Attention-Based Spatial Interpolation for House Price Prediction

Darniton Viana
dav2@cin.ufpe.br
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brazil

Luciano Barbosa
luciano@cin.ufpe.br
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brazil

ABSTRACT

Estimating the market price of a house is important for many businesses such as real estate and mortgage lending companies. The price of a house depends not only on its structural features (e.g. area and number of bedrooms) but also on the spatial context where it is located. In this work we estimate the price of a house based solely on its structural features and the characteristics and price of its neighbors. For that, we propose a hybrid attention mechanism that weights neighbors based on their similarity to the house in terms of structural features and geographic location. For the structural features, we apply an euclidean-based attention and, for the geographic location, we propose an attention layer based on a radial basis function kernel. Those attention mechanisms are then used by a neural network regressor to predict the price of a house and to generate a vector representation of the house based on its implicit context: the house embedding, which can be used as a feature set by any regressor to perform house price prediction. We have performed an extensive experimental evaluation on real-world datasets that shows that: (1) regressors using house embedding obtained the best results on all 4 datasets, outperforming baseline models; (2) the learned house embedding improves the performance of the evaluated regressors in almost all scenarios in comparison to raw features; and (3) simple regressor models such as Linear Regression using house embedding achieved comparable results to more competitive algorithms (e.g. Random Forest and Xgboost).

CCS CONCEPTS

• **Information systems → Geographic information systems**.

KEYWORDS

Spatial Interpolation, Attention Mechanism, House Price Prediction

ACM Reference Format:
Darniton Viana and Luciano Barbosa. 2021. Attention-Based Spatial Interpolation for House Price Prediction. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

E-commerce websites have started to offer price prediction tools to their customers (buyers or sellers) to help them in their process of decision making. This feature is particularly useful to sellers for offer price suggestion and to buyers to compare the price of a selling product to its estimated value. Kelley Blue Book[1] website provides such feature with its Price Advisor tool that estimates the market value of cars. The traveling website Kayak[2] also offers some kind of price prediction by suggesting to users to book a flight or wait based on its forecast price model. In the real estate domain, the website Zillow[3] provides to its users predictions of the market value of houses. To improve the accuracy of its prediction model, they launched in 2017 a 1-million prize competition on Kaggle[4]. This shows the importance of this feature for their business and how challenging it is to perform such task. In this work, we are particularly interested in the problem of house value prediction due to the high value of this asset for people's life and its importance to different businesses such as real estate and mortgage lending companies.

Many factors are relevant to accurately predict the value of a house. The structural features of a house (e.g., number of bedrooms and area) have certainly high influence on its price. The spatial context where the house is located is another valuable factor for price prediction. Houses close to subway stations and parks may have higher values than the ones that do not have such points of interest in their neighborhood. Previous approaches [3, 15, 22, 29] have tried to explicitly capture the spatial context of a house for price prediction by collecting satellite images, points of interest, census and criminality data of its neighborhood or external images of the house. Another way to obtain the spatial context of a house is implicitly by looking at the price of nearby houses: houses with similar structural features and geographically close tend to have similar values.

Since collecting and processing data to explicitly capture the spatial context can be very costly, in this work, we aim to estimate the value of a house based solely on its structural features and characteristics and price of neighboring houses. We model therefore house price prediction as a spatial interpolation problem [19], whose main assumption is that spatially distributed objects are spatially correlated. Traditional interpolation models, such as Inverse Distance Weighting [25], Radial Basis Function [8], and Kriging [17], try to do this by simply weighting the influence of neighboring points based on some pre-defined measures (e.g. inverse distance or spatial variance). Since calculating the neighboring points can have a high

[1] www.kbb.com
[2] www.kayak.com
[3] www.zillow.com
[4] https://www.kaggle.com/c/zillow-prize-1/

computational cost on large datasets, data structures such as k-d tree [2], and uniform grid [18] have been proposed to speed up this process.

In this work, we introduce a *novel spatial interpolation method based on the attention mechanism* [5] [1] that, as opposed to the traditional spatial interpolation approaches, weights the neighbor's influence based on supervised learning. More specifically, we propose an *euclidean-based attention layer* to weight the neighboring houses based on the similarity of their structural features to the targeting house; and a *spatial kernel-based attention layer* based on a Radial Basis Function, which we called Geo Attention, to weight neighbors based on their geographic distance to the targeting house. The vectors created by the attention layers added to the geo location and structural features of the house are fed into a fully-connected network which produces a vector to the regression layer (a single neuron with an activation function) to perform house price prediction. This vector, which we call house embedding, embeds the house's attributes and its spatial context into a common subspace. Our proposed network is therefore a *fixed feature extractor* for the structural features of houses and their spatial context. As a result, the house embedding can be used as input feature set by any regressor to estimate the price of a house.

We have performed an extensive experimental evaluation on 4 datasets. The results show that: (1) the evaluated regressors using house embedding obtained the best results on all 4 datasets, outperforming traditional spatial interpolators and previous deep learning approaches of house price prediction that extract features from images to capture the spatial context, which is very costly, as opposed to our approach that only relies on information of neighboring house such as their price, structural features and geographic location to perform this task; (2) the learned house embedding improves the performance of the evaluated regressors in almost all scenarios in comparison to raw features; and (3) simple regressor models such as Linear Regression using house embedding achieved better or comparable results to more competitive algorithms, such as Random Forest and Xgboost.

The remainder of this paper is organized as follows. In Section 2, we define the problem we are dealing with in this work. Section 3 presents in details our proposed attention network. We describe and analyze the data used in our evaluation in Section 4. The experimental evaluation is presented in Section 5. Finally, we discuss some previous approaches to the problem of house price estimation in Section 6 and conclude in Section 7, wherein we outline directions and future work.

2 PROBLEM DEFINITION

Let $\mathbf{A}_i = \{a_{i,1}, a_{i,2}, \cdots, a_{i,T}\} \in R^T$ be the T structural features of a house i; $\mathbf{G}_i = \{lat_i, lng_i\} \in R^2$, the geographic coordinates (latitude and longitude) of i; $\mathrm{C}_i = \{ (\mathrm{A,G})_{i,1}, (\mathrm{A,G})_{i,2}, \cdots, (\mathrm{A,G})_{i,n}\} \in R^{n\times(T+2)}$ the structural and geographic features of the n geographically nearest houses of i, where $(\mathrm{A,G})_{i,k}$ represents the structural features $(A_{i,k})$ and geographic coordinates $(\mathrm{G}_{i,k})$ of the k-th neighbor of house i; and $\mathbf{Y}_i = \{\mathrm{y}_{i,1}, \mathrm{y}_{i,2}, \mathrm{y}_{i,3}, \cdots, \mathrm{y}_{i,n}\} \in R^n$ the prices of the n geographically nearest houses of i, where $y_{i,k}$ is the price of the k-th neighbor of house i,

[5]https://github.com/darniton/ASI

Definition 1 (Problem Definition). Given $(\mathbf{A},\mathbf{G})_\mathbf{i}$: the structural features (A_i) and geographic coordinates (G_i) of a house i; $\mathbf{C}_i$: the features of the n-nearest neighbors of house i; and $\mathbf{Y}_i$: the prices of the n-nearest neighbors of house i, we aim to estimate the price of i: $\hat{y}_i \in R$.

3 ATTENTION-BASED NETWORK

In this work, we model house price estimation as a spatial interpolation problem [26]. The main assumption of spatial interpolation is that spatially distributed objects are spatially correlated. In our context, we assume that houses with similar structural attributes and geographically close tend to have similar prices. Based on that, to estimate the value of a house our solution relies on the price and characteristics of the houses in its vicinity. Figure 1 presents the hybrid attention-based network that we propose in this work. The input of the network are the attributes of the house i that we aim to estimate the price, composed of its structural features $(\mathbf{A}_i)$ and geographic coordinates $(\mathbf{G}_i)$, and the structural and geographic attributes of the n-nearest houses $(\mathbf{C}_i)$. The euclidean attention weights the influence of the structural features of neighboring houses based on the euclidean distance between them and $\mathbf{A}_i$; and the geographical attention (Geo Attention) learns the spatial correlations between the n-nearest geographic neighbors of the house i, implemented through a spatial kernel-based attention. The output vectors of the two attention layers (v_{euc} and v_{geo}) are concatenated with $\mathbf{A}_i$ and $\mathbf{G}_i$ and fed into a fully-connected neural network (hidden layers), which provides the input to the regression layer (a single neuron with an activation function). The output of the hidden layers embeds the influence of neighbors and the house information on the house's price into a single vector: the house embedding. In the remaining of this section, we give further details about the attention mechanisms used by the network.

3.1 Euclidean Attention

One of our main assumptions in this work is that houses in a same or related region with similar profiles tend to have similar prices. The euclidean attention tries to model that by weighting the houses in the vicinity based on their structural features. That is, nearby houses with similar profile to the targeting house might have high influence in estimating its price.

As depicted in Figure 1, the inputs to the euclidean attention are the structural features of the house i: A_i, and the structural features of the n closest houses to i: $\mathrm{S} = \{A_{i,1}, A_{i,2}, \cdots, A_{i,n}\}$. The score between the structural features of a house $A_{i,j} \in S$ and A_i is the euclidean distance between them:

$$d(A_i, A_{i,j}) = \sqrt{\sum_{p=1}^{T} \left(a_{i,p} - a_{(i,j),p}\right)^2} \tag{1}$$

where $a_{i,p}$ is the p-th structural attribute of A_i and $a_{(i,j)_p}$ is the p-th structural feature of $A_{i,j}$.

To calculate the attention weights for each neighbor j based on their structural features, first the model feeds the input vector $D \in R^n$, which contains the eucledian distances between A_i and its neighbors S, to a fully-connected layer to obtain $H \in R^n$, a hidden representation of D:

$$H = W.D + b \tag{2}$$

where $W \in R^{n \times n}$ is a matrix with weights learned during training and b the bias factors. A softmax layer is then applied on H to produce the normalized attention weight of each neighbor j:

$$a_{str}(A_i, A_{i,j}) = \frac{exp(H_j)}{\sum_{j'=1}^{n} exp(H_{j'})} \tag{3}$$

Finally, the euclidean-attention layer computes the attention vector $v_{euc}(A_i) \in R^{T+1}$:

$$v_{euc}(A_i) = \sum_{j=1}^{n} a_{str}(A_i, A_{i,j})\,[A_{i,j} \oplus y_{i,j}] \tag{4}$$

where $y_{i,j}$ is the price of the neighbor j of house i and $\oplus$ the concatenation operator. The dimension of $v_{euc}(A_i)$ is $T+1$ since $A_{i,j} \in R^T$ and $y_{i,j} \in R^1$. As Equation 4 shows, $v_{euc}(A_i)$ is calculated by first multiplying the vector $[A_{i,j} \oplus y_{i,j}]$ of each neighbor j of house i by its corresponding attention weight $a_{str}(A_i, A_{i,j})$, producing a weighted vector for each neighbor j, and then performing an element-wise sum on these weighted vectors over all n neighbors of i. The elements of $v_{euc}(A_i)$ are therefore the weighted sum of each structural feature and price of the neighbors of house i, where the attention weights are learned during training.

3.2 Geo Attention

Taking into consideration the structural features of nearby houses is important for house price prediction but, based on the spatial interpolation assumption, closer houses in the neighborhood, even with a different profile, tend to have more influence in the price of a house than distant ones.

Based on this observation, we propose an attention mechanism that weights neighbors based on their geographic distance to the targeting house, which we called geographic attention or simply Geo Attention. This attention mechanism is very related to traditional spatial interpolation methods such as Inverse Distance Weighting [25] and Radial Basis Functions [8], which also weight neighbors based on geographic distance. But different from them, our attention layer learns, using supervised learning, weights based on a kernel function that determines the spatial influence of the neighboring houses for price prediction. More specifically, as presented in Figure 1, Geo Attention receives as inputs the geographic coordinates (G_i) of the house i and the geographic coordinates of its neighboring houses: $P = \{G_{i,1}, G_{i,2}, \cdots, G_{i,n}\}$. The geographic score between G_i and an element $G_{i,j} \in P$ is computed using the Gaussian kernel function [29]:

$$s(G_i, G_{i,j}) = exp\Big(-geo\_dist(G_i, G_{i,j})\rho\Big) \tag{5}$$

$$\rho = \frac{\sigma^2}{2} \tag{6}$$

where $geo\_dist(G_i, G_{i,j})$ is the geodesic distance between G_i and $G_{i,j}$, and σ is the hyper-parameter that controls the similarity decaying with respect to the distance. This kernel is used to calculate the geo score between G_i and all elements in P, producing the vector $L \in R^n$. L is the input of a fully-connected layer that outputs $H' \in R^n$:

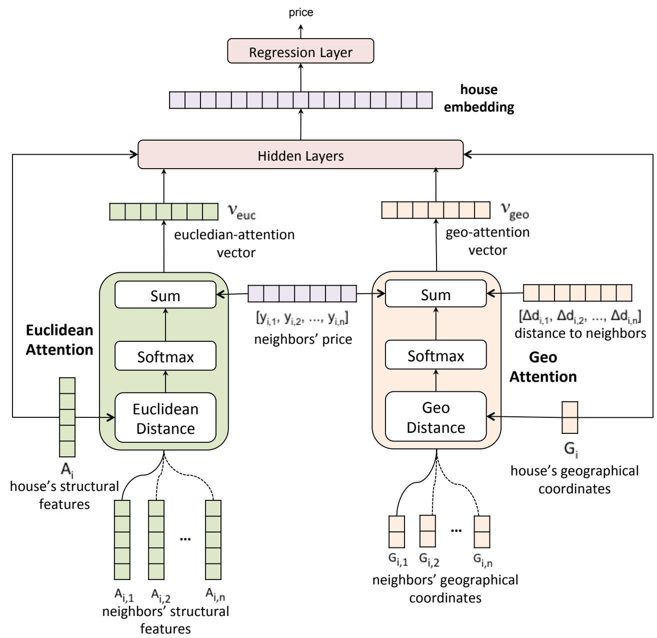

Figure 1: The attention network proposed in this work.

$$H' = W'.L + b' \tag{7}$$

where $W' \in R^{n \times n}$ is a matrix with weights learned during training and b' the bias factors. The normalized geo attention weights for each neighbor j is computed by a softmax layer:

$$a_{geo}(G_i, G_{i,j}) = \frac{exp(H'_j)}{\sum_{j'=1}^{n} exp(H'_{j'})} \tag{8}$$

Lastly, the geo attention vector $v_{geo}(G_i) \in R^4$ is calculated by:

$$v_{geo}(G_i) = \sum_{j=1}^{n} a_{geo}(G_i, G_{i,j})\,[G_{i,j} \oplus A_{i,j} \oplus \Delta d_{i,j} \oplus y_{i,j}] \tag{9}$$

where $\Delta d_{i,j}$ is the geographic distance between house i and its neighbor j, $y_{i,j}$ is the price of the neighbor j, and $\oplus$ the concatenation operator. The dimension of $v_{geo}(G_i)$ is the sum of $G_{i,j} \in R^2$, $A_{i,j} \in R^T$, $\Delta d_{i,j} \in R^1$ and $y_{i,j} \in R^1$. The resulting vector $v_{geo}(G_i)$ is hence the weighted sum of the vectors $G_{i,j}$ concatenated with $\Delta d_{i,j}$ and $y_{i,j}$ weighted by the learned normalized geo attention weights.

3.3 Model Training

To train the model and be able to create the house embedding, on top of the network we use a regression layer, which is a single neuron with an activation function chosen empirically, as we show in Section 4, that receives the house embedding as input. During training, the network minimizes the following loss function:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} |log(y_i) - log(\hat{y}_i)| \tag{10}$$

Table 1: Information about the 4 datasets used in the evaluation.

Region	Attr	Samples	Mean	Std
SP	6	68,848	741,952	411,643
POA	6	15,368	443,798	228,517
FC	12	83,136	155,164	76,507
KC	19	21,608	540,098	367,156

where y_i is the actual price of house i, $\hat{y}_i$ is the predicted value, θ are all learnable parameters in the proposed model, and N the number of houses in the training data. We use Adam optimizer algorithm [14] to minimize the loss function with respect to θ. The backpropagation algorithm is used to compute the parameters of the network.

4 DATA OVERVIEW

In this section, we present and analyze the datasets of the real estate listings we used in the evaluation of our work.

4.1 Data Description

We evaluated our house embedding on 4 datasets. Two of them are from USA counties: King County[6] (KC) in Washington State and Fayette County[7] (FC) in Kentucky, the other two datasets, São Paulo (SP) and Porto Alegre (POA), are from two Brazilian cities, collected from hundreds of Brazilian real-estate Web websites using a web crawler. Table 1 provides some details about the 4 datasets: they varied in terms of number of attributes, number of samples, mean selling price and standard deviation of the price.

4.2 Data Analysis

Figure 2a presents the distribution of the average geodesic distance in kilometers of the 60 nearest neighbors of the houses in each dataset. The numbers show that there is a great variation in terms of geographic density across the datasets: the FC dataset is the densest one with median average distance of about 0.18 Km whereas the KC dataset is the least dense with median average distance of 0.74 km. Figure 2b presents the distribution of the average euclidean distance of the structural features of the 60 nearest neighbors of the properties in each dataset. We normalized the distance for each dataset using min-max. The distributions depict major differences between the datasets: houses in FC and KC datasets are much more homogeneous in terms of structural features with median averages of 0.025 and 0.002 respectively than houses on the SP (0.21) and POA (0.12) datasets .

Recalling that we model house price prediction as a spatial interpolation problem, whose main assumption is that spatially distributed objects are spatially correlated, we empirically verify whether the price of the houses in those datasets holds this property. For that, we calculate the semivariance [28] on each dataset. Semivariance is used to measure spatial correlation between values of observations $Z(X)$ at different locations in space. In our context,

[6]https://www.kaggle.com/harlfoxem/housesalesprediction

[7]https://www.cs.uky.edu/ zach/publications/bessinger2016quantifying

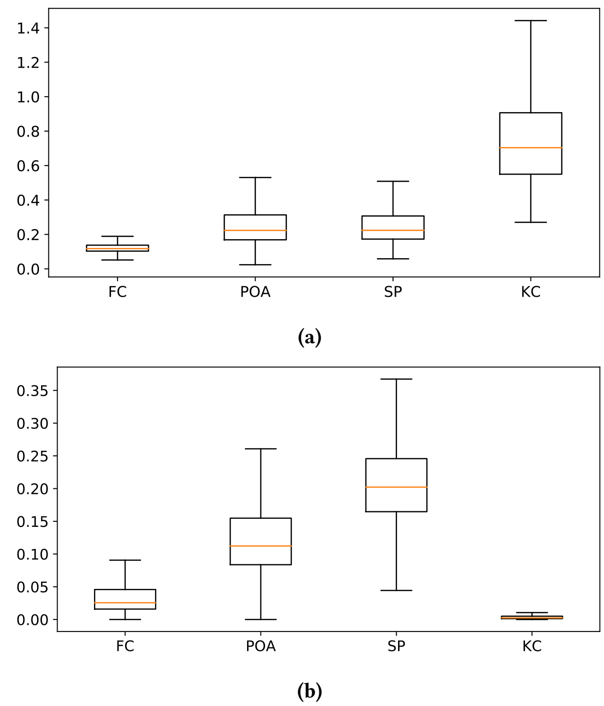

Figure 2: (a) Variation in quantiles of the average geodesic distance (in km) of 60 nearest houses for the 4 datasets. (b) Distribution in quantiles of the average euclidean distance of 60 nearest houses in terms the houses' structural features for the 4 datasets. We normalized the distance values for each dataset using min-max normalization since the datasets have different attributes for the house.

$Z(X)$ is the price of a house X. Semivariance is estimated from the data as follows:

$$\hat{\gamma}(h) = \frac{1}{2N} \sum_{i=1}^{N} (Z(X_i + h) - Z(X_i))^2 \quad (11)$$

where N is the number of pairs of sample observations separated by distance h. The plot of $\hat{\gamma}(h)$ versus h is known as the experimental semivariogram [28]. As a measure of variance, the lower its value, the higher the homogeneity of the phenomenon under study. Figure 3 presents the experimental semivariogram[8] of the log of the house prices in the four datasets varying the distance h from 0 to 1 Km. The semivarigramns show that there is a spatial correlation in the house prices in all cities, i.e., the closer the houses, the higher the price auto-correlation. The price of houses in KC has the lowest semivariance compared to the other ones: houses have semivariance of 0.05 when their distance is close to 0 Km, and 0.14 with the distance of 1Km. FC has a similar semivariance with distance close to 0 Km (0.06), but this value significantly increases as the houses get distant to each other: semivariance of 0.17 when the distance is 1 Km. Regarding the other datasets (SP and POA),

[8]To generate the experimental semivariogram, we used the variog function of the geoR package (cran.r-project.org/web/packages/geoR/). The following parameters were used: type (classical), direction (omnidirectional), and max.dist (1).

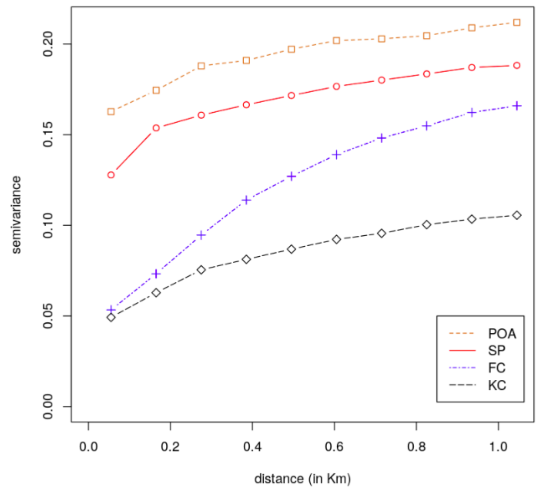

Figure 3: Experimental semivariogram of the 4 datasets.

the semivariances of the price of the houses are much bigger. For POA, for instance, the semivariance of close houses (distance close to 0) is 0.16 and 0.21 for houses with the distance of 1Km.

To further illustrate the spatial auto-correlation between the properties on the datasets, Figure 4 shows the spatial distribution of house prices by quartiles. As one can see, there are spatial clusters of houses with similar prices on all datasets. In addition, there are no abrupt changes of magnitude: houses in the top quartile are not, in most of the cases, close to properties in the lowest quartile.

5 EXPERIMENTS

In this section, we assess the effectiveness of house embedding on different regressors and compare our solution with previous approaches.

5.1 Experimental Setup

Data Split. For SP, POA and KC, we split each dataset in 72% for training, 8% for validation and 20% for test. For FC, we used the training and test sets provided by the authors of the dataset. To make the predictions on houses in the test set, we only consider the neighboring houses present on the training set.

Regressors. We executed the following regressors to evaluate the impact of using house embedding as a feature set on their performance:

- **Linear Regression (LR)**: regular Linear Regression with its default settings in its scikit-learn [21] implementation .
- **Random Forest (RF)** [7]: random forests are ensembles of tree-based models. We used the RF implementation on scikit-learn. The parameters of the RF were optimized by a cross-validation grid-search. We varied the parameter number of trees with the following values: 50, 100, 200, 700 and 1000. We used the version implemented on scikit-learn.
- **Lightgbm (LG)** [13]: lightgbm is a Gradient Boosting Decision Tree. We used a cross-validation grid-search as well varying the following values for the attribute number of trees: 50, 100 and 200; number of leaves: 3, 4, 5, 100 and 300; and learning rate: 0.03, 0.05, 0.07 and 0.1. We executed the implementation available by its authors[9].
- **Xgboost (XB)** [9]: Xgboost is another implementation of Gradient Boosting Decision Tree. We ran a Python package[10] optimizing the following parameters by a cross-validation grid-search: minimum child weight (4, 10 and 20), gamma (0.01, 1, 1.5 and 5), subsample (0.4, 0.2 and 0.6), column sample by tree (0.1, 0.5 and 1.0), learning rate (0.05, 0.1 and 0.01) and max depth (50, 100 and 200).
- **Auto-sklearn (AS)** [11]: AS is a automated machine learning (AutoML) toolkit[11] that performs algorithm selection, hyparemeter tunning and builds ensemble of predictors. Those ensembles are composed of individual regression models with a weight associated to each one of them. AS contains a great diversity of regression models such as Support Vector Regression, KNN, Adaboost, Ridge Regression and so on. We used the following parameters' values for training the models: time_left_for_this_task:39600; per_run_time_limit:30, and ml_memory_limit:6144.
- **Regression Layer (RL)**: this regressor is the last layer from our attention model, which produces the price prediction given the resulting feature map (house embedding) from the previous layers.

Feature Sets. For each regressor, we built models using 5 different feature sets:

- HA: the structural features and the geographic coordinates of the house available on the respective dataset.
- HA+HC: the structural features and geographic coordinates of the house, and the structural features and geographic coordinates of its neighbors and their prices available on the respective dataset.
- HA+POI: the structural features and the geographic coordinates of the house and the POI feature set based on the POIs around it;
- HE: the house embedding of dimension 50 built by our model;
- HE+POI: the house embedding and the POI feature set based on the POIs around it.

Spatial Interpolators. We executed the following traditional spatial interpolation models which, similar to our attention model, try to implicitly capture spatial context from the prices of houses in the vicinity:

- **Inverse Distance Weight (IDW)** [25] is an interpolator that considers the existence of spatial autocorrelation. It estimates the value of a house based on the prices of their neighbors weighted by their inverse distance to it. We executed it using the gstat [20] package of the R language with the following parameters: maxdist: inf and inverse distance weighting power: 2.
- **Universal Kriging (UK)** [17] is another model that considers spatial autocorrelation. In addition to the house's spatial context, UK uses its structural features to perform the prediction. To train the models, we used all the features available

[9]https://github.com/microsoft/LightGBM
[10]https://xgboost.readthedocs.io/
[11]https://automl.github.io/

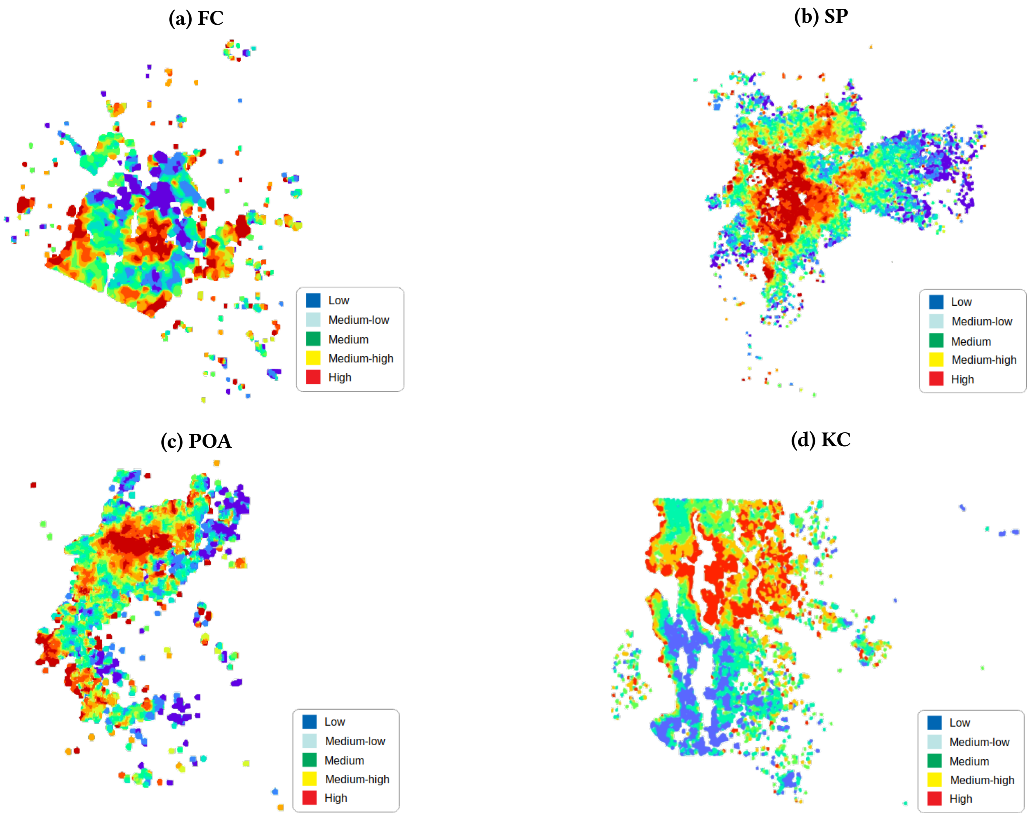

Figure 4: Spatial distribution of house prices on the 4 datasets.

in each dataset except for KC, wherein the model achieved a very poor performance (MALE=0.7). To handle that, we removed each individual feature from the original set and verified that the model for KC produced the best results when the feature "square foot lot" was removed. We also used the gstat package to run it and the Gaussian curve was the theoretical model for the semivariogram.

Due to the restrictions of these models, it was necessary to eliminate houses with the same latitude and longitude. To do so, we grouped each location and calculate the average of the remaining attributes for each dataset. Because of that, there was a reduction in the size of 3 datasets: SP (56,663 instances), POA (10,952) and KC (20.832), but not for FC. We normalized the numerical raw features using z-score normalization (mean=0 and standard deviation=1) and for the categorical raw features we apply one hot encoding [6].

Model Settings. We implemented our attention network using Keras [10]. To choose the best hyper-parameters, we used Hyperas, which is a wrapper around hyperopt [4]. The hyper-parameters, the values we used to search and the best values of the hyper-parameters for each dataset based on the validation set are presented in Table 2. For each dataset, we train the network for 300 epochs and selected the one that presented the best result in the validation data.

Table 2: Values of the hyper-parameters that we varied and their best numbers on the validation set.

		Best Value			
HP	Values	SP	POA	FC	KC
n-nearest houses	5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60	60	30	20	45
sigma(σ)	2, 5, 10, 15, 20	10	10	10	2
nodes layer	5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60	60	15	60	60
LR	[0.001-0.01]	0.001	0.008	0.001	0.001
batch size	250, 300, 400 500	250	250	250	250
act func hidden	Relu and ELU	ELU	ELU	ELU	ELU
act func regression	Relu, ELU and linear	linear	linear	linear	linear

Baselines. Most of previous approaches for house price estimation that applies deep learning does so by using some explicit spatial context such as house or satellite images, or points of interest. Since it is not straightforward to collect this information, we opted to compare our solution with previous ones that used the FC dataset for evaluation with the same training/test sets, without the need to actually implement them. The approaches we compare with are:

- [5]: this approach uses a random forest regressor for house price estimation. The model uses the structural features and features extracted from street-level images of the house using convolutional neural networks.
- [3]: this work proposes a convolutional neural network that produces a vector from features extracted from satellite images of the vicinity of the house and its structural features. This vector is then used by a regressor for house price estimation.

Error Measures. We evaluate the performance of the models by using 3 different error measures widely used in regression tasks: mean absolute log error (**MALE**), root mean square error (**RMSE**) and mean absolute percentage error (**MAPE**):

$$MALE = \frac{1}{N}\sum_{i=1}^{N} |log(y_i) - log(\hat{y}_i)| \tag{12}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{13}$$

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{14}$$

where y_i is the actual price of house i, $\hat{y}_i$ is the predicted value and N the number of houses in the training data. We opted to use those 3 measures because: RMSE was used to evaluate the previous approaches which we compare with in this section; MAPE gives a very intuitive idea regarding the percentage of the error; and MALE has been shown to be a robust measure to evaluate regression models [27].

Points of Interest. As aforementioned, our approach tries to capture the spatial context for house price prediction solely based on the houses in the neighborhood. To compare our solution with an approach that explicitly captures this context, we collected points of interest (POIs) of the houses present in our 4 datasets. They represent establishments such as shops, restaurants, hotels, parks, schools, government buildings, and so on. Table 3 shows details about this data. To collect the POIs, we use the place API of here.com[12]. For each location (latitude and longitude), we retrieved the 50 nearest POIs within 2 km radius. Following [3], we define each POI p containing latitude and longitude and a type of place Γ_p. The set of all places is denoted as Υ. In order to describe a house i in terms of its POIs, we generate the feature vector $P_{i,h}$, which represents the number of POI types within a distance h. Mathematically:

$$P_{i,h} = [P_{i,h,t}], \forall\ t \in \Upsilon, \quad P_{i,h,t} = \{|p|,\ d_{i,p} < h,\ t = \Gamma_p\} \tag{15}$$

[12]http://developer.here.com

where $d_{i,p}$ is the geodesic distance between house i and POI p and $|.|$ is the cardinality operation.

Table 3: Information about the POIs on the 4 datasets.

Characteristic	SP	POA	FC	KC
Num. POI	3,441,975	768,223	4,015,602	1,033,804
Num. Types	49	44	45	50

5.2 Results

First, we evaluate our model primarily as a feature generator for house price prediction by assessing the effectiveness of house embeddings created by the learned models. The results obtained by each regressor on the 4 datasets using the different feature sets are presented in Table 4. The lowest error value for each dataset is highlighted with a dot (•). In all 12 scenarios (4 datasets x 3 measure errors), a regressor using the house embedding obtained the lowest value of MALE, MAPE and RMSE, outperforming the regressors that use raw attributes of the house (HA), and the ones that try to capture spatial context implicitly by looking at the neighbors' attributes (HA+HC) or explicitly by using the POIs on the house's neighborhood (HA+POI). Regressors using only HE obtained the best results in 8 out of 12 of scenarios, whereas HE+POI achieved the lowest error in 5 of them (there was a tie between HE and HE+POI on the FC dataset on MALE). This indicates that the learned house embeddings (HE) in fact capture the implicit spatial context of houses to predict their prices and, in some cases, information from POIs can complement HE to achieve the best result.

Regarding the performance of individual models, Auto-sklearn using house embeddings (HE or HE+POI) had the best performance on most of the scenarios. For instance, on MALE, it obtained the lowest errors on all datasets: 0.134 for SP, 0.142 for POA, 0.097 for FC and 0.113 for KC. Recalling that Auto-sklearn builds an ensemble of models wherein each model has a weight, Table 5 shows that the model with the highest weight in the Auto-sklearn ensemble using HE, in the four datasets, was Linear Regression and/or Support Vector Regression using linear kernel. On the other hand, the ensembles created on raw features predominantly used tree-based ensemble models (Random Forest and Gradient Boosting). With respect to our trained attention network as a regressor, it achieved the lowest values on 3 scenarios: on the KC dataset on the measures RMSE (115763) and MALE (0.113); and on the POA on MAPE (9.58).

Another interesting observation is that although Linear Regression using house embeddding has only obtained the best results on few scenarios, its numbers show that it was very competitive with the other approaches. In terms of RMSE, for instance, it achieved the best result on SP, and POA, and its performance in comparison to the best approaches on the other datasets were: 0.005% worse on FC and 0.004% worse on KC. Its results were, however, very poor using raw features: HA, HA+HC or HA+POI, as opposed to more competitive algorithms such as Random Forest, Xgboost and Lightgbm (see Table 4).

We can conclude from those results that the house embedding in fact can capture the complexities associated with the house's

Table 4: Values of MALE, RMSE and MAPE of the regressors using the 5 features sets. Values marked with star represent the lowest value on the dataset and values in bold the lowest value for each regressor.

		FC			KC			SP			POA		
Mod.	Feature	MALE	RMSE	MAPE	MALE	RMSE	MAPE	MALE	RMSE	MAPE	MALE	RMSE	MAPE
LR	HA	0.219	51177	15.09	0.192	209202	15.31	0.266	264690	22.54	0.261	153806	22.52
	HA+HC	0.154	26982	10.14	0.241	327883	18.04	0.187	201275	14.73	0.241	184296	18.50
	HA+POI	0.202	46473	14.26	0.179	210187	13.94	0.257	257596	21.82	0.239	144529	20.77
	HE+POI	**0.097•**	**22911**	6.38	**0.114**	**116271**	8.05	0.135	155039	9.93	0.144	94416	**10.01**
	HE	**0.097•**	22921	**6.36**	**0.114**	116448	**8.00**	**0.135**	**154964•**	**9.92**	**0.144**	**94201•**	10.08
RF	HA	0.111	26762	6.92	0.123	135268	8.50	0.140	158876	10.12	0.160	100752	11.41
	HA+HC	0.107	24972	7.18	0.132	164206	9.52	0.159	178738	12.33	0.171	107138	12.82
	HA+POI	0.108	26153	6.78	0.124	137725	8.50	0.151	167782	11.52	0.159	100292	11.56
	HE+POI	**0.098**	**23297**	**6.40**	0.119	117329	8.36	0.137	**156865**	10.07	**0.146**	**95421**	**9.98**
	HE	0.099	23395	6.48	**0.118**	**116200**	**8.34**	**0.137**	157288	**10.06**	0.147	95832	10.22
LG	HA	0.108	25069	7.13	**0.115**	124667	**8.03**	0.146	161485	11.19	0.256	101434	12.23
	HA+HC	0.102	23826	6.83	0.117	134942	8.42	0.148	166866	11.37	0.201	104005	12.51
	HA+POI	0.106	24743	7.00	0.115	122451	8.05	0.156	169593	12.48	0.151	97068	10.48
	HE+POI	**0.099**	**23049**	6.47	0.118	129498	8.39	0.136	156161	**9.96**	0.147	95825	10.38
	HE	**0.099**	23111	**6.47**	0.117	**120284**	8.18	**0.136**	**156074**	10.04	**0.147**	**95756**	**10.35**
XB	HA	0.107	24624	7.19	0.116	128289	**7.97**	0.140	159018	10.41	0.154	97256	11.06
	HA+HC	0.108	24827	7.34	0.124	136051	9.12	0.156	172180	12.30	0.175	107515	13.57
	HA+POI	0.104	24148	6.94	**0.115**	124274	8.22	0.158	172786	12.47	0.148	**95423**	**10.49**
	HE+POI	**0.100**	23361	6.64	0.123	123575	8.81	0.137	**156685**	10.19	**0.147**	95904	10.65
	HE	**0.100**	**23224**	**6.51**	0.122	**122493**	8.80	**0.137**	157288	**10.07**	0.148	95798	10.58
AS	HA	0.113	27166	7.08	0.115	127714	7.92	0.144	161359	10.45	0.169	105156	12.28
	HA+HC	0.108	25492	6.82	0.133	123361	7.92	0.165	184744	12.95	0.163	101972	**9.85**
	HA+POI	0.109	25798	7.01	0.120	137029	8.55	0.152	167919	11.51	0.161	102113	11.73
	HE+POI	**0.097•**	**22783•**	**6.29•**	0.113	116546	**7.83•**	0.135	**155115**	9.92	**0.142•**	94418	9.90
	HE	**0.097•**	22837	6.31	**0.113•**	**116109**	9.88	**0.134•**	166866	**9.79•**	0.143	**94311**	12.22
RL	HE	0.098	23177	6.49	0.113•	115763•	8.00	0.135	155585	9.80	0.143	94492	9.58•

Table 5: Weights of each individual regressor in the ensembles created by Auto-sklearn using different feature sets. The values are shown in percentages.

	KC		FC		SP		POA	
Model	HA	HE	HA	HE	HA	HE	HA	HE
Random Forest	58	2	40	-	38	2	24	1
G. Boosting	30	2	2	-	6	-	19	8
Ridge Regress.	6	16	-	4	-	-	-	-
Adaboost	4	-	-	-	-	-	-	-
SVR	2	36	-	2	-	2	-	36
LR	-	36	-	72	-	72	-	36
KNN	-	8	36	8	30	16	52	6
Decision Tree	-	-	4	-	-	2	8	4
Extra Tree	-	-	-	14	26	6	-	-

features and its implicit spatial context, represented by its neighbors, to perform house price prediction, allowing such simple and light models such as Linear Regression to obtain very competitive results. The same argument can be made regarding our attention model, since a simple neuron with a linear activation function (see Table 2) on top of the house embedding obtains good results, and the same for the Auto-sklearn HE: Linear Regression, as we pointed out before, is one of the the models with the highest weight on the ensemble using the HE feature set.

We can observe similar results when we consider the lowest error of each regressor individually regarding the feature sets (presented in bold in Table 4): 56% of the best results are achieved using only HE and 30.67% of them using HE + POI. This means that, in 86.67% of the cases, the use of HE was essential for individual regressors to achieve the best results. For instance, the lowest errors in all scenarios for Linear Regression and Random Forest were achieved with HE or HE+POI feature sets.

Impact of the Attention Mechanism. To verify the impact of the attention mechanism on the performance of the network, we executed ablation experiments removing each one of the attention layers: euclidean and geographical. Table 6 presents the results. The removal of the geo attention has a great impact on the performance of the network (see Tables 4). For the POA dataset, for instance, the MALE value increased 54%: from 0.143 (full attention network) to 0.221. For the euclidean attention, on the other hand, there was not a great increase (see Table 4) on MALE, RMSE and MAPE with its removal from the network. From these numbers, we can conclude that the price of the houses in the vicinity is more useful to house price prediction than houses with similar characteristics.

We also evaluate the impact of the learned weights on the attention layers. Table 6 shows that on both attention layers there is an increase on the MALE, RMSE and MAPE values for all datasets when the weights are not considered in those layers, meaning the

Table 6: Impact of the attention mechanisms in the performance of the network. The column "No Att" represents the model without the respective attention layer and "No weigh." means the learned weights in the respective attention layer were removed.

		Euclidean Att		Geo Att	
City	Error	No Att	No weig.	No Att	No weig.
SP	MALE	0.137	0.138	0.148	0.158
	RMSE	157488	157667	165135	171638
	MAPE	10.23	10.14	10.50	12.37
POA	MALE	0.147	0.158	0.221	0.214
	RMSE	97010	101459	129325	133436
	MAPE	10.27	11.73	19.88	16.73
FC	MALE	0.101	0.101	0.099	0.101
	RMSE	23894	23891	23194	24052
	MAPE	6.66	6.66	6.43	6.48
KC	MALE	0.113	0.117	0.120	0.117
	RMSE	125154	112444	120884	127971
	MAPE	8.09	8.49	9.48	8.45

learned weights in the attention layers contain very useful information for the prediction.

Comparison with baselines. In Table 7, we present a comparison on the Fayette County dataset between our approach and the two baselines aforementioned. We only provide RMSE values because this was the only error measure presented in their work. In addition to the baselines and our attention network, we present the RMSE for the Auto-sklearn HE, which is the regressor with the lowest error for the feature set HE. The numbers show that the approaches that use house embedding (Attention Network and Auto-sklearn HE) present a superior performance in comparison to the baselines. It does so by capturing the spatial context of a house relying only on the information of the houses in its vicinity, as opposed to the two baselines that use more costly strategies to capture the spatial context by extracting features from images related to the house and its neighborhood to make the prediction.

Comparison with Spatial Interpolators. We also compare the performance of our approach with conventional spatial interpolators, which, similar to our attention network, implicitly capture spatial dependence from neighboring points. Table 8 shows that in all scenarios, our geo-attention-based network outperforms them. In fact, UK only achieved better results than linear regression using HA (see Table 4), whereas IDW obtained the worst results on all scenarios mainly because it only takes into consideration the geographic information of the neighbors to make the prediction.

6 RELATED WORK

Many previous approaches have proposed solutions to the problem of house price prediction. In this section, we focus our discussion on the more recent techniques that use deep learning and traditional statistical-learning approaches to estimate the real estate price.

Table 7: RMSE of our solution versus previous approaches on the Fayette County dataset. Recalling that Auto-sklearn HE is Auto-sklearn using the house embedding as feature set.

Model	RMSE
[5]	28281
[3]	24439.64 ± 11.63
Attention Network	**23177**
Auto-sklearn HE	**22837**

Table 8: Results of the spatial interpolators.

	IDW			UK		
City	MALE	RMSE	MAPE	MALE	RMSE	MAPE
SP	0.407	374651	35.49	0.204	218808	16.90
POA	0.390	213127	33.22	0.222	133972	18.75
FC	0.288	61768	21.05	0.173	38958	11.65
KC	0.246	279202	18.76	0.140	135900	14.28

6.1 Deep Learning

Poursaeed et al. [22] propose a solution that analyzes internal and external images of a house to estimate its luxury level using a deep convolutional neural network. Based on this estimation along with the structural features of the house, they perform the price prediction. As opposed to our approach, both approaches do not take into consideration the price and characteristics of the houses in the neighborhood. Another image-based approach is proposed by Bency et al. [3]. They use a convolutional neural network to extract a representation from satellite images of the neighborhood of the house in addition to home characteristics to perform the prediction. To capture the spatial context of a property, You et al. [29] apply a Bidirectional Recurrent Neural Network (BRNN) [24] on a sequence of neighboring properties generated by a random walk algorithm to perform the prediction. They represent the houses in the network with the average vector of the images of the house represented by vectors created using a pre-trained image model. The main drawback of those image-based approaches is the cost of collecting and processing the images to build their models.

6.2 Traditional Statistical Learning

Liu et al. [16] propose the hierarchical spatial functional model (HSFM) using spatial spline regression [23] that decomposes the price of a house in its structural characteristics and the land advantage at its location in two different levels: global (region of interest) and local (sub-region or sub-community). They employ this hierarchical strategy to model irregularly shaped geographic regions. Similar to our approach they use spatial dependency in their model but its main limitation is the high cost of partitioning the sub-regions, mainly when using more extensive datasets. We were not able to compare their approach with ours because of missing details in their methodology. Another statistical approach [12] aims to estimate the investment value of estates (house price) and ranking all estates accordingly in rising and falling markets. To perform the prediction, they propose a geographic method called ClusRanking that leverages three levels of geographical: the characteristics of the neighborhood (individual dependency); features of nearby real estate (peer dependency); and the level of prosperity of the region measured in terms of business conducted in the surroundings of the property (zone dependency). For that, geographic features are extracted from the data; the level of neighborhood popularity is estimated through the analysis of taxi's trajectory data; and how much a property is influenced by the level of prosperity of the region by its proximity to the center of a given region. Finally, a linear model is used to merge the three types of dependency factors and then perform the prediction of the value of the property. This work is related to ours, given that both capture the spatial dependence found between real estate prices, but we assume that spatial autocorrelation can be captured implicitly, as opposed to them that need external data for that.

7 CONCLUSION

In this paper, we presented a novel spatial interpolation approach based on attention networks applied to the problem of house price prediction. To model the features of a house in its neighborhood, we implemented two attention layers. The first one gives weights to neighboring houses based on their similarity to the property one wants to predict (euclidean attention), and the second one weights houses based on their geographic distance to the property (geo attention). This network learns a vector representation (HE) of the house that embeds the house's attributes and its spatial context into a common sub-space.

We performed an extensive evaluation on 4 real-world datasets that shows that in fact regressors using the HE outperforms models using raw and POI features, and even simple models as a linear regression has competitive performance to other modern and more sophisticated approaches as Xgboost and Lightgbm. In addition, our proposed solution outperformed previous deep-learning house price prediction approaches that capture the local context extracting features from images, which is very costly, as opposed to our approach that only relies on information of neighboring house to perform this task. Finally, our proposed solution obtain better results when compared to models that are historically used to model spatial autocorrelation phenomenon. As future work, we plan to extend this approach to problems of spatial interpolation in general, and apply the learned embeddings to other tasks such as product recommendation and entity resolution.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

[2] Gleb Beliakov and Gang Li. 2012. Improving the speed and stability of the k-nearest neighbors method. *Pattern Recognition Letters* 33, 10 (2012), 1296–1301.

[3] A. J. Bency, S. Rallapalli, R. K. Ganti, M. Srivatsa, and B. S. Manjunath. 2017. Beyond Spatial Auto-Regressive Models: Predicting Housing Prices with Satellite Imagery. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 320–329. https://doi.org/10.1109/WACV.2017.42

[4] James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David Cox. 2015. Hyperopt: A Python library for model selection and hyperparameter optimization. *Computational Science Discovery* 8 (07 2015), 014008.

[5] Z. Bessinger and N. Jacobs. 2016. Quantifying curb appeal. In *2016 IEEE International Conference on Image Processing (ICIP)*. 4388–4392.

[6] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

[7] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (Oct. 2001), 5–32.

[8] Martin D Buhmann. 2003. *Radial basis functions: theory and implementations*. Vol. 12. Cambridge university press.

[9] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. ACM, New York, NY, USA, 785–794.

[10] François Chollet et al. 2015. Keras. https://keras.io.

[11] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and Robust Automated Machine Learning. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 2962–2970.

[12] Yanjie Fu, Hui Xiong, Yong Ge, Zijun Yao, Yu Zheng, and Zhi-Hua Zhou. 2014. Exploiting Geographic Dependencies for Real Estate Appraisal: A Mutual Perspective of Ranking and Clustering. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, New York, USA) *(KDD '14)*. ACM, New York, NY, USA, 1047–1056. https://doi.org/10.1145/2623330.2623675

[13] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *NIPS*.

[14] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (12 2014).

[15] Stephen Law, Brooks Paige, and Chris Russell. 2018. Take a look around: using street view and satellite images to estimate house prices. *arXiv preprint arXiv:1807.07155* (2018).

[16] B. Liu, B. Mavrin, D. Niu, and L. Kong. 2016. House Price Modeling over Heterogeneous Regions with Hierarchical Spatial Functional Analysis. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. 1047–1052.

[17] G. Matheron. 1969. *Le krigeage universel (Universal Kriging)*. Vol. 1. Ecole des Mines de Paris, Fontainebleau, 83 p.

[18] Gang Mei, Nengxiong Xu, and Liangliang Xu. 2016. Improving GPU-accelerated adaptive IDW interpolation algorithm using fast kNN search. *SpringerPlus* 5, 1 (2016), 1–22.

[19] Lubos Mitas and Helena Mitasova. 1999. Spatial interpolation. *Geographical information systems: principles, techniques, management and applications* 1, 2 (1999).

[20] Edzer J. Pebesma. 2004. Multivariable geostatistics in S: the gstat package. *Computers Geosciences* 30 (2004), 683–691.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[22] Omid Poursaeed, Tomáš Matera, and Serge Belongie. 2018. Vision-based real estate price estimation. *Machine Vision and Applications* 29, 4 (01 May 2018), 667–676.

[23] Laura M. Sangalli, James O. Ramsay, and Timothy O. Ramsay. 2013. Spatial spline regression models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 75, 4 (2013), 681–703. https://doi.org/10.1111/rssb.12009 arXiv:https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/rssb.12009

[24] M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (Nov 1997), 2673–2681.

[25] Donald Shepard. 1968. A Two-Dimensional Interpolation Function for Irregularly-Spaced Data. *ACM National Conference* 23 (01 1968), 517–524.

[26] Michael L Stein. 2012. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media.

[27] Chris Tofallis. 2015. A better measure of relative prediction accuracy for model selection and model estimation. *Journal of the Operational Research Society* 66, 8 (2015), 1352–1362.

[28] R Webster and M Oliver. 2007. Geostatistics for Environmental Scientists: Second Edition. *Geostatistics for Environmental Scientists: Second Edition* (01 2007).

[29] Q. You, R. Pang, L. Cao, and J. Luo. 2017. Image-Based Appraisal of Real Estate Properties. *IEEE Transactions on Multimedia* 19, 12 (12 2017), 2751–2759.