A Comparison of Residential Apartment Rent Price Predictions Using a Large Data Set: Kriging Versus Deep Neural Network

Seya, Hajime

Shiroi, Daiki

A comparison of residential apartment rent price predictions using a large data set: Kriging versus deep neural network

Hajime Seya
Departments of Civil Engineering, Graduate School of Engineering Faculty of Engineering, Kobe University, 1-1, Rokkodai, Nada, Kobe, Hyogo, 657-8501, Japan,
E-mail: hseya@people.kobe-u.ac.jp; Tel.: +81-78-803-6278

Daiki Shiroi
Gunosy Inc., Roppongi Hills Mori Tower, 6-10-1, Roppongi, Minato, Tokyo, E-mail: daikishiroi@gmail.com

Abstract

Despite several attempts to compare and examine the predictive accuracy of real estate sales and rent prices between the regression-based and neural-network (NN)-based approaches, the results are largely mixed. Prior study limitations include small sample size and a disregard for spatial dependence, which is an essential characteristic of real estate properties. Hence, this study aims to add new empirical evidence to the literature on comparing regression-based with NN-based rent price prediction models through sophistications by (1) examining different and relatively large-scale sample sizes ($n = 10^4$, 10^5, 10^6), and (2) considering the spatial dependence of either the application of nearest neighbor Gaussian processes (NNGP) or the latitude-longitude coordinate function (in the case of a deep neural network [DNN]). A case study of apartment rent prices in Japan shows that, given an increase in sample size, the out-of-sample predictive accuracies of the DNN approaches and that of NNGP are nearly equal in the order of $n = 10^6$. However, the DNN may have higher predictive accuracy than the NNGP for both higher- and lower-end properties whose rent prices deviate from the median.

Key Word: Apartment rent price prediction; Nearest neighbor Gaussian processes (NNGP); Deep neural network (DNN)

1. Introduction

The real estate industry has been observed to lag behind other businesses in terms of digitalization. In recent years, however, efforts to streamline operations and price assessments using new technologies have gained momentum under the name of *ReTech* (Real Estate Tech). Regarding price assessments or predictions, several online automated services such as Zestimate[1] (a service provided by the Zillow Group in the U.S.) or Price Map[2] (a service provided by LIFULL Co., Ltd. in Japan) have gained popularity. Price Map, for instance, represents properties on a map one can use to review reference sales and monthly rent prices by entering information such as room layout and room size. Other similar services also exist. They are typically supported by the vast accumulation of property data and statistics- or machine learning (ML)-based sales and rent price prediction algorithms. These algorithms are progressing rapidly and they affect the real estate industry significantly.

Both real estate agencies and end users require automated price assessment and prediction.[3] From the agency perspective, there is merit in reducing appraisal costs and improving transparency. For example, in Japan, licensed real estate appraisers provide property assessments based on the expected cash flow and by comparison with similar properties. However, since all real estate properties are unique, that is, there are no identical properties, and an appraisal must be conducted considering the supply-demand balance, in addition to the property characteristics themselves, explaining the basis for the appraisal to end users requires a tremendous effort. From the end-user perspective, aside from appraisal costs, there is the issue of information asymmetry between real estate agencies and end users, as observed in the limited disclosure of purchase prices due to privacy policy.[4] This issue implies that if real estate sales or rent prices can be reasonably and quickly predicted, then the information asymmetry between end users and real estate agencies would, at least partially, disappear, thus leading to the revitalization of the real estate market.

While the automated assessment of real estate sales and rent prices using big data and ML approaches in ReTech has garnered attention (Abidoye and Chan 2017; Čeh et al. 2018), regression-based approaches have also been widely used for predicting real estate sales and rent prices. Efron (2020) provided a comparison checklist of the differences between regression-based and ML-based approaches. He suggested that most traditional regression methods depend on some sort of *surface plus noise* formulation, where the surface describes the scientific truths (long-term) we wish to learn (estimate), while ML-based approaches typically focus on empirical predictive accuracy (possibly short-term). He also mentioned that we usually use regression-based approaches not only for prediction but also for *attribution*, that is, the assignment of significance to individual predictors (i.e., significance testing). The tactic of combining weak learners in ML-based approaches (e.g., random forests and XGBoost) is not available for attribution and, therefore, prediction is much easier than attribution.

The economic theory behind attribution is known as the hedonic approach (Rosen 1974; Taylor 2008), under which the marginal benefits of attributes can be evaluated using estimated regression coefficients. Despite these merits, the pure prediction perspective typically adopts simple functional forms, such as linear or logarithmic, which may be insufficient to fully capture nonlinearity in data.[5] Hence, it is important to test the extent to which its predictive accuracy differs from that of the ML-based approach. In relation to the ML-based approach, we focus on a deep neural network (DNN) model. This is because, compared to other ML-based methods centered on trees and forests (Pace and Hayunga 2020), little is known about its predictive accuracy in terms of rent price prediction. Since a DNN is expected to perform well when sample size is large, it is important to assess its performance with a large data set. As will be discussed in the next section, existing studies have limitations in this regard.

Further, in real estate appraisal or prediction, some factors are challenging to accommodate as explanatory variables, such as neighborhood quality (Dubin 1988). It is, therefore, important to consider how these unobserved or omitted factors can be incorporated into the model (Von Graevenitz and Panduro 2015). In geo- (spatial) statistics, regression-based kriging was established as a method of handling these variables as the spatial dependence among errors, typically with assuming a Gaussian process (GP) to the error term (e.g., Dubin 1988; Cressie and Wikle 2011). Studies such as those of James et al. (2005), Bourassa, Cantoni, and Hoesli (2010), and Seya et al. (2011) have reported that kriging provides high predictive accuracy compared to simple multiple regression models (hereinafter, ordinary least squares [OLS]) in the property field. Since the model structure of OLS is fairly simple, parameters can be determined with relatively small samples. With kriging, however, since the price information of neighboring properties is reflected in the predicted results through spatial dependence, the situation is different from that of OLS. That is, there are many benefits of increasing sample size for prediction. This study compares and discusses the results of rent price predictions using two different approaches (regression- and ML-based) for various sample sizes.

Regarding the former, we focus on kriging. However, we also employ OLS as a popular benchmark. As the sample size increases (e.g., when $n = 10^5$), it is increasingly challenging to directly apply kriging, which requires the computational cost of O (n^3) for inverting the variance-covariance matrix. Hence, we use a nearest neighbor Gaussian process (NNGP) model, which allows for the application of kriging to big data via sparse approximation (Datta et al. 2016; Finley et al. 2017; Zhang, Datta, and Banerjee 2019). While there are various approaches for spatial statistical modeling using big data (Yamagata and Seya 2019), NNGP performs consistently well in a comparative study (Heaton et al. 2019). Regarding the latter, we employ a DNN. A DNN can construct complicated non-linear functions and consider spatial dependence through a non-linear function for position coordinates without explicitly modeling the spatial dependence.

Despite several attempts to compare and examine the predictive accuracy of real estate sales and rent prices between the regression- and neural-network (NN)-based approaches, the results are largely mixed. Prior study limitations include [1] a small sample size (except for Zurada, Levitan, and Huan [2011]), [2] disregard for spatial dependence (except for Georgiadis [2018]), and [3] tailored and ad-hoc settings of the hyperparameters in DNN (or artificial neural networks [ANN]). Hence, this study [1] examines different and relatively large-scale sample sizes ($n = 10^4$, 10^5, 10^6), [2] considers the spatial dependence of either the application of NNGP (kriging) or the latitude-longitude coordinate function (in the case of DNN), and [3] optimizes the hyperparameters in DNN. It also provides a new empirical evidence base to the literature on comparing regression- and NN-based rent price prediction models through these sophistications.

For empirical validation, this study employs LIFULL HOME's data set[6] for monthly residential apartment rent prices in Japan.[7] Since there is generally less research on rental prices than on sales prices, it may be a valuable data set,[8] as it comprises snapshots (cross-section data) of rental property data or image data as of September 2015. The former shows rent, lot size, location (municipality, zip code, nearest station, and walk time to nearest station), year built, room layout, building structure, and equipment for 5.33 million properties throughout Japan, whereas the latter comprises 83 million pictures that show the floor plans and interior details for each property. This study employs only the former data. Nevertheless, the latter information may be useful to improve prediction accuracy.

Out of the approximately 5.33 million properties, 4,588,632 properties were retained after excluding missing data, from which $n = 10^4$, 10^5, and 10^6 properties were randomly sampled. While focusing on the difference in sample size, the accuracies of the out-of-sample prediction for property rent prices based on the OLS, NNGP, and DNN approaches were compared via a validation process. Our analysis showed that, with an increase in sample size, the predictive accuracy of DNN was observed to approach that of NNGP. Moreover, in the order

of $n = 10^6$, they were nearly equal. During this experiment, the conventional explanatory variables that had been incorporated into the regression-based hedonic model were used. Our findings suggest that, given these standard settings, even if the sample size is very large (in the order of $n = 10^6$), the use of regression-based NNGP may be a promising option.

The rest of the article is structured as follows. Section 2 reviews the relevant literature. Section 3 briefly explains the models used in the comparison study. Section 4 presents the results of the comparative analysis using the LIFULL HOME's data set. Section 5 presents the conclusion and the scope for future research.

2. Literature review

This section reviews the existing literature on the prediction of real estate sales and rent prices. Existing studies posit a high predictive accuracy for spatial regression models compared to the OLS model (e.g., James et al., 2005). Seya et al. (2011) examined the performance of various spatial prediction models that consider spatial dependence. To this end, they employed a data set of apartment rents in Tokyo's 23 wards for empirical comparison and showed the usefulness of considering spatial dependence in the error term (kriging, geoadditive model, spatial error model) or regression coefficients (geographically weighted regression [GWR] model). However, their sample size was fairly small (i.e., 529 for parameter estimation and 150 for validation).

Geostatistical models (known as kriging) and spatial econometric models are widely used for considering spatial dependence among errors.[9] A considerable number of studies have applied both methods for hedonic price modeling (e.g., Pace, Barry, and Sirmans 1998; James et al. 2005). However, for spatial (i.e., out-of-sample) prediction purposes, although the latter method can be used (Kelejian and Prucha 2007), the former, which does not utilize a spatial weight matrix, is more natural and flexible (Tsutsumi and Seya 2009). Note that, according to the comparison made by Seya et al. (2011), the differences in the predictive accuracy of kriging and of the spatial econometric model were negligible compared to the differences between those of OLS and kriging. In the case of kriging, an application to big data in the order of a million is possible through various approximations (Heaton et al. 2019).

For the modeling of spatial dependence among regression coefficients, various methods have been developed in different fields, including geography, statistics, and ML (e.g., Brunsdon, Fotheringham, and Charlton 1998; Fotheringham, Brunsdon, and Charlton 2002; Gelfand et al. 2003; Murakami et al. 2017; Inoue, Ishiyama, and Sugiura 2020). Since the housing market is often segmented, applying the local model (i.e., spatially varying coefficient [SVC] model) is a reasonable option. Hence, many studies have employed SVC models for hedonic price modeling (e.g., Carruthers and Clark 2010; Huang, Wu, and Barry 2010).[10] However, the application of SVC models to big data is still in the early stages (Bussas et al. 2017; Li and Fotheringham 2020; Murakami et al. 2020). For instance, we applied the *scalable* GWR model, proposed by Murakami et al. (2020), to our data set. Unfortunately, the parameter estimation procedure (i.e., bandwidth selection procedure) could not complete its processing within 24 hours when $n = 10^5$. Thus, currently, it is not easy to apply the model to a data set with a sample size of $n = 10^6$ and above.

Considering the regression-based and NN-based approaches in terms of prediction of real estate sales and rent prices, Kontrimas and Verikas (2011) employed data on home sale transactions to compare the predictive accuracy of the ML approach, including multi-layer perceptron (MLP), which is a subset of DNN, and OLS. They found that the mean absolute percentage differences for MLP and OLS were 23% and 15%, respectively, and OLS outperformed MLP. However, the sample size of their study was no greater than 100.

Similarly, Georgiadis (2018) compared the predictive accuracies of regression-based models and ANN on sales prices of 752 apartments in Thessaloniki, Greece, and found that the GWR model outperformed ANN. While these two studies have shown that the regression-based approach outperformed the NN-based approach in terms of predictive accuracy, the sample sizes used for these studies were merely in the order of $n = 10^2$. Abidoye and Chan (2018) compared ANN with OLS using sales transaction data for 321 residential properties in Lagos, Nigeria. They concluded that ANN outperformed OLS. Similarly, Yalpır (2018) and Selim (2009) compared ANN with OLS and suggested that the former performed better. Yalpır (2018) used 98 observations, whereas Selim (2009) used a relatively larger sample (5,741). Yalpır (2018) used three activation functions (the sigmoid, tangent hyperbolic, and adaptive activation functions) to build an ANN. However, hyperparameters other than activation functions were fixed in the validation process.

According to Valier (2020), there were 57 cases in which ML-based models, including NN, were more accurate in predicting value, as compared to 13 cases in which regression performed better. Zurada, Levitan, and Guan (2011) suggested that, although there have been many studies in the recent years that compare regression with AI-based methods in a mass appraisal context, a meaningful comparison of published results is challenging. A reason they suggested is that, in many studies, the models have been built on relatively small samples. They, thereby, employed a large data set containing over 16,000 transactions of sales to conduct a more comprehensive comparative study, and found that non-traditional regression-based methods performed better in all simulation scenarios, especially with homogeneous data sets. AI-based methods performed well with less homogeneous data sets under some simulation scenarios.

3. Models

This section introduces the NNGP and DNN models employed in our empirical comparison.

3.1. NNGP

Let D be the spatial domain under study and $\boldsymbol{s}$ be a coordinate position, (x, y). The spatial regression model, often termed as the spatial process model, can then be expressed as follows (Banerjee, Carlin, and Gelfand, 2014; Yamagata and Seya, 2019):

$$y(\boldsymbol{s}) = m(\boldsymbol{s}) + w(\boldsymbol{s}) + \varepsilon(\boldsymbol{s}), \qquad \varepsilon(\boldsymbol{s}) \sim N(0, \tau^2), \tag{1}$$

where the spatial process on real estate rental prices, $y(\boldsymbol{s})$, is decomposed to $m(\boldsymbol{s}), w(\boldsymbol{s})$, and $\varepsilon(\boldsymbol{s})$. τ^2 is a variance parameter, which is also termed as a nugget, which represents micro-scale variation and measurement error. Normally, we assume that $m(\boldsymbol{s}) = \boldsymbol{x}(\boldsymbol{s})'\boldsymbol{\beta}$, where $\boldsymbol{x}$ is an explanatory variable vector at point $\boldsymbol{s}$ and $\boldsymbol{\beta}$ is the corresponding regression coefficient vector. Further, $w(\boldsymbol{s})$ is assumed to follow the GP: $w(\boldsymbol{s}) \sim GP(0, C(\cdot, \cdot | \boldsymbol{\theta}))$, where the mean is zero and the covariance function is $C(\cdot, \cdot | \boldsymbol{\theta})$ ($\boldsymbol{\theta}$ is a parameter vector that typically includes the parameter ϕ [where $1/\phi$ is called the range], which controls the range of spatial dependence, and variance parameter σ^2 represents the variance of the spatial process and is termed as the partial sill). Finally, $\varepsilon(\boldsymbol{s})$ is an uncorrelated pure error term.

Let the sample be obtained at point $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n$, and let $y(\boldsymbol{s}_i)$ and $\boldsymbol{x}(\boldsymbol{s}_i)$ denote the dependent and explanatory variables observed at location $\boldsymbol{s}_i$. Thus, $\boldsymbol{w} = \big(w(\boldsymbol{s}_1), w(\boldsymbol{s}_2), \ldots, w(\boldsymbol{s}_n)\big)'$ follows the multivariate Gaussian distribution: $\boldsymbol{w} \sim N(\boldsymbol{0}, \boldsymbol{C}(\boldsymbol{\theta}))$, where the element of

$n{\times}n$ matrix $C(\theta)$ is given by $C(s_i, s_j|\theta)$ $(i = 1, ..., n; j = 1, ..., n)$. The spatial process model can then be expressed as $y{\sim}N$ $(X\beta,$ $\Lambda(\tau^2, \theta))$, where $\Lambda(\tau^2, \theta) = C(\theta) + \tau^2 I$, where I is an $n{\times}n$ identity matrix.

The prediction of response $y(s_0)$ at any given point s_0 is termed as kriging.[11] The kriging predictor requires the inverse of $n{\times}n$ variance-covariance matrix Λ. Thus, the computation requires the cost of $O(n^3)$. This implies that the computation is challenging in the order of $n = 10^5$. Hence, various approaches have been proposed to approximate the spatial process $w(s)$ (Heaton et al., 2019). Among the alternatives, this study employs the NNGP model based on Vecchia (1988). The joint density of the spatial process w (full GP), is expressed as the product of conditional densities, that is, $p(w) = p(w(s_1)) \prod_{i=2}^{n} p(w(s_i)|w(s_1), ..., w(s_{i-1}))$.[12] Vecchia (1988) then assumes the following approximation to this joint density:

$$\tilde{p}(w) = p(w(s_1)) \prod_{i=2}^{n} p(w(s_i)|w(N(s_i))). \tag{2}$$

Here, $N(s_i)$ is a neighbor set of s_i and is given as the k-nearest neighbors of s_i in NNGP. Thus, NNGP approximates the full GP expressed as a joint density using the nearest neighbors. Datta et al. (2016) demonstrated that the approximation of Equation (2) leads to an approximation of precision matrix C^{-1} to $\widetilde{C}^{-1}$ provided in the following equation:

$$\widetilde{C}^{-1} = (I - A)'D^{-1}(I - A), \tag{3}$$

where A is a sparse and strictly lower triangular matrix, with its diagonal given by zero, and has non-zero entries at most k-entries in each row. $D = \text{diag}(d_{ii})$ is a diagonal matrix whose elements are conditional variances based on the full GP model. Here, because A and D can be provided as $k{\times}k$ $(k << n)$ matrices and $\widetilde{C}^{-1}$ is sparse, the computational load can be significantly reduced. The spatial process model provided through NNGP may be expressed as follows:

$$y{\sim}N(X\beta, \ \widetilde{\Lambda}(\tau^2, \theta)), \tag{4}$$

where $\widetilde{\Lambda}(\tau^2, \theta) = \widetilde{C}(\theta) + \tau^2 I$.

The parameters in the NNGP model can be estimated using (Bayesian) Markov chain Monte Carlo (MCMC) (Datta et al. 2016), Hamiltonian Monte Carlo (Wang et al., 2018), and maximum likelihood methods (Saha and Datta 2018). This study employs the MCMC. Since the NNGP parameters are β, and $\phi = (\tau^2, \sigma^2, \phi)' = (\tau^2, \theta)'$, when using MCMC, we must set a prior distribution for each parameter and multiply it by the likelihood function to obtain the conditional posterior distributions (full Bayesian NNGP). Since this study employs massive data to a maximum of $n = 10^6$, it is challenging to implement the full Bayesian NNGP within a practical computational time. Accordingly, the study employs the conjugate NNGP, as proposed by Finley et al. (2017). Suppose $\widetilde{P}(\phi)$ is the approximate nearest neighbor of a spatial correlation matrix that corresponds to an approximate nearest neighbor of $\widetilde{C}(\theta)$. The conjugate NNGP can then be provided as follows:

$$y{\sim}N(X\beta, \ \sigma^2 \widetilde{M}), \tag{5}$$

where $\widetilde{M} = \widetilde{P}(\phi) + \alpha I$ and $\alpha = \tau^2/\sigma^2$. The conjugate NNGP is employed because, when assuming that α and ϕ are known, the conjugate normal-inverse Gamma posterior distribution for β and σ^2 can be used, and the predictive distribution for $y(s_0)$ can also be obtained as a t-distribution. Thus, it is straightforward to perform MCMC sampling. The empirical section explains how we set the

values of α and ϕ.

3.2. DNN

DNN is a mathematical model with a network structure in which layered units are connected with neighboring layers. It thus allows for the construction of extremely complicated non-linear functions. Each element that comprises a network is termed a unit or node. The first layer is termed the input layer and the last, the output layer; all the other layers are referred to as hidden layers. Next, the index for layers is expressed as $l = 1, \ldots, L$, where the first layer is the input layer and the Lth layer is the output layer. In a DNN, the results of non-linear transformations on inputs received from the previous layer are transmitted to the next layer to derive outputs at the output layer as an estimation result. Thus, for each observation, linear transformations via a weight matrix $\boldsymbol{W}_{l+1}$ $(m_l \times m_{(l+1)})$ and non-linear transformations via an activation function $f(.)$ are conducted in each layer. The transformation from the lth layer output z_l $(m_l \times 1)$ to the $l+1$th layer output z_{l+1} $(m_{(l+1)} \times 1)$ can be performed according to the following equations:

$$\boldsymbol{u}_{l+1} = \boldsymbol{W}_{l+1}\, \boldsymbol{z}_l + \boldsymbol{b}_{l+1}, \tag{6}$$

$$\boldsymbol{z}_{l+1} = \boldsymbol{f}(\boldsymbol{u}_{l+1}), \tag{7}$$

where $\boldsymbol{b}_{l+1}$ is the $m_l \times 1$ bias vector and $\boldsymbol{f}(\boldsymbol{u}_{l+1})$ is the activation function vector. The final output is given as $(z_L \equiv \hat{y})$. In determining $\boldsymbol{W}_{l+1}$ and $\boldsymbol{b}_{l+1}$ for regression (where y is continuous), the following mean squared error (MSE) of the actual value y and the predictive value $\hat{y}$ is often used as a loss function h.

$$h = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 . \tag{8}$$

The process of searching $\boldsymbol{W}_{l+1}$ and $\boldsymbol{b}_{l+1}$ that minimize h is termed as DNN learning. Learning is performed by the gradient algorithm, while backpropagation is used to calculate the gradient. In contrast to the case of usual statistical estimation, partial derivatives are computed from the output layer (LeCun, Bengio, and Hinton 2015).

Although the structure of the standard DNN model is simple, there are many hyperparameters we must calibrate, including the number of layers, number of units in the hidden layers, learning rate, and batch size. Additionally, the DNN parameter space has a tree structure, which means we must be alert to the presence of conditional parameters. For example, the number of units in each layer cannot be determined until the number of layers is determined. The presence of these hyperparameters is undoubtedly a source of the plasticity and high predictive accuracy of a DNN. Conversely, there is no denying that the difficulty (and personalization) of the setting is an obstacle for applied researchers and practitioners. The grid and random searches are widely used methods for DNN parameter tuning (Bergstra and Bengio 2012). In the empirical section, we report our strategies of hyperparameter tuning to obtain a well-tuned DNN model.

4. Empirical comparison

4.1. Data set

This study employed LIFULL HOME's data set for rent price predictions. Out of approximately 5.33 million properties, 4,588,632 properties (after excluding missing data) were used as original data. Although the original data did not explicitly contain property positional coordinates $\boldsymbol{s}$, they contained zip codes. Hence, the barycentric coordinates for zip codes (X, Y coordinates projected

to the UTM54N WGS84 reference system) were used instead. When multiple properties share the same location (e.g., a different room in the same apartment), small perturbations (random noise) are given to each positional (X, Y) coordinate within the zip code. This process may bring about some positional errors, but, given that our study is nationwide in scope, these errors are negligible.

The dependent variable is the natural logarithm of the monthly rent price (including maintenance fees) [yen];[13] the explanatory variables shown in Table 1 were used. We selected standard variables to include descriptors of the location of the condominium (location variables) and descriptors of the condominium itself (structural variables). For location variables, we employ "Walk time to nearest (train) station" [m]; "Floor-area ratio" [%]; and "Use district" [dummies]. For structural variables, we employ "Years built" [month]; Number of rooms [#]; "Direction" [dummies]; "Building structure" [dummies], and "Room layout" [dummies]. The number of explanatory variables (K) is 43. Table 1 presents the descriptive statistics.[14] Fig. 1 shows the natural logarithm of the rent price (yen) for each prefecture.

[Table 1-1 Descriptive statistics (continuous variables)] near here

[Table 1-2 List of explanatory variables (discrete variables)] near here

[Table 1-3 Descriptive statistics (discrete variables)] near here

[Figure 1. Log (rent price) for each prefecture.] near here

4.2. Experimental design

We compared the rent prediction accuracy of the NNGP with that of the DNN. We also show the OLS results as a benchmark. Regarding the prediction for the 4,588,632 properties, they were randomly selected at various sizes ($n = 10^4$, 10^5, 10^6), and 80% of these data were used as training data for the learning models. Moreover, the remaining 20% were used as testing data (validation data) to test the prediction accuracy. The sample size for the training and testing data had three patterns: (8000 vs. 2000), (80,000 vs. 20,000), and (800,000 vs. 200,000). Since sampling was completely random, there were no containment relations where, for instance, 10^4 samples are contained in 10^5 samples. However, since the data size was sufficiently large, it would be highly unlikely that the sample bias would conceal trends. Thus, this study design (based not on conditionalization but complete random sampling) would not significantly affect the results. The OLS and the NNGP are estimated using R, whereas the DNN is estimated using Python. To use the same random seed for R and Python, we used a reticulate package that provides an R interface to Python modules, classes, and functions.

For predictive accuracy assessment, the following error measures were used: mean absolute error [MAE], root mean squared error [RMSE], and mean absolute percentage error [MAPE]. Here, $\hat{y}_m$ and y_m are the out-of-sample predictive and observed values, respectively, for the m th data. Note that the first two measures are not robust to outliers for skewed distributions because the noise is unlikely to be Gaussian with constant variance. If we calculate RMSE on a skewed response variable, the resulting statistic is going to be mainly driven by the observations of the highest magnitude (see descriptive statistics). Hence, we calculate all error measures while keeping y_m log-transformed. Note, however, that MAPE for log-transformed variables does not have means that are interpretable as percentages (see Swandon, Tayman, and Barr 2000). Hence, we also calculated MAPE for real-scale.

$$MAE = \frac{1}{M} \sum_{m=1}^{M} |y_m - \hat{y}_m|, \tag{9}$$

$$RMSE = \sqrt{\frac{1}{M} \sum_{m=1}^{M} (y_m - \hat{y}_m)^2}, \tag{10}$$

$$MAPE = \frac{100}{M} \sum_{m=1}^{M} \left| \frac{y_m - \hat{y}_m}{y_m} \right|. \tag{11}$$

4.3. Model settings

4.3.1. OLS

Table 1 presents the explanatory variables, except for the X and Y coordinates. For reference, Table 2 presents the regression analysis results based on the OLS estimation when $n = 10^6$. The adjusted R^2 value is 0.5165.

[Table 2 Regression analysis results using OLS (example of $n = 10^6$)] near here

4.3.2. NNGP

We use the conjugate NNGP proposed by Finley et al. (2017) (Section 3.1). The conjugate NNGP is a pragmatic approach that accelerates sampling by assuming α and ϕ to be "known." The full Bayesian NNGP is theoretically sound. This study, however, employs massive data with up to $n = 10^6$ of data. Hence, it is practically challenging to implement a full Bayesian NNGP. In such cases, the conjugate NNGP is a beneficial alternative. Finley et al. (2017) proposed to assign values to α and ϕ via the grid point search algorithm based on the cross-validation (CV) score. However, the computational load remains high for performing a grid point search for $n = 10^6$ of data. Therefore, this study undertakes the following ad-hoc strategy in assigning values to α and ϕ.[15]

From the remaining data not used for comparison, 10,000 properties were randomly sampled, and parameters were defined by iteratively re-weighted generalized least squares (Schabenberger and Gotway 2005, 256–9) in semivariogram $\gamma(d) = C(0) - C(d)$, which is conversely related to the covariance function. Fig. 2 shows the fitting results of the empirical variogram, where distance d is divided into R units of sections $\hbar_r (r = 1, \ldots, R)$ without mutually overlapping ranges and determines the average value of non-similarity of each section to the theoretical functions by the Cressie–Hawkins's robust estimation (Cressie and Hawkins 1980). Starting from the left, the Gaussian, spherical, and exponential models are shown; the Gaussian model had the best CV score and, hence, it was used. We can see that the Gaussian model is a particularly good fit for near-distances that are subject to prediction results. Given these observations, the values for each parameter are as follows: $\phi=1/25.8$, $\tau^2 = 0.04$, and $\sigma^2 = 0.03$.

Next, the model parameters were used to develop an NNGP model. The spConjNNGP function in the spNNGP package of R was used for implementation. An NNGP model requires a determination of the number of nearest neighbors to consider. In the default setting of the spConjNNGP function, it is 15.[16] When the relationship between the number of nearest neighbors k and CV score (MSE) was plotted,[17] there was a tendency for the MSE to decrease to approximately $k = 30$ and increase thereafter (Fig. 3). Thus, the number of nearest neighbors was set as $k = 30$ in performing the validation.

[Figure 2. Fitting of variogram functions (Gaussian model; Spherical model; Exponential model).] near here

[Figure 3. Change in the MSE according to the number of nearest neighbors (in the case of $n=10^5$).] near here

4.3.3. DNN

This subsection explains the DNN settings. As noted in the DNN subsection of the methodology section, DNN has several hyperparameters to be determined. The grid and random searches are widely known as typical methods for DNN parameter tuning. However, they are inefficient. This study adopts a more efficient optimization technique, known as the tree-structured Parzen estimator (TPE) (Bergstra et al. 2011). It was adopted for its ability to adequately address the tree-structured parameter space of DNNs and its numerous records of adoption; moreover, its performance has been proven to some degree (Bergstra et al. 2011; Bergstra, Yamins, and Cox 2013). The parameter space (range of search) and obtained parameters were set, as shown in Tables 3 and 4.

Typical activation functions include the traditional sigmoid, hyperbolic tangent, softmax, and the recently popularized rectified linear unit (ReLU). ReLU has an advantage over the others in terms of computation because it induces sparsity in the hidden units (Glorot, Bordes, and Bengio 2011), as well as the non-saturation of its gradient, which accelerates convergence (Krizhevsky, Sutskever, and Hinton 2012). Hence, this study adopts ReLU. Regarding the optimizer for the DNN, since relatively large differences were found in the results according to the type of algorithm used, only the results using typical algorithms, RMSprop (Tieleman and Hinton 2012), and adaptive moment estimation (Adam) (Kingma and Ba 2014) are shown. This study did not use techniques designed to prevent overtraining, such as regularized terms and dropouts. It instead employed Keras[18] for the development of a DNN and Optuna[19] for TPE implementation with Python.

The learning procedure for a concrete model was as follows. First, based on the tth hyperparameter candidate vectors $\boldsymbol{\delta}_t$ and the results of applying five-fold CV with training data for each $\boldsymbol{\delta}_t$ (MSE, eq. [8]), a 50-fold search was performed using TPE. Second, a model was created once again using the optimal hyperparameter vector and all the training data to assess the predictive accuracy of the testing data. The explanatory variables used were standardized in advance. Table 4 shows the optimization results of the hyperparameters.

[Table 3 DNN hyperparameters and search range] near here

[Table 4 DNN hyperparameters after optimization (Adam)] near here

4.4. Results

The predictive accuracies by sample size for each model are shown in Figs. 4 (for log-scale) and 5 (MAPE for real-scale). These figures show that, for DNN, when the sample size is relatively small (i.e., $n = 10^4$), the RMSprop optimizer performed better; however, when the sample size is large (i.e., $n = 10^5$ and 10^6), the Adam optimizer performed better for log-scale. For real-scale, the differences in the performance are minor. The predictive accuracies of OLS did not show large differences, even if the sample size increased, because OLS, which does not use local spatial information, has a simple model structure such that $n = 10^4$ was sufficiently large for determining the parameters. In fact, in the case of real-scale, increased sample size worsened the predictive accuracy. This is caused by the increase in the number of high-priced properties in test data.

The NNGP showed the best results of all three models, for any sample size and any error measures. Even with a relatively smaller sample size ($n = 10^4$), it had a high accuracy (MAPE = 1.110 for log-scale and 12.63 for real-scale). At $n = 10^4$, DNN (Adam) had a larger error than that of OLS in terms of the MAE (OLS = 0.2113; DNN (Adam) = 0.2118). However, it had a larger margin of improvement in accuracy with an increase in sample size. Further, at $n = 10^6$, it reached the same level as that of the NNGP.

These results implied that the DNN could be useful, particularly in a context in which the sample size is large. That is, in a context in which the sample size is relatively small (i.e., $n = 10^4$), its predictive accuracy does not differ significantly from that of the OLS. The latter result would likely have led to the mixed results of prior studies, as discussed in Section 2. Fig. 6 shows the scatter plots depicting predicted and actual rent prices at $n = 10^6$ for log-scale.

From Fig. 7, we can see that, across all models, predictive accuracy is poor in areas where the rent price is high. The MAPE per logarithmic rent price range for each model is shown in Fig. 7 for a closer evaluation. The comparison between NNGP and DNN shows that the DNN was more accurate in the high-rent areas with a logarithmic rent price of 12 or greater and low-rent areas with a logarithmic rent price of 10.5 or less. By contrast, NNGP performed better in the median-rent areas with a logarithmic rent price of 11 to 12, where the number of cases is large. The number after the log rent range in Fig. 7 denotes the number of cases.

These results suggest that, regarding rent price prediction models using standard explanatory variables, if the sample size is moderate ($n = 10^4$, 10^5), kriging (NNGP) is useful, whereas the DNN may be promising if a sufficient sample size is secured ($n = 10^6$). The DNN is especially useful for the prediction of apartment rent prices in higher-end markets.

[Figure 4. Prediction results by sample size for each model (log-scale).] near here

[Figure 5. Prediction results by sample size for each model (real-scale).] near here

[Figure 6. Scatter plot of predicted (horizontal axis) and actual (vertical axis) rent prices for each model (in the case of $n = 10^6$).] near here

[Figure 7. MAPE per log rent (in the case of $n = 10^6$).] near here

5. Concluding remarks

As mentioned in the introduction, there is a need for an accurate prediction model of real estate sales and rent prices for businesses and end users. Despite several attempts to compare and examine the predictive accuracy of real estate sales and rent prices between the regression- and NN-based approaches, the results are largely mixed. Prior study limitations include a small sample size and the disregard for spatial dependence, which is an essential characteristic of real estate properties. Hence, this study compared and discussed the rent price prediction accuracy of regression approaches ([1] OLS, [2] spatial statistical model [kriging] and, [3] the DNN) using various sample sizes. According to Efron (2020), although the ML-based approaches have been popularized recently, regression-based models are still useful because the obtained coefficients can be used not only for prediction but also for attribution (i.e., significance testing).

As the sample size increases (e.g., $n = 10^5$), it is increasingly more challenging to straightforwardly apply kriging, which requires the cost of O (n^3) for the inverse matrix calculation of a variance-covariance matrix. Hence, as a spatial statistical model, NNGP was used, which allows for the application of kriging to big data. For empirical validation, this study employed LIFULL HOME's data set[20] for apartment rent prices in Japan—which includes data on rent, lot size, location (municipality, zip code, nearest station, and walk time to nearest station), year built, room layout, building structure, and equipment for approximately 5.33 million properties across Japan. Thus, to assess the effect that the sample size has on the difference in predictive accuracy, properties with missing data were eliminated. Furthermore, $n = 10^4$, 10^5, and 10^6 properties were completely randomly sampled to compare the rent price prediction accuracy based on approaches [1], [2], and [3].

Our analysis showed that, with an increase in sample size, the predictive accuracy of the DNN approached that of the NNGP. Moreover, they were nearly equal in the order of $n = 10^6$. During this experiment, standard explanatory variables that had typically been incorporated into the regression-based hedonic model were used. It is no exaggeration to say that, under these standard settings, the use of

regression-based NNGP is sufficient even if the sample size is in the order of $n = 10^6$. Note, however, that the DNN is expected to be useful in contexts where K is even larger (e.g., when image data is used for explanatory variables). This possibility must await further investigation.

Moreover, regarding both higher-end and lower-end properties whose rent prices deviate from the median, our study suggested that the DNN may have higher predictive accuracy than the NNGP because, unlike the NNGP, the DNN can explicitly consider the nonlinearity of the function form. Thus, the usefulness of the regression approaches that consider the nonlinearity of the function form, as in the geoadditive model (Kammann and Wand, 2003), was demonstrated by the experiment of Seya et al. (2011) using small samples. It will be worthwhile to test such semiparametric approach using big data in the future.

In this study, many DNN hyperparameters were determined using optimization techniques to eliminate tailored and ad-hoc settings as much as possible. Nevertheless, a certain portion of this procedure, including the setting of the parameter search range, had to depend on trial and error. Since the difficulty of setting hyperparameters in DNNs poses an obstacle to their actual operation for applied researchers and practitioners who are involved in the prediction of real estate sales and rent prices, there is an urgent need to accumulate study results to resolve this issue. Additionally, it is also important to establish an effective means to set NNGP hyperparameters, as well as compare it to other neural network models, including graph convolutional networks or even other ML methods such as gradient boosting families.

Notes
1 https://www.zillow.com/
2 https://www.homes.co.jp/price-map
3 See Glumac and Des Rosiers (2020) for more details on automated valuation.
4 For example, although information on individual transactions of real estate properties is officially available from the Land General Information System (http://www.land.mlit.go.jp/webland/) of the Ministry of Land, Infrastructure, Transport and Tourism in Japan, location and price details are not available. That is, only approximate locations and prices are available.
5 Semiparametric functional forms, such as penalized spline, can also be used (Seya et al. 2011). Kuminoff, Parmeter, and Pope (2010) noted that it may be time to reconsider the quadratic Box-Cox model and other flexible specifications for empirical hedonic research.
6 https://www.nii.ac.jp/dsc/idr/lifull/homes.html
7 This was provided by LIFULL Co., Ltd. to the researchers free of charge through the National Institute of Informatics.
8 Fotheringham and Park (2018) conducted hedonic rent price modelling for Seoul. See Melser (2020) about the merit of rents in terms of the hedonic analysis.
9 Semiparametric methods such as the geoadditive model are also employed (Seya et al. 2011; Von Graevenitz and Panduro 2015).
10 Recently, Murakami et al. (2019) suggested the importance of scale in SVC models.
11 Or $m(s_0) + w(s_0)$, see Cressie (1993).
12 Although the results depend on the ordering of the samples, Datta et al. (2016) showed that the NNGP is insensitive to ordering. We performed ordering based on the x-coordinate locations.
13 We also considered the linear functional form, but predictive accuracy assessed at real-scale was worse compared to the log-linear form for all the cases.
14 Of all explanatory variables, information regarding use district (zoning) and floor-area ratio was often lacking in the original database. Therefore, these data were separately prepared from the National Land Numerical Information database (http://nlftp.mlit.go.jp/ksj-e/index.html)
15 One possible means of improvement is to apply the methods of hyperparameters value setting for the DNN, as mentioned in the next section. Future studies can consider the development of a concrete algorithm.
16 In the default setting of the spConjNNGP function, the value is 15.
17 Since $n = 104$, and $n = 106$ did not produce large differences, the results for $n = 105$ are shown here.
18 https://keras.io
19 A framework developed via Preferred Networks, Inc. (https://optuna.org)
20 https://www.nii.ac.jp/dsc/idr/lifull/homes.html

References

Abidoye, R. B., and A. P. Chan. (2017). "Artificial Neural Network in Property Valuation: Application Framework and Research Trend." *Property Management* 35(5), 554–71.

Abidoye, R. B., and A. P. Chan. (2018). "Improving Property Valuation Accuracy: A Comparison of Hedonic Pricing Model and Artificial Neural Network." *Pacific Rim Property Research Journal* 24 (1), 71–83.

Banerjee, S., B. P. Carlin, and A. E. Gelfand. (2014). *Hierarchical Modeling and Analysis for Spatial Data,* 2nd ed. Boca Raton: Chapman & Hall/CRC.

Bergstra, J. S., and Y. Bengio. (2012). "Random Search for Hyper-Parameter Optimization." *Journal of Machine Learning Research* 13, 281–305.

Bergstra, J. S., R. Bardenet, Y. Bengio, and B. Kégl. (2011). "Algorithms for Hyper-Parameter Optimization." *Proceedings for Advances in Neural Information Processing Systems* 24, 2546–54.

Bergstra, J. S., D. Yamins, and D. Cox. (2013). "Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures." In *Proceedings of the 30th International Conference on Machine Learning*, 28, 115–23.

Bourassa, S., E. Cantoni, and M. Hoesli. (2010). "Predicting House Prices with Spatial Dependence: A Comparison of Alternative Methods." *Journal of Real Estate Research* 32(2), 139–59.

Brunsdon, C., S. Fotheringham, and M. Charlton. (1998). "Geographically Weighted Regression." *Journal of the Royal Statistical Society: Series D (The Statistician)* 47(3), 431–43.

Bussas, M., C. Sawade, N. Kühn, T. Scheffer, and N. Landwehr. (2017). "Varying-Coefficient Models for Geospatial Transfer Learning." *Machine Learning* 106(9-10), 1419–40.

Carruthers, J. I., and D. E. Clark. (2010). "Valuing Environmental Quality: A Space-Based Strategy." *Journal of Regional Science* 50(4), 801–32.

Čeh, M., M. Kilibarda, A. Lisec, and B. Bajat. (2018). "Estimating the Performance of Random Forest Versus Multiple Regression for Predicting Prices of the Apartments." *ISPRS International Journal of Geo-Information* 7(5), online.

Cressie, N. (1993). *Statistics for Spatial Data*. New York: Wiley.

Cressie, N., and D. M. Hawkins. (1980). "Robust Estimation of the Variogram: I." *Journal of the International Association for Mathematical Geology* 12(2), 115–25.

Cressie, N., and C. K. Wikle. (2011). *Statistics for Spatio-Temporal Data*. Hoboken: John Wiley and Sons.

Datta, A., S. Banerjee, A.O. Finley, and A.E. Gelfand. (2016). "Hierarchical Nearest-Neighbor Gaussian Process Models for Large Geostatistical Datasets." *Journal of the American Statistical Association* 111(514), 800–12.

Dubin, R. A. (1988). "Estimation of Regression Coefficient in the Presence of Spatially Autocorrelated Error Terms." *The Review of Economics and Statistics* 70(3), 466–74.

Efron, B. (2020). "Prediction, Estimation, and Attribution." *Journal of the American Statistical Association* 115(530), 636–55.

Finley, A. O., A. Datta, B. C. Cook, D. C. Morton, H. E. Andersen, and S. Banerjee. (2017). "Applying Nearest Neighbor Gaussian Processes to massive Spatial Data Sets: Forest Canopy Height Prediction Across Tanana Valley Alaska." https://arxiv.org/pdf/1702.00434.pdf

Fotheringham, A. S., C. Brunsdon, and M. Charlton. (2002). *Geographically Weighted Regression*. Chichester: John Wiley and Sons.

Fotheringham, A. S., and B. Park. (2018). "Localized Spatiotemporal Effects in the Determinants of Property Prices: A Case study of Seoul." *Applied Spatial Analysis and Policy* 11(3), 581–98.

Gelfand, A. E., H. J. Kim, C. F. Sirmans, and S. Banerjee. (2003). "Spatial Modeling with Spatially Varying Coefficient Processes." *Journal of the American Statistical Association* 98(462), 387–96.

Georgiadis, A. (2018). "Real Estate Valuation Using Regression Models and Artificial Neural Networks: An Applied Study in Thessaloniki." *RELAND: International Journal of Real Estate & Land Planning* 1, online.

Glorot, X., A. Bordes, and Y. Bengio. (2011). "Deep Sparse Rectifier Neural Networks." *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* 15, 315–23.

Glumac, B., and F. Des Rosiers. (2020). "Practice Briefing–Automated Valuation Models (Avms): Their Role, Their Advantages and Their Limitations." *Journal of Property Investment & Finance*, in print.

Heaton, M. J., A. Datta, A. O. Finley, R. Furrer, R. Guhaniyogi, F. Gerber, R. B. Gramacy, D.

Hammerling, M. Katzfuss, F. Lindgren, D. W. Nychka, F. Sun, and A. Zammit-Mangion. (2019). "A Case Study Competition Among Methods for Analyzing Large Spatial Data." *Journal of Agricultural, Biological and Environmental Statistics* 24(3), 398–425.

Huang, B., B. Wu, and M. Barry. (2010). "Geographically and Temporally Weighted Regression for Modeling Spatio-Temporal Variation in House Prices." *International Journal of Geographical Information Science* 24(3), 383–401.

Inoue, R., R. Ishiyama, and A. Sugiura. (2020). "Identifying Local Differences With Fused-MCP: An Apartment Rental Market Case Study On Geographical segmentation detection." *Japanese Journal of Statistics and Data Science* 3, 183–214.

Kelejian, H. H., and I. R. Prucha. (2007). "The Relative Efficiencies of Various Predictors in Spatial Econometric Models Containing Spatial Lags." *Regional Science and Urban Economics* 37(3), 363–74.

Kuminoff, N. V., C. F. Parmeter, and J. C. Pope. (2010). "Which Hedonic Models Can We Trust to Recover the Marginal Willingness to Pay for Environmental Amenities?." *Journal of Environmental Economics and Management* 60(3), 145–160.

Krizhevsky, A., I. Sutskever, and G. E. Hinton. (2012). "Imagenet Classification with Deep Convolutional Neural Networks." *Proceedings of the 25th International Conference on Neural Information Processing Systems* 1, 1097–105.

James, V., S. Wu, A. Gelfand, and C. Sirmans. (2005). "Apartment Rent Prediction Using Spatial Modeling." *Journal of Real Estate Research* 27(1), 105–36.

Kammann, E. E., and M. P. Wand. (2003). "Geoadditive Models." *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 52(1), 1–18.

Kingma, D. P., and J. Ba. (2014). "Adam: A Method for Stochastic Optimization." arXiv preprint arXiv:1412.6980.

Kontrimas, V., and A. Verikas. (2011). "The Mass Appraisal of the Real Estate by Computational Intelligence." *Applied Soft Computing* 11(1), 443–8.

LeCun, Y., Y. Bengio, and G. Hinton. (2015). "Deep Learning." *Nature* 521(7553), 436.

Li, Z.. and A. S. Fotheringham. (2020). "Computational Improvements to Multi-Scale Geographically Weighted Regression." *International Journal of Geographical Information Science*, in print.

Melser, D. (2020). "Estimating the Housing Capitalization Effects of New Infrastructure: Should We Be Using Rents Instead of Prices?." *Transportation Research Part A: Policy and Practice* 138, 402–21.

Murakami, D., B. Lu, P. Harris, C. Brunsdon, M. Charlton, T. Nakaya, and D. A. Griffith. (2019). "The Importance of Scale in Spatially Varying Coefficient Modeling." *Annals of the American Association of Geographers* 109(1), 50–70.

Murakami, D., N. Tsutsumida, T. Yoshida, T. Nakaya, and B. Lu. (2020). "Scalable GWR: A Linear-Time Algorithm for Large-Scale Geographically Weighted Regression with Polynomial Kernels." *Annals of the American Association of Geographer,* in print.

Murakami, D., T. Yoshida, H. Seya, D. A. Griffith, and Y. Yamagata. (2017). "A Moran Coefficient-Based Mixed Effects Approach to Investigate Spatially Varying Relationships." *Spatial Statistics* 19, 68–89.

Pace, R. K., R. Barry, and C. F. Sirmans. (1998). "Spatial Statistics and Real Estate." *The Journal of Real Estate Finance and Economics* 17(1), 5–13.

Pace, R. K., and D. Hayunga. (2020). "Examining the Information Content of Residuals from Hedonic and Spatial Models Using Trees and Forests." *The Journal of Real Estate Finance and Economics* 60(1-2), 170–80.

Rosen, S. (1974). "Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition." *Journal of Political Economy* 82(1), 34–55.

Saha, A. and A. Datta. (2018). "BRISC: Bootstrap for Rapid Inference on Spatial Covariances." *Stat* 7(1), e184.

Schabenberger, O. and C. A. Gotway. (2005). *Statistical Methods for Spatial Data Analysis*. Boca Raton: Chapman Hall/CRC.

Selim, H. (2009). "Determinants of House Prices in Turkey: Hedonic Regression Versus Artificial Neural Network." *Expert Systems with Applications* 36(2), 2843–52.

Seya, H., M. Tsutsumi, Y. Yoshida, and Y. Kawaguchi. (2011). "Empirical Comparison of the Various Spatial Prediction Models: In Spatial Econometrics, Spatial Statistics, and Semiparametric Statistics." *Procedia-Social and Behavioral Sciences* 21, 120–29.

Swandon, D. A., J. Tayman, and C. F. Barr. (2000). "A Note on the Measurement of Accuracy for Subnational Demographic Estimates." *Demography* 37(2), 193–201.

Taylor, L. O. (2008). "Theoretical Foundations and Empirical Developments in Hedonic

Modeling." In *Hedonic Methods in Housing Markets*, 15-37, edited by A. Baranzini, J. Ramirez, C. Schaerer, and P. Thalmann. New York: Springer.

Tieleman, T., and G. Hinton. (2012). "Lecture 6.5-rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude." *COURSERA: Neural Networks for Machine Learning* 4(2), 26–31.

Tsutsumi, M., and H. Seya. (2009). "Hedonic Approaches Based on Spatial Econometrics and Spatial Statistics: Application to Evaluation of Project Benefits." *Journal of Geographical Systems* 11(4), 357–80.

Valier, A. (2020). "Who Performs Better? AVMs vs Hedonic Models." *Journal of Property Investment & Finance,* in print.

Vecchia, A. V. (1988). "Estimation and Model Identification for Continuous Spatial Processes." *Journal of the Royal Statistical Society: Series B (Methodological)* 50(2), 297–312.

Von Graevenitz, K., and T. E. Panduro. (2015). "An Alternative to the Standard Spatial Econometric Approaches in Hedonic House Price Models." *Land Economics* 91(2), 386–409.

Wang, C., M. A. Puhan, R. Furrer, and SNC Study Group. (2018). "Generalized Spatial Fusion Model Framework for Joint Analysis of Point and Areal Data." *Spatial Statistics* 23, 72–90.

Yalpır, Ş. (2018). "Enhancement of Parcel Valuation with Adaptive Artificial Neural Network Modeling." *Artificial Intelligence Review* 49(3), 393–405.

Yamagata, Y., and H. Seya. (2019). *Spatial Analysis Using Big Data: Methods and Urban Applications*. Cambridge, USA: Academic Pres.

Zhang, L., A. Datta, and S. Banerjee. (2019). "Practical Bayesian Modeling and Inference for Massive Spatial Data Sets on Modest Computing Environments." *Statistical Analysis and Data Mining: The ASA Data Science Journal* 12(3), 197–209.

Zurada, J., A. Levitan, and J. Guan. (2011). "A Comparison of Regression and Artificial Intelligence Methods in a Mass Appraisal Context." *Journal of Real Estate Research* 33(3), 349–87.

Figures

Legend:
- [10.7 to 10.8)
- [10.8 to 10.9)
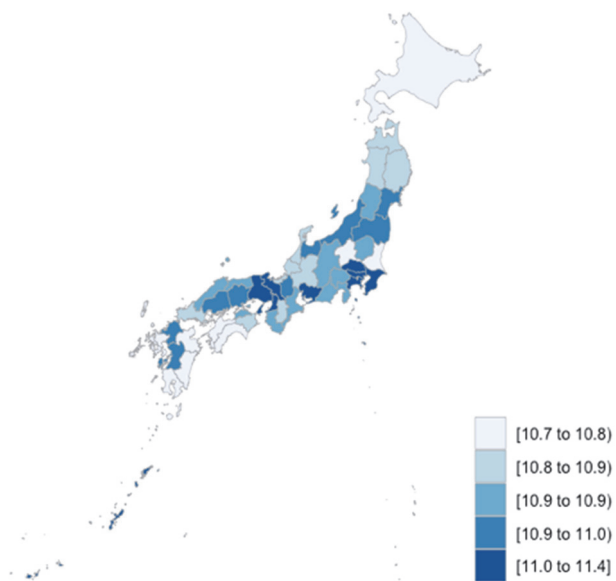- [10.9 to 10.9)
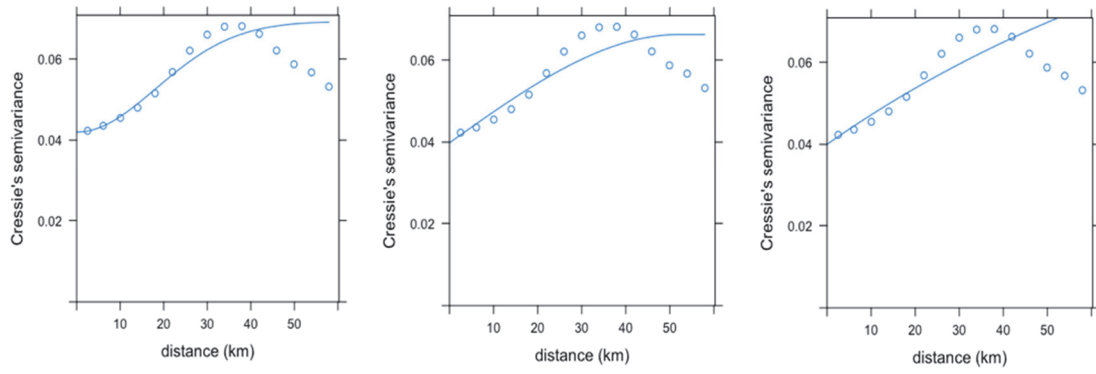- [10.9 to 11.0)
- [11.0 to 11.4]

Figure 1: log (rent prices) for each prefecture

Figure 2: Fitting of variogram functions
(Gaussian model; Spherical model; Exponential model)
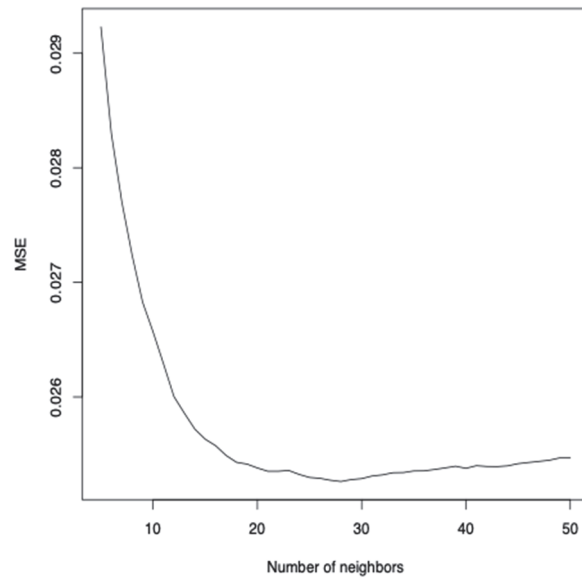
Figure 3: Change in the MSE according to the number of nearest neighbors (in the case of $n=10^5$)

Figure 4: Prediction results by sample size for each model (log scale)

MAPE (real scale)

Legend: ■ N: 10^6 ■ N: 10^5 ■ N: 10^4
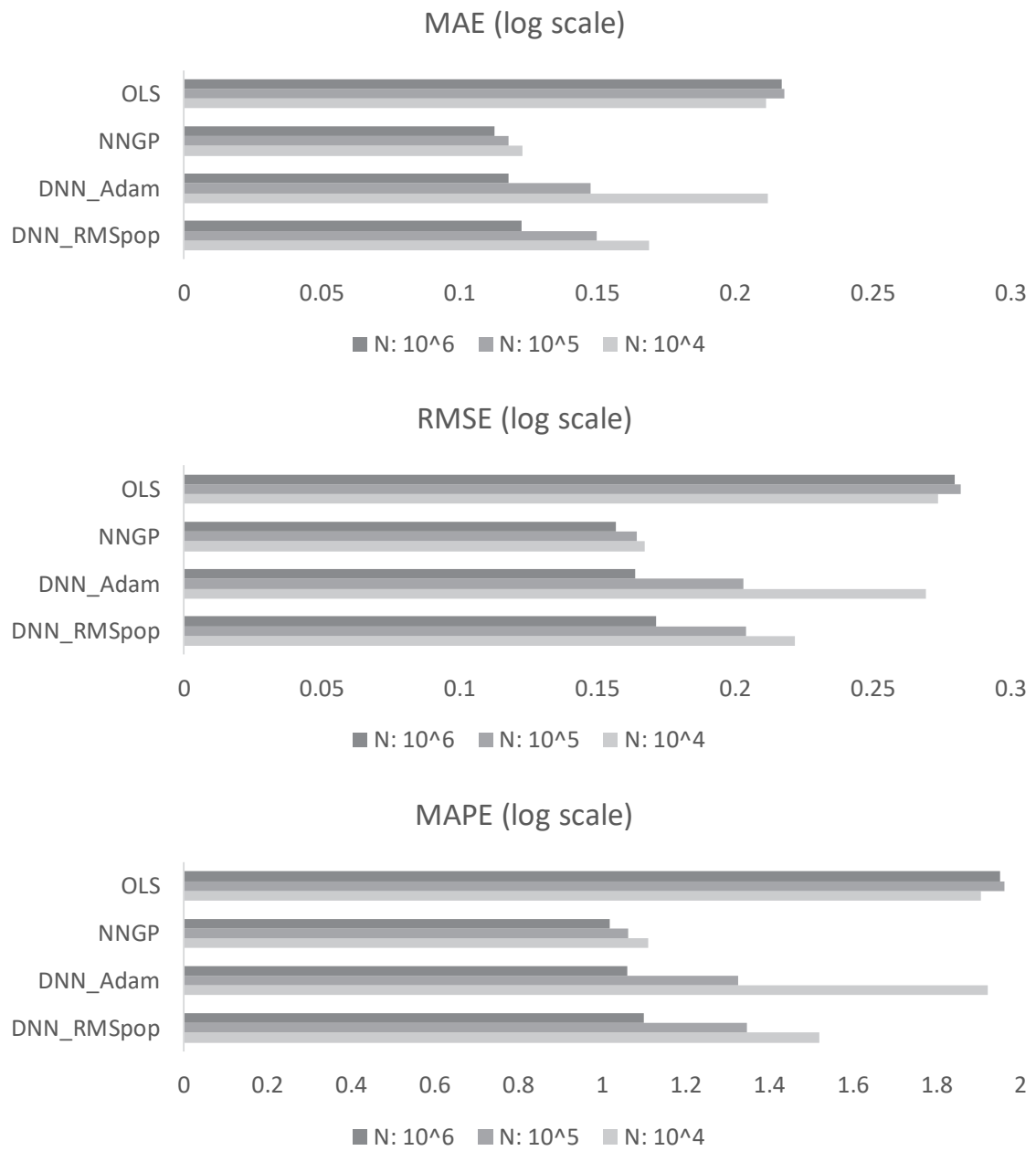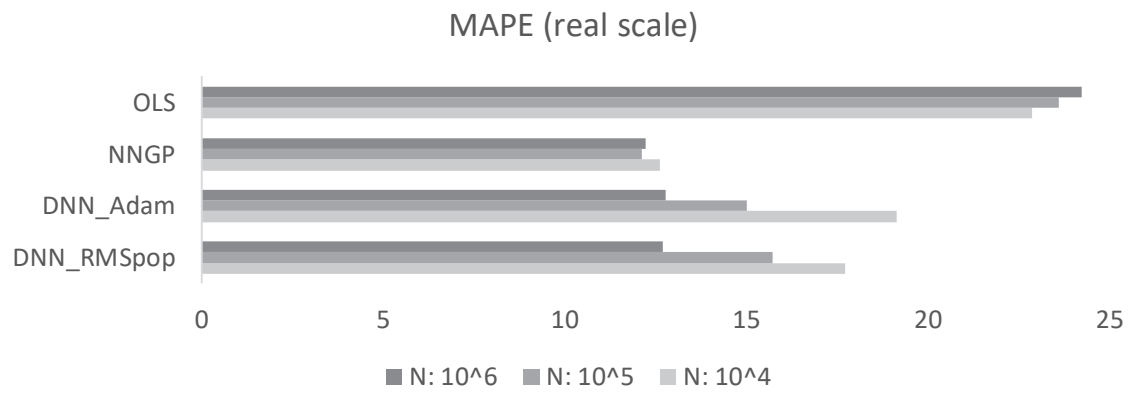
Figure 5: MAPE by sample size for each model (real scale)
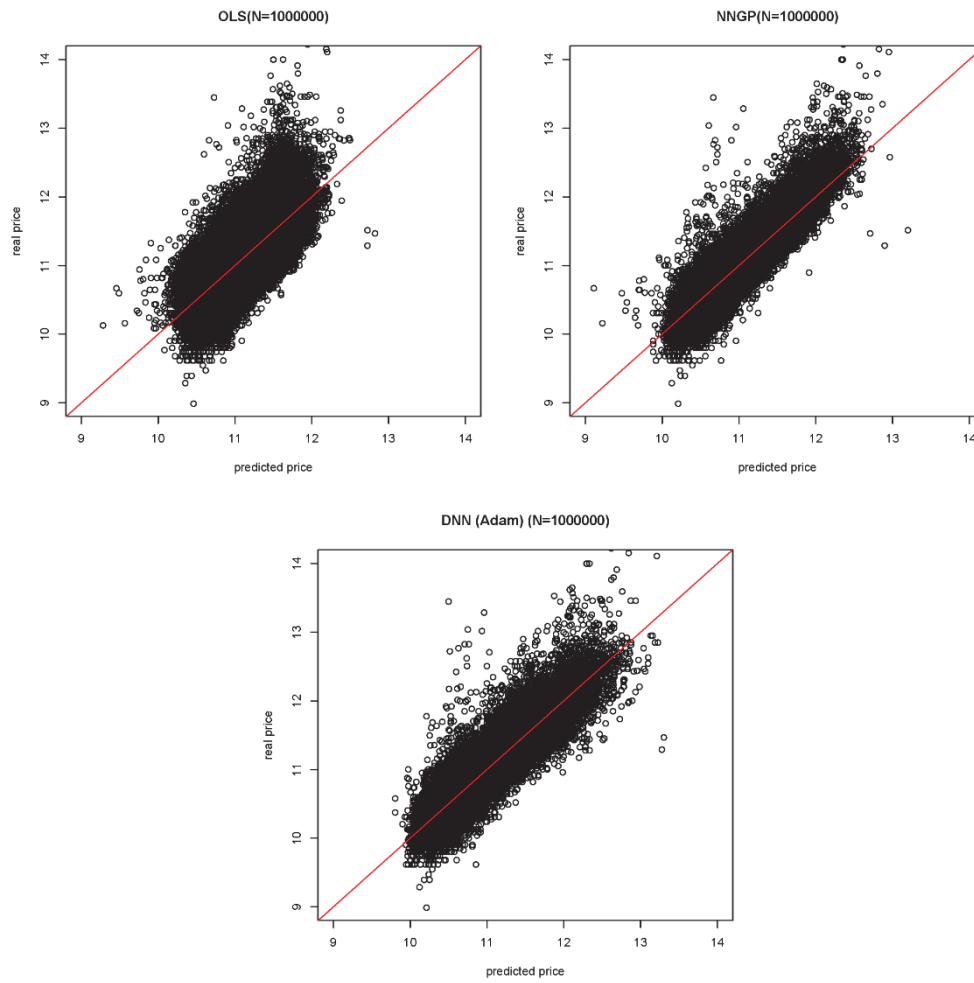
Figure 6: Scatter plot of predicted (horizontal axis) and actual (vertical axis) log rent prices for each model (in the case of $n = 10^6$)
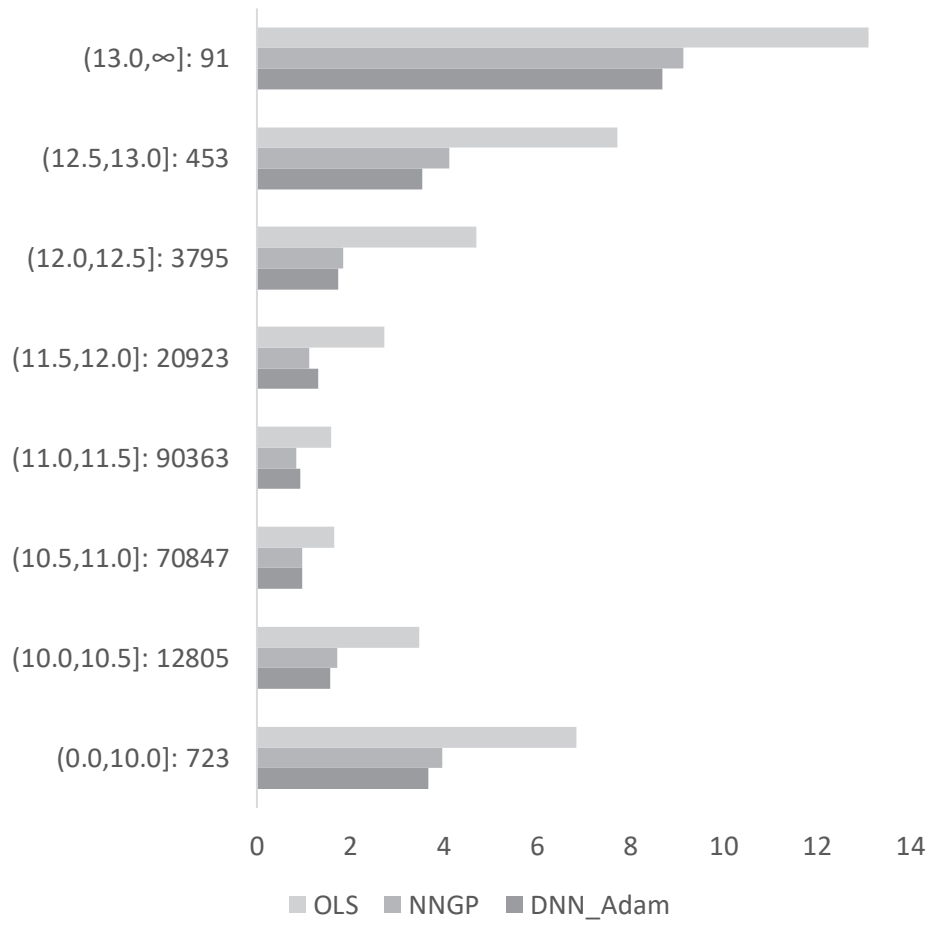
Figure 7: MAPE per log rent range (in the case of $n = 10^6$)

Tables

Table 1-1: Descriptive statistics (continuous variables)

	Min	Max	Median	Mean	SD
Rent price (yen)	5250	1250000000	63000	72850	1381893
Years built (month)	5	1812	228	236	135.6
Walk time to nearest (train) station (m)	1	88000	640	781.5	661.3
Number of rooms (#)	1	50	1	1.48	0.71
Floor-area ratio (%)	50	1000	200	234.1	130.6
X (km)	−841	783.1	352.2	181.5	273.3
Y (km)	2958	5029	3931	3942	195.3

The "rent price" includes maintenance fees

Table 1-2: List of explanatory variables (discrete variables)

Direction	North, Northeast, East, Southeast, South, Southwest, West, Northwest, Other
Building structure	W, B, S, RC, SRC, PC, HPC, LS, ALC, RCB, Others
Room layout	R, K, SK, DK, SDK, LK, SLK, LDK, SLDK
Use district	Category I exclusively low residential zone (1 Exc Low), Category II exclusively low residential zone (2 Exc Low), Category I exclusively high-medium residential zone (1 Exc Med), Category II exclusively high-medium residential zone (2 Exc Med), Category I residential zone (1 Res), Category II residential zone (2 Res), Quasi-residential zone (Quasi-Res), Neighborhood commercial zone (Neighborhood Comm), Commercial zone (Commercial), Quasi-Industrial zone (Quasi-Ind), Industrial zone (Industrial), Exclusive industrial zone (Exc Ind), Others (Others)

For building structure: W: Wooden; B: Concrete block; S: Steel frame; RC: Reinforced concrete; SRC: Steel frame reinforced concrete; PC: precast concrete; HPC: Hard precast concrete; LS: Light steel, RCB: Reinforced concrete block

For room layout: The R refers to a room where there is only one room and there is no wall to separate the bedroom from the kitchen. For the others, K: includes a kitchen; D: includes a dining room: L: includes a living room; S: additional storage room. For example, LDK is a Living, Dining, and Kitchen area.

For use district: Category I exclusively low residential zone, Category II exclusively low residential zone, Category I exclusively medium-high residential zone, Category II exclusively medium-high residential zone, Category I residential zone, Category II residential zone, Quasi-residential zone, Neighborhood commercial zone, Commercial zone, Quasi-industrial zone, Industrial zone, Exclusively industrial zone

Table 1-3: Descriptive statistics (discrete variables)
 # denotes the number of cases

	Direction			Structure			Use district			Room layout	
	Count	Share		Count	Share		Count	Share		Count	Share
North	156843	0.0342	W	1024081	0.2232	1 Exc Low	780638	0.1701	R	423815	0.0924
Northeast	81173	0.0177	B	570	0.0001	2 Exc Low	25793	0.0056	K	1729903	0.3770
East	595252	0.1297	S	844184	0.1840	1 Exc Med	689879	0.1503	SK	6919	0.0015
Southeast	473041	0.1031	RC	1892428	0.4124	2 Exc Med	321441	0.0701	DK	890584	0.1941
South	1749315	0.3812	SRC	190048	0.0414	1 Res	1030319	0.2245	SDK	5123	0.0011
Southwest	458125	0.0998	PC	11924	0.0026	2 Res	211076	0.0460	LK	516	0.0001
West	404994	0.0883	HPS	802	0.0002	Quasi-Res	59863	0.0130	SLK	138	0.0000
Northwest	78836	0.0172	LS	559974	0.1220	Neighborhood Comm	386531	0.0842	LDK	1505821	0.3282
Others	591053	0.1288	ALC	58373	0.0127	Commercial	615630	0.1342	SLDK	25813	0.0056
			RCB	597	0.0001	Quasi-Ind	371672	0.0810			
			Others	5651	0.0012	Industrial	83826	0.0183			
						Exc Ind	11949	0.0026			
						Others	15	0.0000			

Table 2: Regression analysis results using OLS (example of $n = 10^6$)

Variables	Estimate	t-values	Pr(>|t|)	sig.code
Constant term	1.08E+01	4.50E+03	2.00E-16	***
Years built	-1.15E-03	-4.42E+02	2.00E-16	***
Walk time to nearest station	-4.88E-05	-9.87E+01	2.00E-16	***
Floor-area ratio	1.30E-03	2.30E+02	2.00E-16	***
Number of rooms	1.49E-01	2.57E+02	2.00E-16	***
Direction_Northeast	8.09E-02	2.80E+01	2.00E-16	***
Direction_East	-4.45E-03	-2.32E+00	2.05E-02	*
Direction_Southeast	5.40E-03	2.72E+00	6.63E-03	**
Direction_South	-2.33E-02	-1.29E+01	2.00E-16	***
Direction_Southwest	2.46E-03	1.23E+00	2.19E-01	
Direction_West	1.94E-02	9.67E+00	2.00E-16	***
Direction_Northwest	7.39E-02	2.53E+01	2.00E-16	***
Direction_Others	-6.85E-02	-3.53E+01	2.00E-16	***
Structure_B	1.88E-01	6.39E+00	1.66E-10	***
Structure_S	9.41E-02	9.39E+01	2.00E-16	***
Structure_RC	2.40E-01	2.71E+02	2.00E-16	***
Structure_SRC	3.67E-01	2.06E+02	2.00E-16	***
Structure_PC	2.14E-01	3.48E+01	2.00E-16	***
Structure_HPC	9.13E-02	4.19E+00	2.78E-05	***
Structure_LS	5.34E-02	4.75E+01	2.00E-16	***
Structure_ALC	9.17E-02	3.19E+01	2.00E-16	***
Structure_RCB	1.20E-01	4.55E+00	5.44E-06	***
Structure_Others	1.61E-01	1.81E+01	2.00E-16	***
Room layout_K	4.22E-02	3.62E+01	2.00E-16	***
Room layout_SK	1.10E-01	1.39E+01	2.00E-16	***
Room layout_DK	1.37E-01	1.00E+02	2.00E-16	***
Room layout_SDK	3.65E-01	3.95E+01	2.00E-16	***
Room layout_LK	2.79E-01	1.01E+01	2.00E-16	***
Room layout_SLK	3.04E-01	5.75E+00	9.07E-09	***
Room layout_LDK	2.76E-01	2.12E+02	2.00E-16	***
Room layout_SLDK	6.06E-01	1.39E+02	2.00E-16	***
Use district_2 Exc Low	-1.15E-01	-2.72E+01	2.00E-16	***
Use district_1 Exc Med	-1.52E-01	-1.22E+02	2.00E-16	***
Use district_2 Exc Med	-2.77E-01	-1.83E+02	2.00E-16	***
Use district_1 Res	-2.36E-01	-1.97E+02	2.00E-16	***
Use district_2 Res	-2.48E-01	-1.38E+02	2.00E-16	***
Use district_ Quasi-Res	-2.92E-01	-9.98E+01	2.00E-16	***
Use district_ Neighborhood Comm	-2.63E-01	-1.56E+02	2.00E-16	***
Use district_ Commercial	-4.63E-01	-1.83E+02	2.00E-16	***
Use district_ Quasi-Ind	-1.91E-01	-1.26E+02	2.00E-16	***
Use district_ Industrial	-2.46E-01	-9.76E+01	2.00E-16	***
Use district_ Exc Ind	-3.17E-01	-5.13E+01	2.00E-16	***
Use district_Others	5.00E-01	3.57E+00	3.62E-04	***
Adjusted R^2		0.5165		

* significant at 5%; ** significant at 1%; *** significant at 0.1%.

Table 3: DNN hyperparameters and search range

Hyper parameters	Search range
# of hidden layers	[1, 5]
# of unites	[10, 50]
Batch size	[32, 128]
# of epochs	[10, 30]
Learning rate	$[10^{-5}, \ 10^{-2}]$ (log)

Table 4: DNN hyperparameters after optimization (Adam)

	$n = 10^4$	$n = 10^5$	$n = 10^6$
# of hidden layers	5	4	4
# of unites	[10, 10, 42, 37, 10]	[48, 26, 50, 36]	[37,43, 50, 42]
Batch size	34	71	103
# of epochs	27	29	22
Learning rate	0.002896	0.007036	0.0007060