.1997.9.8.1735`.

1 Appendix

1.1 Variable Description

Macroeconomic Variables

- **Netto extern vermogen als % bbp:** Net external assets as % of GDP [%]
- **Reele effectieve wisselkoers 3-jaars mutatie in %:** Real effective exchange rate 3-year change [%]
- **Reele huizenprijsindex jaarmutatie in %:** Real house price index annual change [%]
- **Kredietstroom private sector als % bbp:** Credit flow to the private sector as % of GDP [%]
- **Totale schuld financiele sector jaarmutatie in %:** Total debt of the financial sector annual change [%]
- **Jeugdwerkloosheid 3-jaars mutatie in %:** Youth unemployment 3-year change [%]
- **Overheid geconsolideerd — Overheidssaldo EMU:** Consolidated government balance according to the European Monetary Union (EMU) standards in % of GDP [%]
- **Totaal binnenlandse sectoren — Nationaal vorderingensaldo:** The total national claims balance for domestic sectors: the net financial position (assets minus liabilities) of all domestic sectors combined [x Mln Euro]
- **Totaal binnenlandse sectoren — Saldo kredieten aan private sector:** The net amount of loans and credits extended to the private sector by all domestic sectors combined in % of GDP [%]
- **Arbeidsinkomensquote:** The share of labor compensation (employees and self-employed) in the total earned income of a country [%]

Demographic Variables

- **AANTAL_HH**: Total households [absolute]
- **AANT_INK**: Number of income recipients [absolute]
- **AANT_INW**: Number of inhabitants [absolute]
- **AANT_MAN**: Men [absolute]
- **AANT_VROUW**: Women [absolute]
- **BEV_DICHTH**: Population density [number of inhabitants per km^2]
- **GEBOO_TOT**: Total births [absolute]
- **GEM_HH_GR**: Average household size [absolute]
- **P_GEBOO**: Birth rate [per 1,000 inhabitants]
- **P_STERFT**: Mortality rate [per 1,000 inhabitants]
- **STERFT_TOT**: Total deaths [absolute]
- **P_00_14_JR**: Persons 0 to 15 years [%]

- **P_15_24_JR**: Persons 15 to 25 years [%]
- **P_25_44_JR**: Persons 25 to 45 years [%]
- **P_45_64_JR**: Persons 45 to 65 years [%]
- **P_65_EO_JR**: Persons 65 years and older [%]
- **P_ONGEHUWD**: Unmarried [%]
- **P_GEHUWD**: Married [%]
- **P_GESCHEID**: Divorced [%]
- **P_VERWEDUW**: Widowed [%]
- **P_WEST_AL**: Western total [%]
- **P_N_W_AL**: Non-western total [%]
- **P_EENP_HH**: Single-person households [%]
- **P_HH_M_K**: Households with children [%]
- **P_HH_Z_K**: Households without children [%]
- **Alleenstaande**: Number of single persons in private households [absolute]
- **Ouder in eenouderhuishouden**: Number of households with: Person in a private household who has a parent-child relationship with one or more resident children and who has no partner in the same household [absolute]
- **Overig lid huishouden**: Number of households where holds: Person who is part of a private household other than as a partner, single-parent, or resident child [absolute]
- **Partner in paar met kind**: Number of households with: Person who forms a couple with another person in a private household with resident children [absolute]
- **Partner in paar, geen kinderen**: Number of households with: Person who forms a couple with another person in a private household without resident children [absolute]
- **Persoon in institutioneel huishouden**: Number of households with: One or more persons who occupy a living space and are provided with daily necessities on a commercial basis. Housing is also provided commercially [absolute]
- **Thuiswonend kind**: Number of households with: Person regardless of age or marital status who has a child-parent relationship with one or two parents belonging to the household [absolute]
- **Totaal personen in huishoudens**: Total number of persons in households [absolute]

Socioeconomic Variables

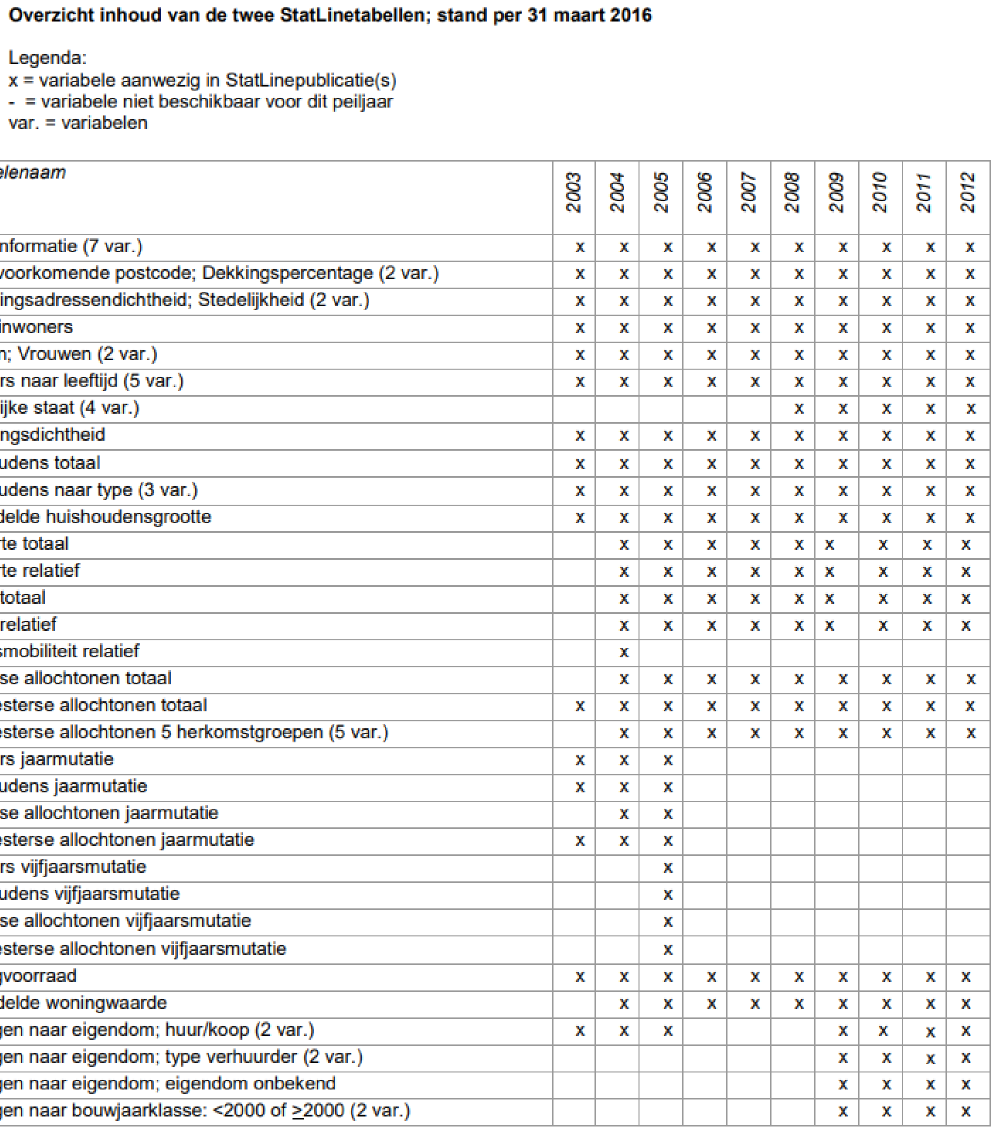
- **INK_INW**: Average income per inhabitant [x 1,000 euros]
- **INK_ONTV**: Average income per income recipient [x 1,000 euros]
- **P_HOOG_INK**: High incomes [%]
- **P_LAAG_INK**: Low incomes [%]

- **AUTO_HH**: Passenger cars per household [per household]
- **AO_UIT_TOT**: Total disability benefits [absolute]
- **WWB_UITTOT**: Total general social assistance benefits [absolute]
- **Gemiddeld inkomen_Leeftijd: 15 tot 25 jaar**: Average income per age: 15 to 25 years [absolute]
- **Gemiddeld inkomen_Leeftijd: 25 tot 45 jaar**: Average income per age: 25 to 45 years [absolute]
- **Gemiddeld inkomen_Leeftijd: 45 tot 65 jaar**: Average income per age: 45 to 65 years [absolute]
- **Gemiddeld inkomen_Leeftijd: 65 jaar of ouder**: Average income per age: 65 years or older [absolute]
- **Gemiddeld inkomen_Totaal personen**: Average income total persons [absolute]
- **Mediaan inkomen_Leeftijd: 15 tot 25 jaar**: Median income per age: 15 to 25 years [absolute]
- **Mediaan inkomen_Leeftijd: 25 tot 45 jaar**: Median income per age: 25 to 45 years [absolute]
- **Mediaan inkomen_Leeftijd: 45 tot 65 jaar**: Median income per age: 45 to 65 years [absolute]
- **Mediaan inkomen_Leeftijd: 65 jaar of ouder**: Median income per age: 65 years or older [absolute]
- **Mediaan inkomen_Totaal personen**: Median income total persons [absolute]

Basic Variables

- **DEK_PERC**: Percentage most common postal code [scale]
- **GM_CODE**: Numerical designation of municipalities [code]
- **GWB_CODE**: Numerical designation of municipality, district or neighborhood [code]
- **STED**: Urbanization based on the environmental address density [scale]
- **year**: year [absolute]
- **OPP_LAND**: Area covered by land [absolute]
- **OPP_TOT**: Area in total [absolute]
- **OPP_WATER**: Area covered by water [absolute]
- **x**: X-coordinate [absolute]
- **y**: Y-coordinate [absolute]
- **AF_ARTSPR**: The average distance of all residents in an area to the nearest general practitioner's office, calculated over the road [km]

- **AF_KDV**: The average distance of all residents in an area to the nearest daycare center, calculated over the road [km]
- **AF_ONDBAS**: The average distance of all residents in an area to the nearest primary school, calculated over the road [km]
- **AGRA_BEDR**: Number of agricultural businesses [absolute]
- **BEDR_TOT**: The number of business establishments [absolute]
- **WONINGEN**: Number of buildings intended for habitation [absolute]
- **AUTO_LAND**: Cars by area [per km^2]
- **AUTO_TOT**: Cars in total [absolute]



Overzicht inhoud van de twee StatLinetabellen; stand per 31 maart 2016

Legenda:
x = variabele aanwezig in StatLinepublicatie(s)
- = variabele niet beschikbaar voor dit peiljaar
var. = variabelen

Variabelenaam	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
Regio-informatie (7 var.)	x	x	x	x	x	x	x	x	x	x
Meest voorkomende postcode; Dekkingspercentage (2 var.)	x	x	x	x	x	x	x	x	x	x
Omgevingsadressendichtheid; Stedelijkheid (2 var.)	x	x	x	x	x	x	x	x	x	x
Aantal inwoners	x	x	x	x	x	x	x	x	x	x
Mannen; Vrouwen (2 var.)	x	x	x	x	x	x	x	x	x	x
Inwoners naar leeftijd (5 var.)	x	x	x	x	x	x	x	x	x	x
Burgerlijke staat (4 var.)						x	x	x	x	x
Bevolkingsdichtheid	x	x	x	x	x	x	x	x	x	x
Huishoudens totaal	x	x	x	x	x	x	x	x	x	x
Huishoudens naar type (3 var.)	x	x	x	x	x	x	x	x	x	x
Gemiddelde huishoudensgrootte	x	x	x	x	x	x	x	x	x	x
Geboorte totaal		x	x	x	x	x	x	x	x	x
Geboorte relatief		x	x	x	x	x	x	x	x	x
Sterfte totaal		x	x	x	x	x	x	x	x	x
Sterfte relatief		x	x	x	x	x	x	x	x	x
Verhuismobiliteit relatief		x								
Westerse allochtonen totaal		x	x	x	x	x	x	x	x	x
Niet-westerse allochtonen totaal	x	x	x	x	x	x	x	x	x	x
Niet-westerse allochtonen 5 herkomstgroepen (5 var.)		x	x	x	x	x	x	x	x	x
Inwoners jaarmutatie	x	x	x							
Huishoudens jaarmutatie	x	x	x							
Westerse allochtonen jaarmutatie		x	x							
Niet-westerse allochtonen jaarmutatie	x	x	x							
Inwoners vijfjaarsmutatie			x							
Huishoudens vijfjaarsmutatie			x							
Westerse allochtonen vijfjaarsmutatie			x							
Niet-westerse allochtonen vijfjaarsmutatie			x							
Woningvoorraad	x	x	x	x	x	x	x	x	x	x
Gemiddelde woningwaarde		x	x	x	x	x	x	x	x	x
Woningen naar eigendom; huur/koop (2 var.)	x	x	x				x	x	x	x
Woningen naar eigendom; type verhuurder (2 var.)							x	x	x	x
Woningen naar eigendom; eigendom onbekend							x	x	x	x
Woningen naar bouwjaarklasse: <2000 of $\geq$2000 (2 var.)							x	x	x	x

Figure 19: An overview of a fraction of the variables in the series of "Kerncijfers Wijken en Buurten", depicting whether these variables have been included in the yearly publication of the dataset. The years are ranging from 2003 until 2012.

1.2 Missing Values

Variable	**Missing Values**	**Imputation Method**
AANT_INK	15522	Forward/Backward Fill, MICE Imputation
AF_ARTSPR	46056	Forward/Backward Fill, Mean Imputation (Fixed Region)
AF_KDV	36905	Forward/Backward Fill, Mean Imputation (Fixed Region)
AF_ONDBAS	20177	Forward/Backward Fill, Mean Imputation (Fixed Region)
AGRA_BEDR	16330	Forward/Backward Fill, Mean Imputation (Fixed Region)
AO_UIT_TOT	2071	Forward/Backward Fill, MICE Imputation
AUTO_HH	8905	Mean Imputation, Linear Interpolation, Mean Imputation (Fixed Region)
AUTO_LAND	8905	Mean Imputation (Fixed Region)
BEDR_TOT	33919	Linear Interpolation, Mean Imputation (Fixed Region)
BEV_DICHTH	7	Mean Imputation (Fixed Year)
GEM_HH_GR	147	Forward/Backward Fill
Gemiddeld inkomen_Leeftijd: 15 tot 25 jaar	24573	Linear Interpolation, Regional Mean Imputation, MICE Imputation
Gemiddeld inkomen_Leeftijd: 25 tot 45 jaar	21443	Linear Interpolation, Regional Mean Imputation, MICE Imputation
Gemiddeld inkomen_Leeftijd: 45 tot 65 jaar	21460	Linear Interpolation, Regional Mean Imputation, MICE Imputation
Gemiddeld inkomen_Leeftijd: 65 jaar of ouder	23840	Linear Interpolation, Regional Mean Imputation, MICE Imputation
Gemiddeld inkomen_Totaal personen	21026	Linear Interpolation, Regional Mean Imputation, MICE Imputation
INK_INW	29024	Linear Interpolation, Mean Imputation (Fixed Region)
INK_ONTV	31665	Regional Mean Imputation, MICE Imputation
Mediaan inkomen_Leeftijd: 15 tot 25 jaar	24573	Linear Interpolation, Regional Mean Imputation, MICE Imputation
Mediaan inkomen_Leeftijd: 25 tot 45 jaar	21443	Linear Interpolation, Regional Mean Imputation, MICE Imputation
Mediaan inkomen_Leeftijd: 45 tot 65 jaar	21460	Linear Interpolation, Regional Mean Imputation, MICE Imputation
Mediaan inkomen_Leeftijd: 65 jaar of ouder	23840	Linear Interpolation, Regional Mean Imputation, MICE Imputation
Mediaan inkomen_Totaal personen	21026	Linear Interpolation, Regional Mean Imputation, MICE Imputation
OAD	2	Mean Imputation (Fixed Year)
OPP_WATER	27461	Forward/Backward Fill
P_00_14_JR	480	Mean Imputation (Fixed Region)
P_15_24_JR	480	Mean Imputation (Fixed Region)

P_25_44_JR	480	Mean Imputation (Fixed Region)
P_45_64_JR	480	Mean Imputation (Fixed Region)
P_65_EO_JR	480	Mean Imputation (Fixed Region)
P_GEBOO	5486	MICE Imputation
P_GEHUWD	36437	Linear Interpolation, Regional Mean Imputation, MICE Imputation
P_GESCHEID	36437	Linear Interpolation, Mean Imputation (Fixed Region)
P_HOOG_INK	21988	Forward/Backward Fill, Linear Interpolation, Mean Imputation (Fixed Region)
P_LAAG_INK	21449	Forward/Backward Fill, Linear Interpolation, Mean Imputation (Fixed Region)
P_N_W_AL	3387	Mean Imputation (Fixed Region)
P_ONGEHUWD	36436	Linear Interpolation, Mean Imputation (Fixed Region)
P_STERFT	7141	MICE Imputation
P_VERWEDUW	36437	Linear Interpolation, Mean Imputation (Fixed Region)
P_WEST_AL	620	Mean Imputation (Fixed Region)
P_EENP_HH	195	Mean Imputation (Fixed Region)
P_HH_M_K	160	Mean Imputation (Fixed Region)
P_HH_Z_K	157	Mean Imputation (Fixed Region)
STED	2	Forward/Backward Fill
WWB_UITTOT	2245	Forward/Backward Fill, Mean Imputation, MICE Imputation

Table 14: This table presents the variables that contained missing values, the number of missing values imputed for each variable, and the respective imputation methods applied in the order they were implemented.

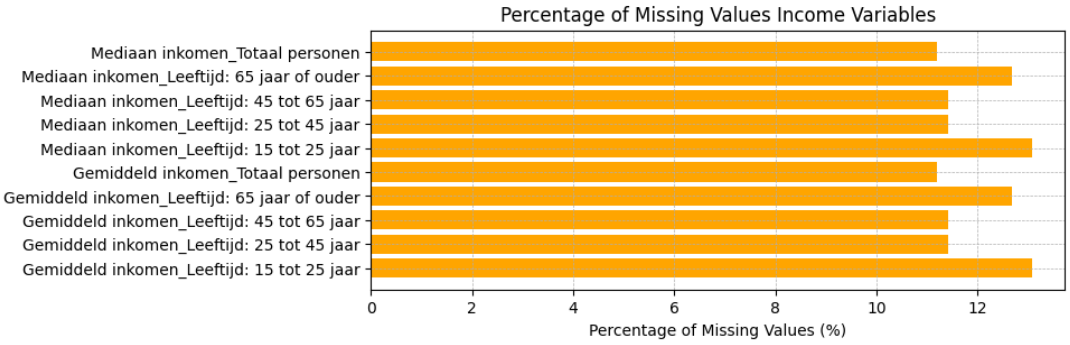


Figure 20: Percentage of missing values per variable of the additional income data (Only variables used in the final dataset).

1.3 Performance Metrics Validation Data

Metrics	MAE	MSE	RMSE	MBE	MAPE	R2
ARIMA	0.024	0.0009	0.030	-0.02	0.19	0.81
XGBoost	**0.015**	**0.0005**	**0.022**	-0.02	0.12	**0.87**
FNN	0.024	0.0010	0.032	0.01	**0.10**	0.84
RNN	0.029	0.0019	0.043	**0.00**	0.10	0.67
LSTM	0.026	0.0015	0.039	-0.01	0.08	0.74

Table 15: Performance Metrics Validation Data.

	MAE	**MSE**	**RMSE**	**MBE**	**MAPE**	R^2
Complete						
XGBoost	0.015	0.0005	0.022	-0.02	0.12	0.87
FNN	0.024	0.0010	0.032	0.01	0.04	0.84
RNN	0.029	0.0019	0.043	0.00	0.10	0.67
LSTM	0.026	0.0015	0.039	-0.01	0.08	0.74
Without socioeconomic variables						
XGBoost	0.034	0.0016	0.040	-0.03	0.27	0.57
FNN	0.056	0.0049	0.070	-0.08	0.36	0.20
RNN	0.042	0.0036	0.060	-0.06	0.28	0.39
LSTM	0.034	0.0024	0.049	-0.04	0.23	0.58
Without demographic variables						
XGBoost	0.025	0.0009	0.030	-0.02	0.20	0.76
FNN	0.034	0.0021	0.045	0.00	0.15	0.67
RNN	0.027	0.0022	0.047	-0.02	0.19	0.58
LSTM	0.026	0.0015	0.038	0.01	0.11	0.75
Without macroeconomic variables						
XGBoost	0.015	0.0004	0.020	-0.01	0.11	0.89
FNN	0.022	0.0008	0.028	-0.01	0.11	0.87
RNN	0.023	0.0014	0.037	-0.01	0.15	0.69
LSTM	0.032	0.0018	0.043	0.01	0.12	0.69

Table 16: Performance metrics on validation data of the models trained on the complete dataset

1.4 Performance Metrics Rescaled

	MAE	MSE	RMSE	MBE	MAPE	R2
Validation						
FNN	41.9086	4632.394	68.0617	-18.7305	0.11	0.87
XGB	39.1775	2838.464	53.2772	-32.9931	0.11	0.89
Test						
FNN	50.4804	5830.569	76.3582	-30.8614	0.10	0.84
XGB	63.8430	8498.802	92.1889	-56.1344	0.14	0.72

Table 17: Performance metrics in rescaled units for FNN and XGBoost models on validation and test sets of the best performing models on the reduced dataset C (without macroeconomic variables).

1.5 Performance Metrics Arima Minimal Time Series

	MAE	MSE	RMSE	MBE	MAPE	R^2
Validation	0.0244	0.00079	0.0281	-0.02	0.23	0.89
Test	0.0467	0.00254	0.0504	-0.05	0.33	0.76

Table 18: Performance metrics for ARIMA on the shortest time series of 10 time points for 195 neighborhoods on validation and test sets

1.6 Feature Importance

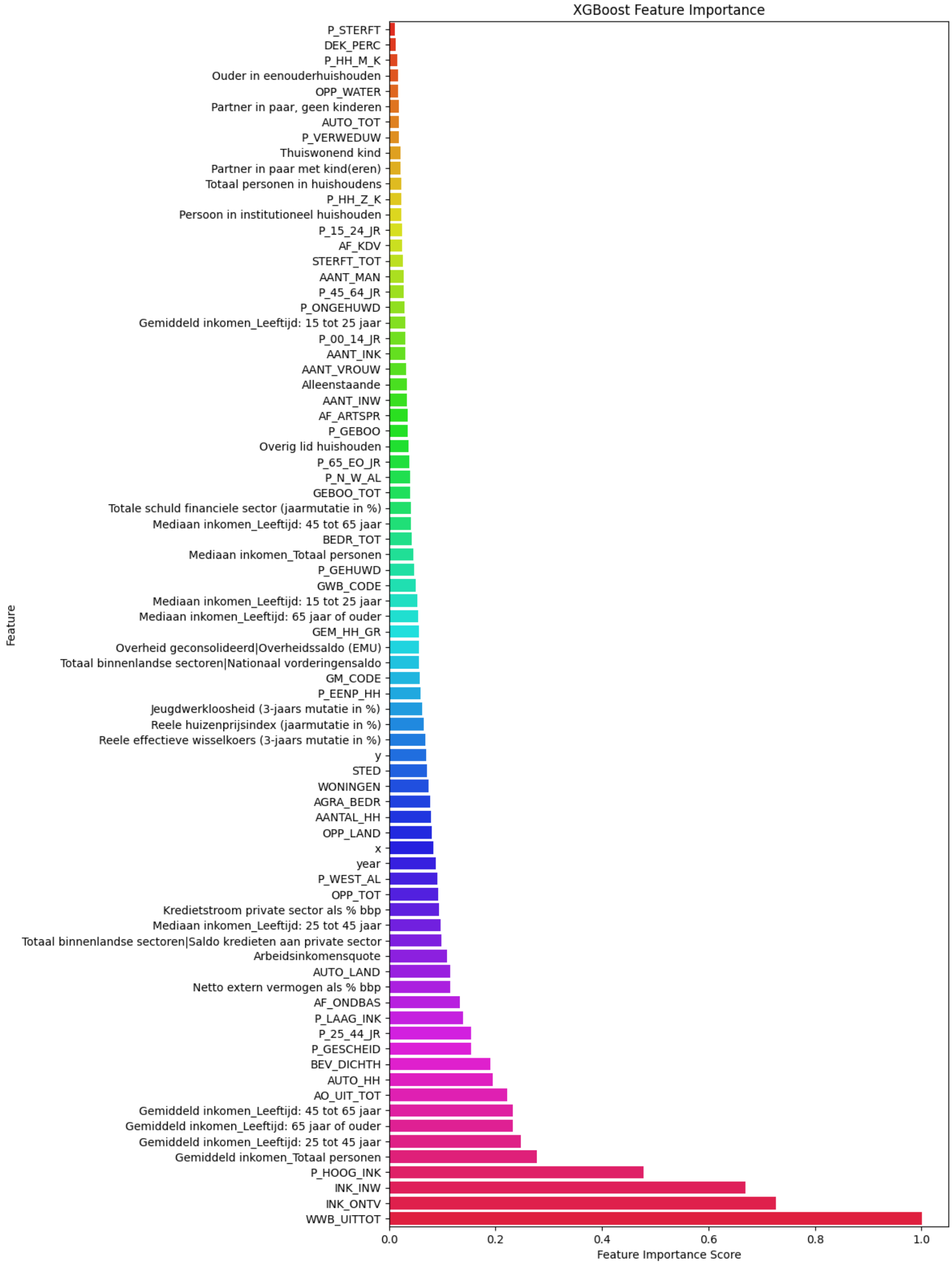
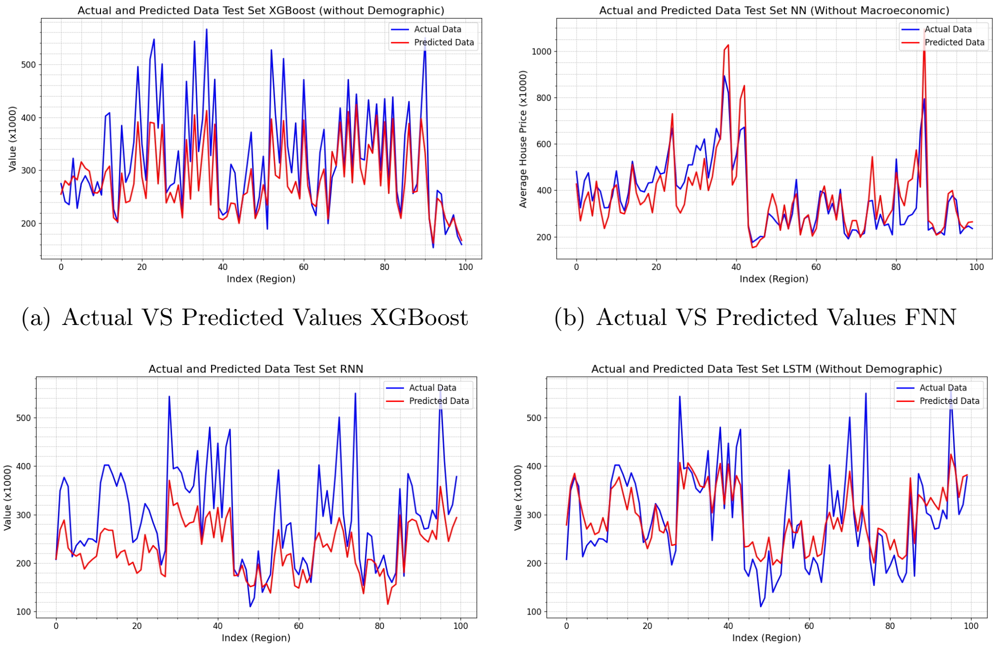


Figure 21: XGBoost feature importance (Gain) for each variable in the complete dataset.

1.7 Actual versus Predicted Plots



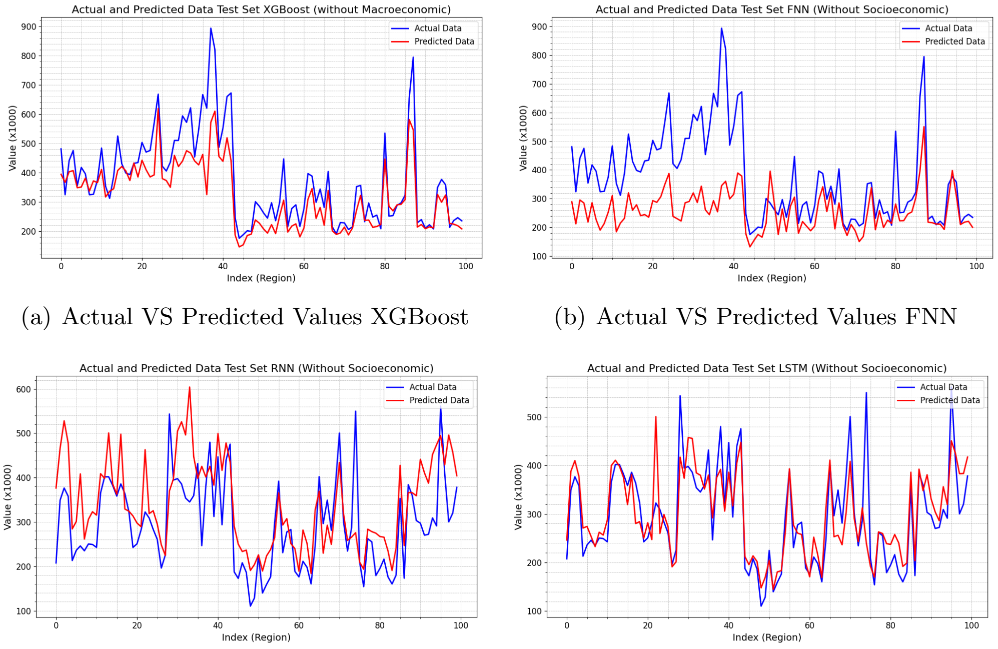
(a) Actual VS Predicted Values XGBoost

(b) Actual VS Predicted Values FNN

(c) Actual VS Predicted Values RNN

(d) Actual VS Predicted Values LSTM

Figure 22: The predictions of each machine learning model on the reduced dataset without demographic variables compared against the actual values for a sample of 100 observations of the test set.



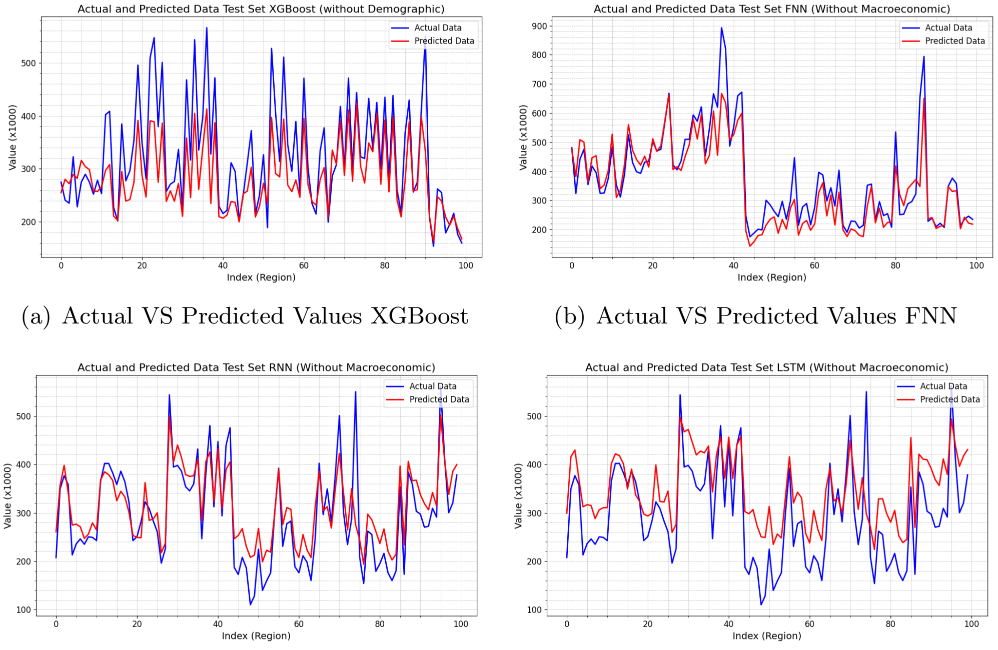
(a) Actual VS Predicted Values XGBoost

(b) Actual VS Predicted Values FNN

(c) Actual VS Predicted Values RNN

(d) Actual VS Predicted Values LSTM

Figure 23: The predictions of each machine learning model on the reduced dataset without socioeconomic variables compared against the actual values for a sample of 100 observations of the test set.



(a) Actual VS Predicted Values XGBoost

(b) Actual VS Predicted Values FNN

(c) Actual VS Predicted Values RNN

(d) Actual VS Predicted Values LSTM

Figure 24: The predictions of each machine learning model on the reduced dataset without macroeconomic variables compared against the actual values for a sample of 100 observations of the test set.

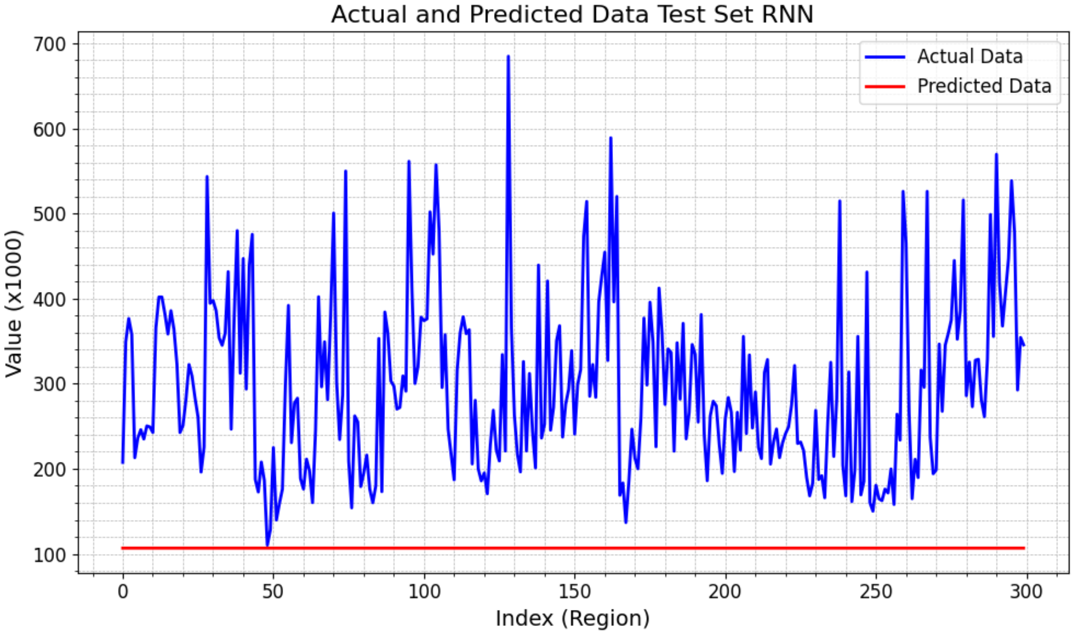


Figure 25: The predicted versus actual values during training of a 6-layer recurrent neural network with vanishing gradients on the validation set.