Spatio-Temporal Deep Learning Approaches for Hedonic Price Models and Relation Discovery

A Study of Real Estate Appraisal in Amsterdam

SUBMITTED IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTER OF SCIENCE

David H. Chen
13642111

MASTER INFORMATION STUDIES
Data Science Track
FACULTY OF SCIENCE
UNIVERSITY OF AMSTERDAM

	Internal Supervisor	External Supervisor	Examiner
Title, Name	Prof. Marc K. Francke	Dr. Alonso M. Acuña	Dr. Fernando P. Dos Santos
Affiliation	FEB, UvA	Municipality of Amsterdam	FNWI, UvA
Email	m.k.francke@uva.nl	a.acunasanchez@amsterdam.nl	f.p.santos@uva.nl

Gemeente
Amsterdam

Spatio-Temporal Deep Learning Approaches for Hedonic Price Models and Relation Discovery

A Study of Real Estate Appraisal in Amsterdam

Author: David H. Chen
david.chen2@student.uva.nl
University of Amsterdam
the Netherlands

ABSTRACT

The Hedonic method has been widely adopted for real estate appraisal and has been enhanced with machine learning and deep learning algorithms. Meanwhile, econometrics Hedonic models with time and space variant components have also been developed to capture the temporal and spatial auto-correlation in housing price. However, these machine learning or parametric models produce only predictions in training periods. This project aims to fill the gap by introducing a novel hybrid method which utilises XG-Boost regressors and a Spatio-Temporal Neural Network (STNN) to predict out-of-training-period predictions. Experiments are conducted on a housing dataset of Amsterdam. Results show that the hybrid method with STNN produces up to 22% lower forecast error than its equivalent with Recurrent Neural Network (RNN) and up to 1.7% beneath plain XG-Boost. In addition, it successfully unveils the underlying spatial correlations amongst districts in the housing market of Amsterdam. The method could be further developed by incorporating multiple levels of spatio-temporal variation and expanding the dimensions of the STNN.

KEYWORDS

Hedonic Method, Real Estate Appraisal, Forecasting, Machine Learning, Deep Learning, Gradient Boosting, Neural Network

1 INTRODUCTION

Rapid-growing prices for real estate properties have been a major issue for most cities in the world. Amsterdam is no exception. Housing prices of the Dutch capital are estimated to have risen by 14.9% from 2020 to 2021 [14] and by 17.6% for the next year [15]. In this period, even the relatively affordable western suburbs (*Slotermeer*, *Osdorp*), north Amsterdam and *Bijlmer* in the southeast have seen marked increases while *De Gouden Bocht* remains the most prestigious area of the city [16].

As the cost of land is one of the most decisive factors in real estate prices, accurate prediction of land value plays a crucial part in the municipality's efforts to control the fast growth in housing costs. On the one hand, the administration aims to keep housing affordable to its residents. On the other hand, appraisal of land in the city should reflect the true value to achieve efficient utilization of limited resources. Currently the residual method, which valuates the land value by subtracting the estimated development cost from the total worth of all the exiting properties upon the land [13], is adopted by the Municipality of Amsterdam to appraise the value of land for redevelopment. Therefore, the problem is again transformed back to real estate appraisal.

The Hedonic pricing method [7], built upon parametric regression models, is one of the most widely used approach for valuation of real estate properties. This method is also utilised and extended in the machine learning field, with the Boston Housing Project being the most famous use case. The objective of the project is to build regression models with machine learning algorithms to predict housing prices with features of the houses as well as the neighbourhood, based on a dataset obtained in suburbs of Boston [1]. However, most of the models from the machine learning projects produce only predictions in the training period while out-of-period predictions, or forecasts, appear to be more valuable for real estate investments, mortgage calculation and land redevelopment, just as the use case of the Municipality of Amsterdam. Given such circumstances, this project aims to utilise state-of-the-art machine learning and deep learning algorithms, especially algorithms for forecasting, to provide out-of-period predictions for real estate appraisal. Furthermore, this project also explores the additional insights of spatial features brought by the deep learning model deployed.

2 RELATED WORK

With the rapid evolution of artificial intelligence technologies and exponential growth of computing power, there have been applications of machine learning and deep learning algorithms on the Hedonic method as well as forecasting tasks. In the meantime, various statistical models have also been developed to capture the spatial and temporal variation in real estate prices. This section briefly describes the state-of-the-art parametric and non-parametric approaches related to real estate appraisal and forecasting.

2.1 Machine Learning Approaches on Hedonic Method

In a study of housing prices of Onondaga County, New York, *Yoo et al.* find that Random Forest generates Hedonic price predictions of higher accuracy than *Cubist* method, a support vector regressor, and statistical regression models with ordinary least squares (OLS) method [18]. *Yazdani* has similar findings on Random Forest, but he also suggests that Artificial Neural Networks (ANN) outperforms parametric regression on property data of Boulder, Colorado [17]. On the contrary, *Mayer et al.* find that gradient boosting provides more accurate Hedonic predictions over parametric regression, Random Forest and Neural Networks [8]. Furthermore, *Ming et al.* conclude that XG-Boost produce predictions of higher accuracy than Random Forest Regressor and LightGBM on house rental data from Chengdu [9].

2.2 Parametric Spatio-Temporal Models on Real Estate Appraisal

As transaction date and geographic location are the two key factors in housing prices, various statistical regression models with spatial and temporal components have been developed by economists to capture the variation. The Hierarchical Trend Model (HTM) published by *M. Francke* is a state-space model with the assumptions that the log-transformed housing prices can be decomposed into time-variant and time-invariant components. The time-invariant part mainly consists static features of real estate properties, e.g. number of rooms and area of living, which could be modelled by traditional Hedonic method. The time-variant part includes a hierarchy of underlying trends from general to neighbourhood level, which shows characteristics of time series such as random walk or drift [5]. The parameters of HTM are optimised through Kalman filter and smoother [5, 6].

Other spatio-temporal models for real estate appraisal are mainly variants of traditional regression. By incorporating spatial and temporal auto-regression, *Pace et al.* develop a spatio-temporal regression model, producing 8% lower sum of square error (SSE) than OLS regression with spatial indicator variables on a one-step ahead forecast experiment of housing price in Baton Rouge, Louisiana [11]. *Fotheringham et al.* develop a hybrid model named GWR-TS by integrating time-series forecast methods into a geographically weighted regression (GWR) model, built with local spatial modelling techniques. The GWR-TS model achieve lower root-mean-square error (RMSE) than OLS regression on housing price forecast in London [4]. *Crosby*, *Davis*, *Damoulas* and *Jarvis* claim to have beaten machine learning algorithms such as Random Forest and M5P-regression-tree on RMSE forecasting British real estate properties with their spatio-temporal Gaussian Process Regression [2].

2.3 Spatio-Temporal Deep Learning Algorithms

Since the invention of Recurrent Neural Network (RNN) [12], numerous deep learning algorithms have been utilised for forecasting purposes. The Spatio-Temporal Neural Network (STNN) proposed by *Delasalles et al.* is a variant of RNN and also a state-space model [3]. As opposed to RNN which captures latent dynamics of a one-dimensional series, STNN is designed to learn and capture, by Stochastic Gradient Descent, the temporal variation of a multi-dimensional series with dependencies amongst neighbouring dimensions or, alternatively, multiple temporal series from a spatial scope with spatial auto-correlation amongst neighbouring series. Apart from learning with pre-defined spatial correlation, STNN is developed with two other modes, denoted by STNN-R (Refine) and STNN-D (Discovery), which are capable of refining (with prior information) and discovering (without prior information) underlying spatial correlations. The algorithm is applied on forecasting and imputation of disease spreading, weather conditions and road traffic and outperforms classical auto-regressive model and other deep learning algorithms such as RNN and vectorial auto-regressive multilayer perceptron (VAR-MLP) on RMSE [3].

The STNN algorithm has been modified for similar forecast projects. *Niu et al.* build STNN with Augmented Spatial States (STNN-A) and STNN with Input Gate (STNN-I) to forecast spreading of Covid-19 in China, USA and Italy. Both STNN-A and STNN-I outperform parametric regression models, ANN and RNN on RMSE [10]. *Zhang* and *Patras* propose a double-STNN model which reduces prediction error on 10-hour traffic forecast by up to 61% compared to common methods including Auto-regressive Integrated Moving Average (ARIMA) and Convolutional Long Short-Term Memory [19].

3 RESEARCH QUESTIONS

Based on the literature review in Section 2, we hereby identify a research gap where non-parametric spatio-temporal deep learning algorithms are yet to be applied on Hedonic price models and forecasting housing price. Therefore, we set the main research question of this project to be:

To what extent can spatio-temporal deep learning approaches improve Hedonic method on real estate appraisal?

This research question is further divided into three sub-questions:

3.1 Sub-question 1

To what extent can spatio-temporal deep learning approaches outperform Hedonic method by classic machine learning algorithms?

3.2 Sub-question 2

To what extent can spatio-temporal deep learning approaches outperform Hedonic method with machine learning algorithms for temporal forecasting?

3.3 Sub-question 3

To what extent do the spatial relations refined and discovered by spatio-temporal deep learning approaches differ from human-determined values?

4 METHODOLOGY

To investigate the performance of spatio-temporal deep learning approaches on Hedonic method and answer the research questions, the Spatio-Temporal Neural Networks (STNN) and its variants STNN-R and STNN-D developed by *Delasalles et al.* [3] are utilised as the main algorithm of this project. As the STNN is developed from Recurrent Neural Network (RNN), an RNN is also deployed as the benchmark algorithm of temporal forecasting. Due to the superb performance of Extreme Gradient Boosting (XG-Boost) on Hedonic method as described in literature, an XG-Boosted tree-based non-linear regression model is adopted as the baseline model.

4.1 Baseline Method: XG-Boosted Regression

As the baseline method, the XG-Boost regressor assumes that there is no spatial or temporal auto-correlation of real estate prices. Instead, the spatial (geographic locations) and the temporal (transaction dates) features are treated in the same way as the static features (characteristics of properties, e.g. area of living, number of rooms, etc.). This regression model is described in Equation 1:

$$y_i = f(X_i) + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2) \tag{1}$$

In Equation 1, y_i is the natural log of the transaction price per m^2 of real estate property i, X_i is the feature vector of property i, which includes all the static features as well as the spatial and temporal features. f is the non-linear function to be learnt by the XG-Boost regressor. ε_i is the error term of property i and we assume that the error follows normal distribution with zero mean.

4.2 Hybrid Method with Recurrent Neural Network

Inspired by the Hierarchical Trend Model (HTM) [5] developed by *Francke et al.*, we assume that there exists only temporal but no spatial auto-correlation in the underlying trend of transaction price. As compared to X_i in Equation 1, the temporal feature is excluded in X_i' in Equation 2. However, the spatial feature remains in X_i' along with the static features.

f' is a non-linear function similar to f in Equation 1. For the purpose of performance comparison, *f'* is learnt by an XG-Boost regressor with the same hyper-parameter settings as the baseline method.

We further assume that there exists a certain level of property price, denoted as Φ_t, for every temporal period *t*, and the log-transformed property price is the product of temporal price level Φ_t and static estimator $f'(X_i')$, plus a zero-mean normally distributed error term.

$$y_{it} = \Phi_t \cdot f'(X_i') + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2) \tag{2}$$

For a dataset consisting of real estate transactions of T temporal periods, all the price level values Φ_1, Φ_2, ..., Φ_T compose an auto-correlated sequential series, which could be modelled by Recurrent Neural Network (RNN). The RNN algorithm deployed in this project captures the dynamic process and is expressed in a latent space. The latent factor of temporal period t is denoted by Z_t in Equation 3. g is the dynamic transition function that maps the latent factor Z_t of temporal period t to the next sequential latent factor Z_{t+1}. Note that this approach is different from the traditional RNN algorithm [3], where the latent factor is also generated with the ground truth of series values, i.e. $Z_{t+1} = g(Z_t, \Phi_t)$. However, as the method in section 4.3 also learns the dynamic latent factors entirely in the latent space, the same approach is adopted for RNN for the purpose of performance comparison with the hybrid method using STNN introduced in Section 4.3.

The dynamic transition function g is learnt by RNN with back propagation and stochastic gradient descent.

$$Z_{t+1} = g(Z_t) \tag{3}$$

For every temporal period, RNN generates an estimation of the series value, denoted by $\hat{\Phi}_t$ in Equation 4, using the decoder function *d*.

$$\hat{\Phi}_t = d(Z_t) \tag{4}$$

4.3 Hybrid Method with Spatio-temporal Neural Network

On top of section 4.2, we further assume that there exists both temporal and spatial auto-correlation in the underlying trend of property price. In Equation 5, the temporal and spatial features are excluded in the feature matrix X_i''. The non-linear function *f"* is also learnt by an XG-Boost regressor with the same hyper-parameters as in sections 4.1 and 4.2.

Similar to section 4.2, we assume that there exists a certain level of property price, denoted as Ψ_{gt} for every temporal period t and spatial division *g*. The log-transformed property price is the product of spatio-temporal price level Ψ_{gt} and static estimator $f''(X_i'')$, plus a zero-mean normally distributed error term.

$$y_{igt} = \Psi_{gt} \cdot f''(X_i'') + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2) \tag{5}$$

For a dataset consisting of real estate transactions of T temporal periods and G spatial divisions, all the price level values $\Psi_{1,1}$, $\Psi_{1,2}$, ..., $\Psi_{1,T}$, $\Psi_{2,T}$, ... $\Psi_{G,T}$ compose of a $G \times T$ panel, which could be modelled by the STNN algorithms. In contrast to the scalar latent factor Z_t in section 4.2, the latent factor Z_t' in Equation 6 is a vector of length *G*, which denotes the latent state of all *G* spatial divisions at temporal period *t*.

The spatial auto-correlation is integrated with matrix *W*, which is a $G \times G$ correlation matrix of all *G* spatial divisions. When STNN is deployed, *W* stays static with pre-defined input values. STNN-R takes the pre-defined *W* as initial values and refines the non-zero cells. In the case of STNN-D, all cells of *W* are initiated with equal value 1 / ($G \times G$) and the ultimate output values are learnt in the training process.

The dynamic transition function *h* in Equation 6 generates the latent factor at *(t+1)* Z_{t+1}' from its own latent state at time at *t* Z_t' (intra-dependency) and a weighted mean of latent states of all other spatial divisions $W \cdot Z_t'$ (inter-dependency). θ_0 and θ_1 are linear mappings of any element of Z_t' to itself and all other elements, respectively.

$$Z_{t+1}' = h(Z_t' \cdot \theta_0 + W \cdot Z_t' \cdot \theta_1) \tag{6}$$

With forecast of any latent factor Z_t', estimation of the corresponding spatio-temporal price level $\hat{\Psi}_{gt}$ could be generated using the decoder *d'* and linear mapping θ_g.

$$\hat{\Psi}_{gt} = d'(Z_t') \cdot \theta_g \tag{7}$$

4.4 Iterative Method and Price Standardisation

Contrary to the Hierarchical Trend Model, which is optimised with Kalman filter and smoother [6], there exists no one-shot solution to optimised the hybrid methods for RNN and STNN. Therefore, iterative approaches, including multiple training and re-calibration of XG-Boost regressors, have to be adopted in order to separate the dynamic and static components of both hybrid methods.

Such iterative approaches are described in pseudo-codes of Algorithm 1. The pseudo-codes briefly describe the use case of T-fold cross-validation of forecast horizon h with rolling training window of length of T_0 periods. Note that arithmetic-mean is adopted in the averaging process of ratios due to better results compared to harmonic-mean and geometric-mean.

5 EXPERIMENTAL SETUP

To compare the performance of the three methods introduced in Section 4, experiments are conducted on a case of real estate appraisal of residential properties of Amsterdam. This section describes the dataset and the features, the cross-validation method for forecast

Algorithm 1 Iterative Method for Hybrid Methods

Optimise XG-Boost hyper-parameters with data in initial T_0 periods
for k in 1 to T **do**
  training window $\leftarrow [k, T_0 + k - 1]$
  period to forecast $\tau \leftarrow T_0 + h + k$
  Fit dynamic XG-Boost regressor R_d with data in training window
  min_t $\leftarrow k$ (minimum value of training window)
  if model is RNN **then**
    Calculate "static" predictions $\widetilde{y}_i$ with R_d and min_t
    Calculate ratios $r_{it} \leftarrow y_{it}/\widetilde{y}_i$
    Average ratios r_{it} by time to get levels $\hat{\Phi}_t$
    Re-calibrate static prediction $\widetilde{y}_i \leftarrow y_{it}/\hat{\Phi}_t$
    Train RNN with $\hat{\Phi}_t$
    Generates dynamic forecast $\hat{\Phi}_\tau$ of period τ
  else ▷ model is STNN/STNN-R/STNN-D
    one-hot encoded spatial vector $\mathbf{S} \leftarrow \mathbf{0}$
    Calculate "static" predictions $\widetilde{y}_i$ with R_d, min_t and $\mathbf{S}$
    Calculate ratios $r'_{igt} \leftarrow y_{igt}/\widetilde{y}_i$
    Average ratios r'_{igt} by time and space to get levels $\hat{\Psi}_{gt}$
    Re-calibrate static prediction $\widetilde{y}_i \leftarrow y_{igt}/\hat{\Psi}_{gt}$
    Train STNN/STNN-R/STNN-D with $\hat{\Psi}_{gt}$
    Generates dynamic forecast $\hat{\Psi}_{g\tau}$ of of period τ
  end if
  Fit static XG-Boost regressor R_s with re-calibrated $\widetilde{y}_i$
  Calculate static forecast $\widetilde{y}_j$ of period τ with R_s
  if model is RNN **then**
    Combined forecast values $\hat{y}_{j\tau} \leftarrow \hat{\Phi}_\tau \cdot \widetilde{y}_j$
  else ▷ model is STNN/STNN-R/STNN-D
    Combined forecast values $\hat{y}_{jg\tau} \leftarrow \hat{\Psi}_{g\tau} \cdot \widetilde{y}_j$
  end if
end for

as well as the hyper-parameter settings for the machine learning and deep learning models.

5.1 Dataset and Features

This thesis project utilises a dataset from *Watson + Holmes* [1] with **100,727** unique entries of transaction records of residential real estate properties in Amsterdam from **1 Jan. 2010** to **31 Dec. 2021**. Tables 2 and 3 of Appendix A provide summaries for numerical features with the original response variable (transaction price per m^2 before log-transformation) and categorical/binary features, respectively.

5.1.1 Static Features. The features listed in Table 1 describe the characteristics of the real estate properties and therefore considered to be static. Variance from these features are captured by an XG-Boost regressor in hybrid methods with RNN or STNN.

5.1.2 Spatial and Temporal Features. The dataset from *Watson + Holmes* provides spatial features of different levels for every transaction record. These spatial features are *Stadsdeel* (District), *Gebied* (region), *Wijk* (quarter), *Buurt* (Neighbourhood) and geographic coordinates (latitude and longitude values) from high to low level.

[1] https://www.watsonholmes.nl/

Feature	Type
Single-/Multi-family Type	Binary
Parking	Binary
Furnished	Binary
Upholstered	Binary
Building Form	Binary
Object Type	Categorical
Energy Label	Categorical
Maintenance Level	Categorical
Construction Year	Numerical
Area of Living	Numerical
Number of Rooms	Numerical

Table 1: Static Features and Types. Feature names in Dutch are originals from the dataset. Feature names in English are unofficial translations and for reference only.

The dataset also provides exact transaction date for every data entry. However, to utilise the RNN and STNN models, data series and data panels have to be constructed using discrete spatial and temporal features. Therefore, data points have to be aggregated in temporal or spatio-temporal bins to generate temporal series Φ_t and spatio-temporal panel Ψ_{gt}.

High level of aggregation (e.g. *Stadsdeel* × Quarter) results in loss of information while low level of aggregation (e.g. *Buurt* × Month) leads to fluctuation in Φ_t and Ψ_{gt} or even lack of data points in certain combination. Therefore, the level of details in the aggregation process of spatial and temporal features must be chosen with caution. Experimental results show that aggregation by *Gebied* and month generates the best and most stable forecast results. Hence, *Gebied* and transaction month are chosen as the spatial and temporal features.

Figure 1 shows the mean transaction price per m^2 by month of the whole dataset. Figure 2 shows the mean price per m^2 by *Gebied* of the whole dataset scaled in colour. The temporal and spatial auto-correlations in the underlying trend of the transaction price could be effortlessly observed.

5.2 Cross-Validation

To compare experimental results and validate models, we adopt a method similar to time-series cross-validation. As shown in Figure 3, the original dataset is split into rolling training windows and forecast testing sets based on temporal periods, skipping certain periods in between. In the experiments in Section 6 particularly, 15-fold one-month forecast cross-validations are conducted. Forecast horizons are set from 1 to 24 month(s) ahead so the rolling training windows have length 129 to 106 months, respectively.

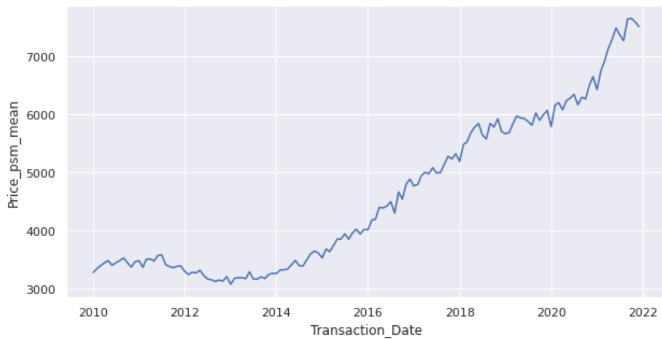

Figure 1: Temporal Feature. Mean price per m^2 by month of the whole dataset.

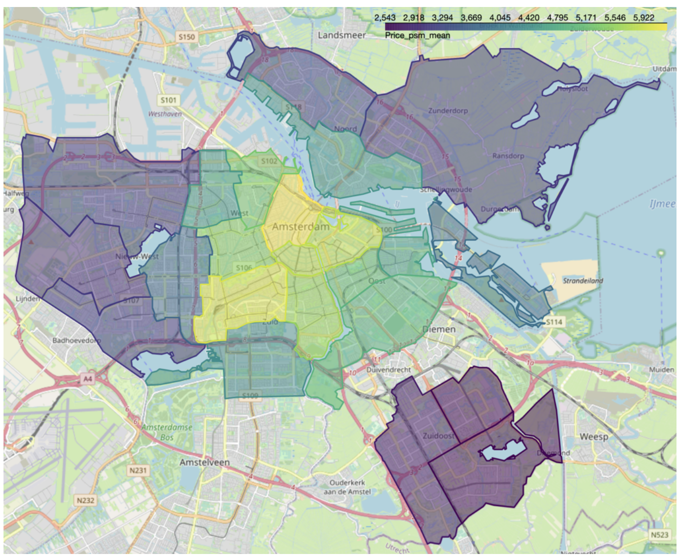

Figure 2: Spatial Feature. Mean price per m^2 by *Gebied* of the whole dataset, scaled in colour.

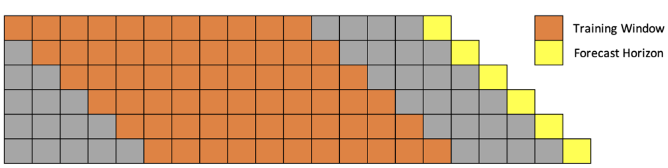

Figure 3: Cross-validation with rolling training window.

5.3 Pre-defined Spatio-Weights

As described in Equation 6 in Section 4.3, STNN and STNN-R models require a $G \times G$ matrix as input, which indicates the (initial) correlation weight of every pair of spatial divisions. In the experiments in Section 6, as *Gebied* is chosen as the spatial division, W is a 22×22 matrix for 22 *Gebieden* in Amsterdam. Based on domain knowledge, *Gebied*-pairs sharing land borders are assigned with weight 1.0, pairs bordered by small water-bodies (e.g. lakes) are assigned with weight 0.5 while pairs separated by River Amstel, which flows through the centre of the city, are assigned with weight 0.2. Meanwhile, the diagonal of W is set as zero, with the assumption of zero self auto-correlation. The pre-defined W is visualised in the left plot of Figure 9.

5.4 Hyper-parameter Settings

5.4.1 Hyper-parameters for XG-Boost. For fair comparison of model performance, the XG-Boost regressors of the baseline method as well as the two hybrid methods are all constructed with 500 gradient boosted trees. The other hyper-parameters are optimised by a 5-fold grid search using data of the initial training window of each cross-validation experiment. The selection grid is listed in Section B of Appendix.

5.4.2 Hyper-parameters for RNN and STNN. The experiments of this project directly utilise STNN/STNN-R/STNN-D models developed by *Delasalles et al.*, which are optimised by grid search [3]. The neural networks are configured with one hidden layer along with identity function or hyperbolic tangent (*tanh*) as activation functions. Note that it is not a common practice to use the linear identity function as the activation function of a neural network. However, due to the good experimental results from *Delasalles et al.*, we decide to include the linear function in our experiments.

In the meantime, we experiment with other hyper-parameters, such as multiple hidden layers, *ReLU* and *Softplus* activation functions but forecast results are not improved. As the magnitude of the sample data series provided by *Delasalles et al.* are of similar magnitude of the (spatio-)temporal series of this project, we decide to re-scale the series and adopt the configuration from *Delasalles et al.*.

6 EXPERIMENT RESULTS

With the experimental setup described in Section 5, multiple experiments are conducted on different forecast horizons ranging from 1 to 24 months. Subsequently, the experiment with the best performance in terms of forecast error and stability is selected and we further investigate model performance on every month in the cross-validation periods. In addition, we compare the spatial correlation matrix W refined and discovered by the STNN models to the pre-defined values.

6.1 RMSE Percentage of Individual Forecast Horizons

As described in Section 5.2, all experiments are cross-validated on the last 15 months of the whole temporal period of the dataset, namely Oct. 2020 to Dec. 2021. We evaluate the forecast error by the ratio (as percentage) of the RMSE of the forecast values and the mean of the ground truth on the 15-month cross-validation period. Figure 4 shows the RMSE percentage of the five methods on individual forecast horizons, with RNN/STNN/STNN-R/STNN-D having *tanh* as the activation function. Figure 5 shows results from the same experiments, except for identity activation function for the neural network models. The exact RMSE percentage values are listed with 3-month intervals in Table 4 of Appendix C.

From Figures 4 and 5, we can effortlessly learn that RNN is the worst model in terms of forecast error. STNN generates lower forecast error than RNN but is far from the performance of the other three methods. The positions of curves of STNN-R and STNN-D appear to be swapped in the two graphs. In Figure 4, STNN-R is slightly above XG-Boost while STNN-D is below the baseline curve starting from 11-month horizon. The biggest gap between STNN-D and XG-Boost is 1.12% at 21-month horizon, which also sees the biggest gap between STNN-D and RNN at 21.11%. In Figure 5, STNN-D stays above XG-Boost for all horizons with a steep increase from 21 to 24 months, while STNN-R is below the baseline curve from 10-month horizon onwards, with largest difference of 1.77% also at 21-month horizon. The largest difference between STNN-R and RNN is also recorded at 21-month horizon with 22.49 %. To summarise, the baseline method performs the best for short forecast horizons but is beaten by STNN-D with *tanh* activation and STNN-R with identity function for long horizons.

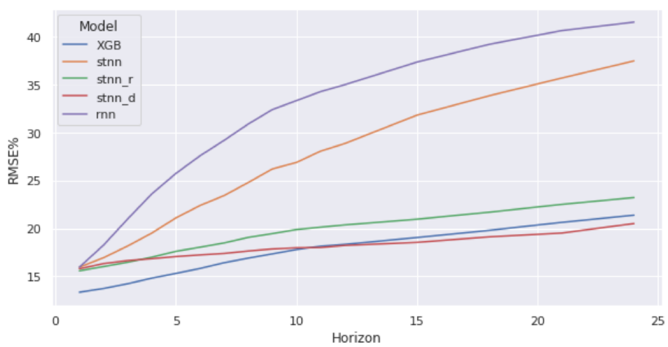

Figure 4: Results from individual models of cross-validation experiments of different forecast horizons. Values are percentage of RMSE over mean price per m^2 of every forecast horizon. RNN, STNN, STNN-D, STNN-R models are configured with *tanh* activation function.

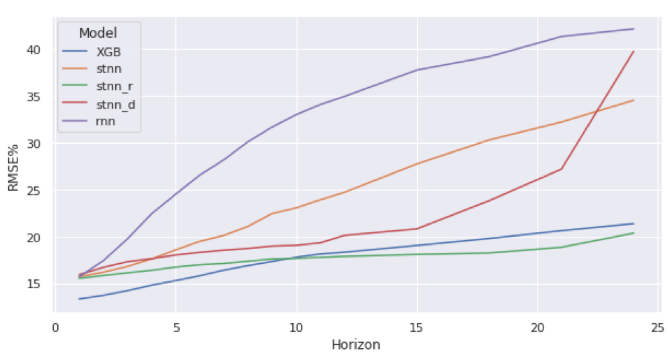

Figure 5: Results from individual models of cross-validation experiments of different forecast horizons. Values are percentage of RMSE over mean price per m^2 of every forecast horizon. RNN, STNN, STNN-D, STNN-R models are configured with identity activation function.

6.2 RMSE Percentage of Individual Forecast Periods

Although the biggest gap of RMSE percentage appears at 21-month horizon, it is not necessarily the best performance of the STNN hybrid method, as longer forecast horizon brings larger forecast errors. In this subsection, we compare RMSE percentage of the five methods by month in the 15-month cross-validation period, namely Oct. 2020 to Dec. 2021.

Figures 6, 7 and 8 show the RMSE percentage by month of the five methods of forecast horizons of 3 months, 11 months and 21 months respectively. The exact RMSE values are listed in Tables 5, 6 and 7 of Appendix D respectively. The RNN and STNN/STNN-R/STNN-D models all use *tanh* as activation function. Again, it could be easily noticed that RNN and STNN methods are outperformed in almost every month by XG-Boost, STNN-R and STNN-D methods, regardless of the forecast horizon.

In all three graphs, the RMSE percentage of XG-Boost method has a sharp increase between Feb. and Apr. 2021. In the same period, STNN-D hybrid method starts to surpass the baseline model for 11-month and 21-month horizons. Checking back on Figure 1, the period saw a boom in the housing market of Amsterdam. The results imply that STNN-D hybrid method beats the baseline model on detecting turning points of the market trends.

In Figures 7 and 8 particularly, where the STNN-D hybrid method transcends the baseline on overall RMSE percentage, the RMSE percentage remains below 20% after the steep growth of price around Mar. 2021. However, before the boom period, the STNN-D hybrid method generates RMSE of over 20% for 21-month horizon but stays under 20% for 11-month horizon. In other words, the STNN-D method produces more stable forecast results for 11-month horizon than 24-month.

In short, the STNN-D hybrid method captures steep growth of underlying trend better than the baseline method, but loses stability in the case of extra-long forecast horizons.

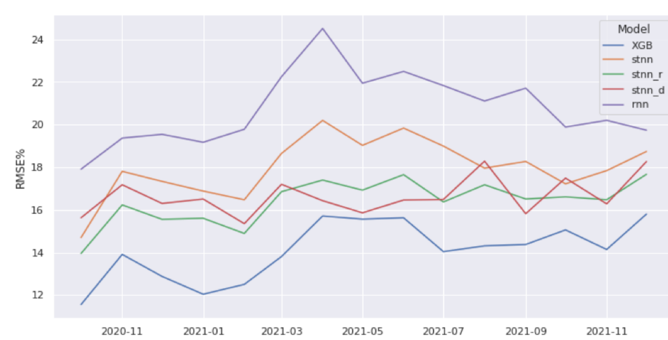

Figure 6: Results from individual models of 15 cross-validation with *tanh* activation function of 3-month forecast horizon. Values are percentage of RMSE over mean price per m^2 of every month of cross-validation.

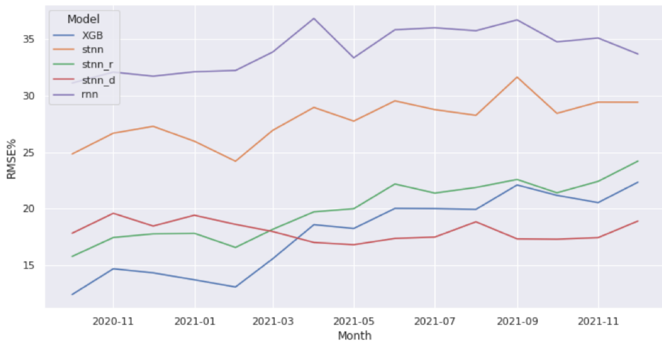

Figure 7: Results from individual models of 15 cross-validation with *tanh* activation function of 11-month forecast horizon. Values are percentage of RMSE over mean price per m^2 of every month of cross-validation.

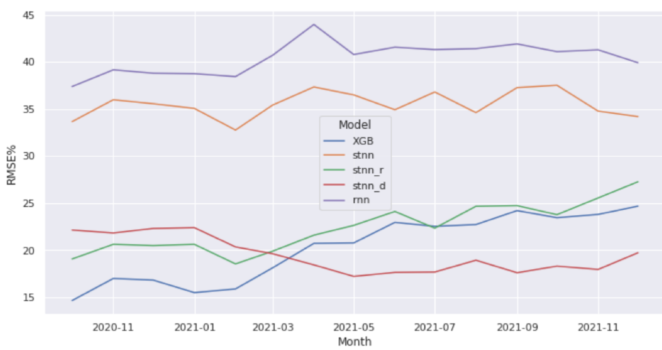

Figure 8: Results from individual models of 15 cross-validation with *tanh* activation function of 21-month forecast horizon. Values are percentage of RMSE over mean price per m^2 of every month of cross-validation.

6.3 Spatial Relations Refined and Discovered by STNN

Apart from forecast values of individual property price, the STNN-R/STNN-D hybrid methods also generate refined/discovered spatial correlation weight matrix as output, which provide insights to the housing market when compared to human-pre-defined input.

Figure 9 shows the spatial correlation matrices of pre-defined input for STNN, refined values from STNN-R and discovered values from STNN-D, from left to right respectively. All three matrices are from the last cross-validation period (Dec. 2021) of a 11-month forecast experiment, with *tanh* activation function. The *Gebieden* are sorted in ascending alphabetical order both horizontally and vertically and thus the matrices appear to be symmetric along the diagonal.

As described in Section 5.3, correlations of neighbouring *Gebieden* are pre-defined as 1.0 while pairs across lakes and River Amstel are assigned with 0.5 and 0.2 respectively. Compared to the pre-defined grid on the left, the STNN-R refined grid in the middle consists of visibly lower values, ranging from 0.3 to 0.8 for most cells. In other words, results from STNN-R indicate that neighbouring *Gebieden* have lower correlation in property prices than anticipated.

Results from STNN-D method are also unexpected - the discovered correlation grid on the right in Figure 9 contains far fewer cells with zero values than the pre-defined grid on the left. Besides, the range of values is narrower, with the highest correlation captured at 0.46 between *Bijlmer-Centrum* and *Gaasperdam / Driemond*. The STNN-D discovered spatial correlation values imply that there exist weak underlying correlations amongst disconnected or even distant *Gebieden*.

As the exact *Gebieden* pairs are obscure and the geographic locations are invisible from the grids, we further investigate the refined and discovered correlation values of two *Gebieden*, namely *Centrum-West* and *Bijlmer-Centrum*, on the map of Amsterdam. Figures 10 and 11 show the 22 *Gebieden* of Amsterdam included in this project with colour scaled by the refined and discovered spatial correlations respectively with *Centrum-West*, marked with a star. Figures 12 and 13 show the same plots for *Bijlmer-Centrum*.

6.3.1 Spatial Correlations of Centrum-West Gebied. *Centrum-West* lies in the centre of Amsterdam and is regarded as one of the *Gebieden* where the highest-priced properties are located. The STNN-R refined values show that *Centrum-West* has spatial correlations from 0.46 to 0.53 with three adjacent counterparts, *Westerpark*, *Oud-West / De Baarsjes* and *Centrum-Oost*. However, the highest correlation occurs with *Oud-Noord* across River Amstel, which contradicts to the common belief that housing properties on the north side of the city are worth much less than those on the south side. The STNN-D discovered results, however, are aligned with such belief and indicate only 0.07 correlation between the two *Gebieden* across the river. Moreover, the STNN-D method unveils relatively high correlation of *Centrum-West* to *De Pijp / Rivierenbuurt* (0.32) and *Watergraafsmeer* (0.31), which are both urban *Gebieden* non-adjacent to *Centrum-West*.

6.3.2 Spatial Correlations of Bijlmer-Centrum Gebied. Contrary to *Centrum-West*, *Bijlmer-Centrum* is a suburban area developed in recent decades in southeastern Amsterdam. As shown in Figure 12, due to the restrictions of STNN-R, algorithm-refined results suggest high correlations only with *Bijlmer-Oost* (1.0) and *Gaasperdam / Driemond* (1.0), the solely two adjacent *Gebieden*. Without the neighbouring restrictions, algorithm-discovered results of STNN-D, however, display no correlation to *Watergraafsmeer*, the closest non-adjacent *Gebied* northwest to *Bijlmer-Centrum*. Instead, output results manifest correlations to *Gebieden* on the brink of the city, such as *Osdorp* (0.27) and *Geuzenveld-Slotermeer-Sloterdijken* (0.32) in the west and *Noord-West* (0.33) and *Noord-Oost* (0.32) in the north across River Amstel, all four of which are newly developed areas of the city. These results suggest that spatial auto-correlations do not necessarily depend on geographic adjacency but alternatively on the development history of the area or characteristics such as distance to the city centre.

7 DISCUSSION AND LIMITATIONS

The experiment results in Section 6 show lower RMSE values from all three STNN models than RNN, indicating that spatio-temporal

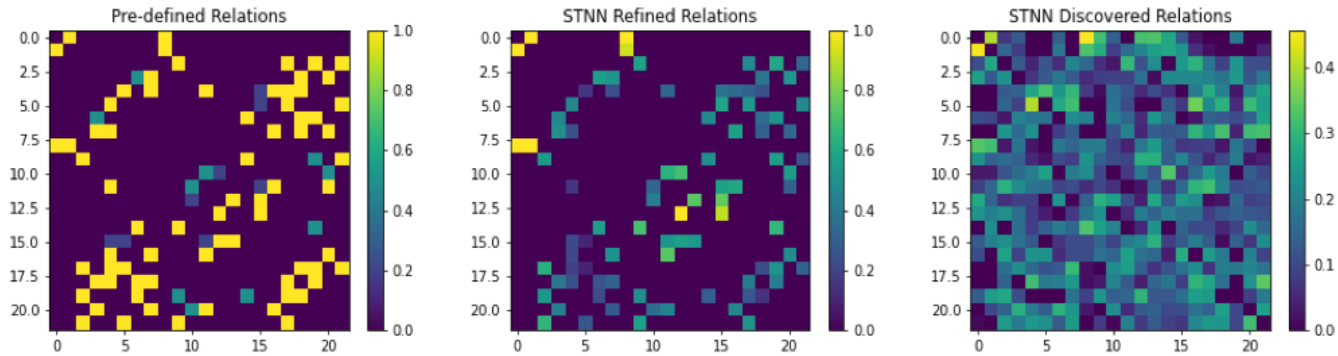

Figure 9: *Gebied* correlation weight matrix output from the last cross-validation (Dec. 2021) of 11-month forecast. *Gebieden* are sorted in ascending alphabetical order. Left: pre-defined matrix for STNN. Middle: output matrix from STNN-R. Right: output matrix from STNN-D.

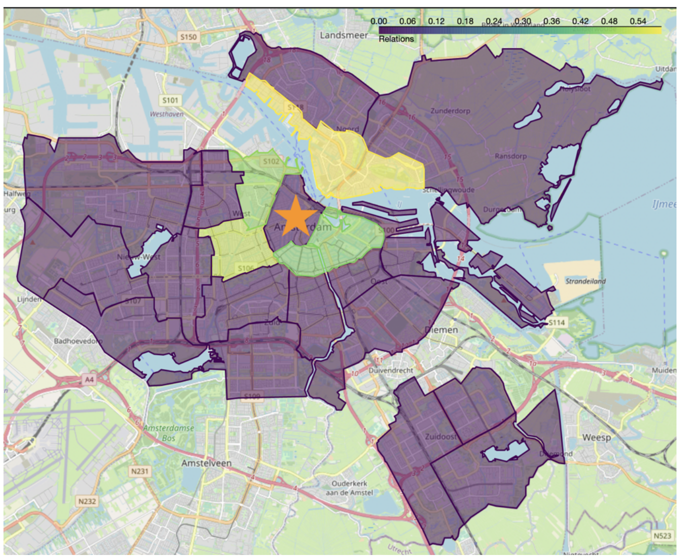

Figure 10: STNN-R refined spatial correlations of *Centrum-West Gebied*, scaled in colour, from the last cross-validation (Dec. 2021) of 11-month forecast.

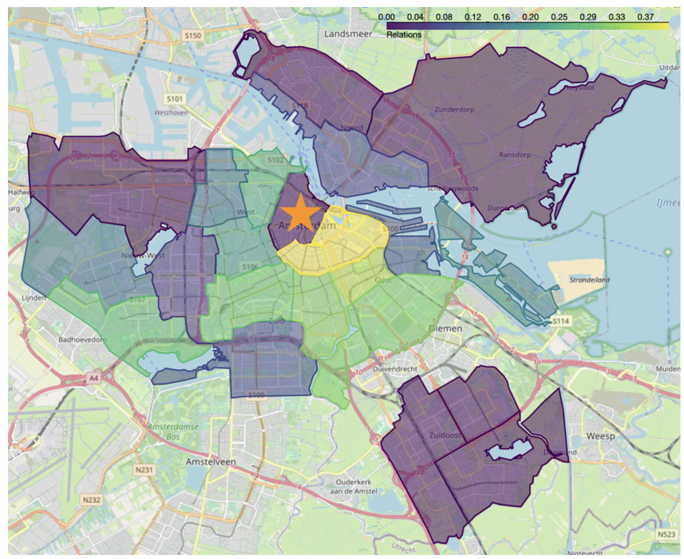

Figure 11: STNN-D discovered spatial correlations of *Centrum-West Gebied*, scaled in colour, from the last cross-validation (Dec. 2021) of 11-month forecast.

deep learning algorithms outperform their counterparts for temporal forecasting. Meanwhile, the three STNN models fail to beat XG-Boost on short forecast horizons and from Oct. 2020 to Mar. 2021 on long forecast horizons. This might be due to the strong predictive power of XG-Boost for in-period prediction but potentially over-fitting for long out-of-period forecast. However, the loss of information in the aggregation process of the iterative method cannot be ignored. The poor results from the RNN hybrid method, of which the aggregation level is highest, is strong evidence. Meanwhile, the HTM suggests a hierarchy of underlying trends of multiple neighbourhood levels or even house types [5]. And from domain knowledge, we learn that real estate investors consider location of properties at neighbourhood levels such as *Buurt* and *Wijk*. Thus, the loss of information of the STNN hybrid method might also be due to incoporating only one layer of spatial trend (*Gebied*) and the method could be improved by expanding the spatial dimensions. A possible solution is further discussed in Section 8.

Alternatively, a one-shot solution similar to Kalman filter and smoother for HTM [6] might also reduce the information loss in the aggregation process. The XG-Boost regressors are included in the hybrid method to separate the static and dynamic components, i.e. standardisation, which serve only as a work-around solution and would no longer be necessary if the static components are incorporated in the neural network. Details are also given in Section 8.

Contrary to XG-Boost, the process of hyper-parameter optimisation for neural networks could be time-consuming and does not always result in convergence. Therefore, instead of full grid-search, we scale the data panel and utilise the hyper parameter settings of STNN optimised by *Delasalles et al.* [3]. However, as experimental

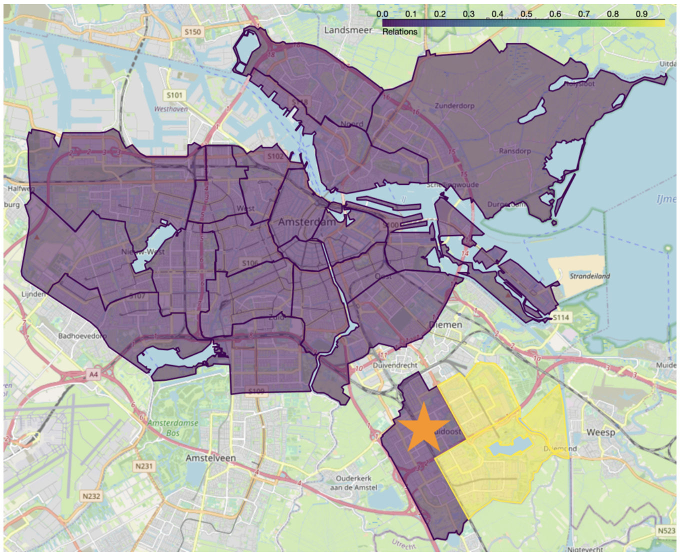

Figure 12: STNN-R refined spatial correlations of *Bijlmer-Centrum Gebied*, scaled in colour, from the last cross-validation (Dec. 2021) of 11-month forecast.

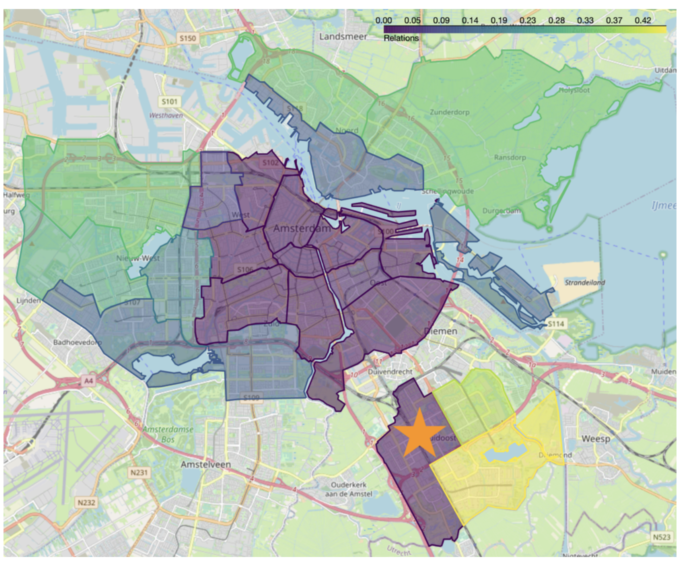

Figure 13: STNN-D discovered spatial correlations of *Bijlmer-Centrum Gebied*, scaled in colour, from the last cross-validation (Dec. 2021) of 11-month forecast.

results show that STNN produces lower forecast errors in booming period as described in Section 6.2, multiple hidden layers with other activation functions such as *Sigmoid* could be better configurations and the solution adopted in this project might have led to STNN algorithms under-performing.

8 CONCLUSION AND FUTURE WORK

This project aims to compare forecast performance on real estate appraisal by a spatio-temporal deep learning algorithm STNN to state-of-the-art machine learning algorithm XG-Boost and widely used temporal forecast algorithm RNN. Another objective of this project is to explore the insights on real estate market from the refined and discovered spatial correlations from the STNN-R and STNN-D models.

Based on the experimental results, we could conclude that STNN outperforms RNN on all forecast horizons with up to 22% smaller RMSE. STNN also beats the baseline method XG-Boost with up to 1.7% lower forecast errors on long forecast horizons and in booming periods of the housing market. The spatial correlations discovered by the STNN-D algorithm are closer to reality than pre-defined values and the refined output by STNN-R. The discovered results also suggest that spatial auto-correlations do not necessarily depend on adjacency but instead depend on the development history of the area or characteristics such as distance to the city centre.

This project could be extended and improved in several aspects. As mentioned in Section 7, the spatio-temporal panel could be constructed with different neighbourhood levels, e.g. *Buurt* and *Wijk*. This process could be further improved by defining synthetic spatial "bins" instead of administration-defined divisions. For instance, spatial "bins" could be defined by ranges of latitude and longitude values, which could average the size of each spatio-temporal "bin" and therefore stabilise the values of the spatio-temporal panel. Furthermore, as space and time are by nature continuous, converting the deep learning neural network to consume spatial and temporal features in continuous form could potentially result in forecasts of higher accuracy.

One fundamental assumption of the hybrid method for RNN and STNN of this project is that the worth of the characteristics of residential properties, such as number of rooms and area of living, is invariant over space and time and therefore considered "static". By breaking such assumption and incorporating the "static" components into the neural networks, the XG-Boost regressors and standardisation process could be dropped and forecast performance could be improved, as suggested in Section 7.

The STNN models utilised in this project are implemented using PyTorch tensor [2] objects with one temporal dimension and one spatial dimension. The tensor objects could be expanded with more spatial dimensions to incorporate multiple neighbourhood levels. Similarly, the numerical characteristic features, such as area of living and number of rooms, could be integrated with additional dimensions. The categorical characteristic features could either be transformed into binaries using one-hot encoding (such as Object Type) or transformed with ordinal scale in one dimension (such as Energy Label or Maintenance Level). The expanded STNN no longer relies on the assumption that characteristics of properties are spatio-temporally invariant and hence no longer requires aggregation or standardisation in the training process. In other words, one-shot training and optimisation is possible for the expanded STNN.

[2] https://pytorch.org/docs/stable/tensors.html

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisors Prof. Marc Francke and Dr. Alonso Acuña for their patience and guidance. I would also like to thank the Municipality of Amsterdam for the opportunity of this project. Last but not least, special thanks to Charmaine for her long-term mental support.

REFERENCES

[1] David A. Belsley, Edwin Kuh, and Roy E. Welsch. 2005. *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity.* John Wiley & Sons.

[2] Henry Crosby, Paul Davis, Theo Damoulas, and Stephen A. Jarvis. 2016. A spatio-temporal, Gaussian process regression, real-estate price predictor. *SIGSPACIAL '16: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (October 2016), 1–4. https://doi.org/10.1145/2996913.2996960

[3] Edouard Delasalles, Ali Ziat, Ludovic Denoyer, and Patrick Gallinari. 2019. Spatio-temporal neural networks for space-time data modeling and relation discovery. *Knowledge and Information Systems* 61 (January 2019), 1241–1267. https://doi.org/10.1007/s10115-018-1291-x

[4] A. Stewart Fotheringham, Ricardo Crespo, and Jing Yao. 2015. Exploring, modelling and predicting spatiotemporal variations in house prices. *The Annals of Regional Science* 54 (March 2015), 417–436. https://doi.org/10.1007/s00168-015-0660-6

[5] Marc K. Francke. 2008. The Hierarchical Trend Model. *Mass appraisal methods: An international perspective for property valuers* (August 2008), 164–180. https://doi.org/10.1002/9781444301021.ch8

[6] Marc K. Francke and A. F. de Vos. 2000. Efficient Computation of Hierarchical Trends. *Journal of Business and Economic Statistics* 18, 1 (January 2000), 51–57. https://www.jstor.org/stable/1392136

[7] Kelvin J. Lancaster. 1966. A New Approach to Consumer Theory. *Journal of Political Economy* 74, 2 (April 1966), 132–157. https://doi.org/10.1086/259131

[8] Michael Mayer, Steven C. Bourassa, Martin Hoesli, and Donato Scognamiglio. 2019. Estimation and Updating Methods for Hedonic Valuation. *Journal of European Real Estate Research* 12, 1 (April 2019), 134–150. https://doi.org/10.1108/JERER-08-2018-0035

[9] Yue Ming, Jie Zhang, Jiaming Qi, Tian Liao, Maolin Wang, and Lingli Zhang. 2020. Prediction and Analysis of Chengdu Housing Rent Based on XGBoost Algorithm. *ICBDT 2020: Proceedings of the 2020 3rd International Conference on Big Data Technologies* (September 2020), 134–150. https://doi.org/10.1145/3422713.3422720

[10] Yi-Shuai Niu, Wentao Ding, Junpeng Hu, Wenxu Xu, and Stephane Canu. 2021. Spatio-Temporal Neural Network for Fitting and Forecasting COVID-19. *Knowledge and Information Systems* (March 2021). https://doi.org/10.48550/arXiv.2103.11860

[11] R. Kelly Pace, Ronald Barry, Otis W. Gilley, and C. F. Sirmans. 2000. A method for spatial–temporal forecasting with an application to real estate prices. *International Journal of Forecasting* 16, 2 (April 2000), 229–246. https://doi.org/10.1016/S0169-2070(99)00047-3

[12] David E. Rumelhart, Geoffery E. hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323 (October 1986), 533–536. https://doi.org/doi.org/10.1038/323533a0

[13] Sarah Sayce, Judy Smith, Richard Cooper, and Piers Venmore-Rowland. 2006. *Real Estate Appraisal: From Value to Worth* (1st ed.). Oxford: Blackwell.

[14] Carola de Groot Stefan Groot. 2021. *Price Growth Hit Its Peak, but Dutch Housing Market Is Far From Cooling Down.* Retrieved February 16, 2022 from https://economics.rabobank.com/publications/2021/december/price-growth-hit-its-peak-but-dutch-housing-market-is-far-from-cooling-down/

[15] Carola de Groot Stefan Groot. 2022. *Double-digit house price growth this year, despite increased uncertainty.* Retrieved March 22, 2022 from https://economics.rabobank.com/publications/2022/march/dutch-housing-market-quarterly/

[16] Amsterdam Tips. 2022. *Amsterdam Property Price Map 2013-2020.* Retrieved January 13, 2022 from https://www.amsterdamtips.com/amsterdam-house-prices/

[17] Mahdieh Yazdani. 2021. Machine Learning, Deep Learning, and Hedonic Methods for Real Estate Price Prediction. (October 2021). arXiv:2110.07151

[18] Sanglim Yoo, Jungho Im, and John E. Wagner. 2012. Variable selection for hedonic model using machine learning approaches: A case study in Onondaga County, NY. *Elsevier* (July 2012). https://doi.org/10.1016/j.landurbplan.2012.06.009

[19] Chaoyun Zhang and Paul Patras. 2018. Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks. *Mobihoc '18: Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (June 2018), 231–240. https://doi.org/10.1145/3209582.3209606

A SUMMARY OF DATASET

Feature	Mean	Std.	Min.	25%	50%	75%	Max.
Transaction Price per m^2	4742.06	2019.67	22.73	3229.17	4368.42	6057.69	19942.20
Construction Year	1948.09	51.70	1530	1919	1951	1989	2020
Area of Living (m^2)	88.33	48.79	9	58	78	104	1200
Number of Rooms	3.41	1.47	1	3	3	4	27

Table 2: Response variable and numerical features. This table displays the distribution of the response variable, transaction price per m^2, before log-transformation as well as all the numerical features of the whole dataset.

Feature	Type	Unique Values	Value of Most Occurance	Frequency
Gebied	Categorical	22	*De Pijp / Rivierenbuurt*	10.22%
Object Type	Categorical	6	Apartment	87.50%
Energy Label	Categorical	7	Unclassified	25.85%
Maintenance Level	Categorical	9	Good	65.69%
Building Form	Binary	2	Existing Building	97.60%
Single-/Multi-family Type	Binary	2	Multi-family Type	87.50%
Parking	Binary	2	False	85.42%
Furnished	Binary	2	False	99.64%
Upholstered	Binary	2	False	91.73%

Table 3: Categorical and binary features. This table shows a summary of all the categorical and binary features of the whole dataset.

B HYPER-PARAMETER SELECTIONS FOR GRID SEARCH OF XG-BOOST

- Number of gradient boosted trees ∈ [500]
- Maximum depth of each tree ∈ [10, 15]
- Gradient boosting learning rate ∈ [0.03, 0.05, 0.07]
- Sub-sample ratio of the training instance ∈ [0.5, 0.7]
- Sub-sample ratio of features when constructing each tree ∈ [0.5, 0.7]
- L1 regularization term on weights (alpha) ∈ [0.01, 0.1, 1.0]
- L2 regularization term on weights (lambda) ∈ [0.01, 0.1, 1.0]

C TABLE OF RMSE OF INDIVIDUAL FORECAST HORIZONS

Model	Act. Function	Forecast Horizons (Months)								
		1	3	6	9	12	15	18	21	24
XG-Boost		**13.34**	**14.22**	**15.81**	**17.34**	18.33	19.04	19.79	20.63	21.37
RNN	identity	15.70	19.74	26.55	31.68	34.95	37.76	39.19	41.34	42.16
	tanh	15.95	20.97	27.56	32.38	34.98	37.34	39.20	40.62	41.51
STNN	identity	15.72	16.80	19.48	22.45	24.72	27.75	30.30	32.22	34.55
	tanh	15.95	18.16	22.38	26.20	28.84	31.81	33.83	35.68	37.46
STNN-R	identity	15.54	16.13	16.99	17.61	**17.89**	**18.09**	**18.25**	**18.86**	**20.37**
	tanh	15.57	16.47	18.03	19.45	20.36	20.96	21.69	22.50	23.21
STNN-D	identity	15.93	17.31	18.31	18.98	20.12	20.81	23.82	27.19	39.78
	tanh	15.78	16.64	17.23	17.85	18.22	18.54	19.11	19.51	20.50

Table 4: Results from individual models of cross-validation experiments of different forecast horizons. Values are percentage of RMSE over mean price per m^2 of every forecast horizon. Best results of every horizon are in bold.

D TABLES OF RMSE OF INDIVIDUAL FORECAST PERIODS

Model	2020			2021											
	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
XG-Boost	**11.56**	**13.91**	**12.88**	**12.04**	**12.50**	**13.81**	**15.71**	**15.56**	**15.63**	**14.04**	**14.31**	**14.37**	**15.06**	**14.14**	**15.80**
RNN	17.91	19.37	19.54	19.17	19.78	22.25	24.51	21.94	22.50	21.83	21.11	21.71	19.89	20.20	19.74
STNN	14.70	17.81	17.34	16.88	16.47	18.65	20.20	19.03	19.84	18.99	17.96	18.27	17.22	17.84	18.74
STNN-R	13.95	16.23	15.56	15.60	14.89	16.85	17.40	16.93	17.65	16.37	17.18	16.51	16.61	16.47	17.67
STNN-D	15.63	17.18	16.30	16.51	15.35	17.20	16.43	15.86	16.46	16.49	18.29	17.49	17.49	16.28	18.27

Table 5: Results from individual models of 15 cross-validation with *tanh* activation function of 3-month forecast horizon. Values are percentage of RMSE over mean price per m^2 of every month of cross-validation. Best results are in bold.

Model	2020			2021											
	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
XG-Boost	**12.40**	**14.68**	**14.32**	**13.70**	**13.07**	**15.56**	18.58	18.24	20.02	20.00	19.93	22.09	21.18	20.53	22.34
RNN	31.12	32.08	31.72	32.11	32.23	33.86	36.83	33.35	35.83	36.01	35.75	36.71	34.76	35.10	33.69
STNN	24.84	26.68	27.28	25.97	24.20	26.93	28.96	27.75	29.54	28.77	28.26	31.64	28.44	29.43	29.42
STNN-R	15.77	17.44	17.77	17.82	16.57	18.17	19.72	19.99	22.18	21.38	21.87	22.58	21.40	22.41	24.21
STNN-D	17.82	19.59	18.46	19.42	18.61	17.98	**17.00**	**16.81**	**17.36**	**17.48**	**18.84**	**17.32**	**17.30**	**17.43**	**18.90**

Table 6: Results from individual models of 15 cross-validation with *tanh* activation function of 11-month forecast horizon. Values are percentage of RMSE over mean price per m^2 of every month of cross-validation. Best results are in bold.

Model	2020			2021											
	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.	Apr.	May	Jun.	Jul.	Aug.	Sep.	Oct.	Nov.	Dec.
XG-Boost	**14.66**	**17.00**	**16.83**	**15.50**	**15.89**	**18.14**	20.73	20.78	22.96	22.54	22.73	24.21	23.46	23.81	24.69
RNN	37.40	39.17	38.81	38.76	38.45	40.72	43.99	40.80	41.58	41.32	41.42	41.92	41.10	41.29	39.93
STNN	33.67	35.99	35.57	35.07	32.77	35.43	37.35	36.51	34.93	36.82	34.62	37.28	37.53	34.78	34.20
STNN-R	19.07	20.63	20.50	20.63	18.55	19.90	21.60	22.64	24.12	22.35	24.67	24.74	23.78	25.55	27.28
STNN-D	22.15	21.83	22.31	22.40	20.35	19.63	**18.44**	**17.22**	**17.65**	**17.68**	**18.94**	**17.61**	**18.31**	**17.96**	**19.73**

Table 7: Results from individual models of 15 cross-validation with *tanh* activation function of 21-month forecast horizon. Values are percentage of RMSE over mean price per m^2 of every month of cross-validation. Best results are in bold.