Computational Economics (2024) 64:579–594
https://doi.org/10.1007/s10614-023-10456-6

Housing GANs: Deep Generation of Housing Market Data

Bilgi Yilmaz[1,2]

Accepted: 10 August 2023 / Published online: 23 August 2023

Abstract
Modeling housing markets is a challenging and central research area since they are highly related to the economy. However, the limited available data prevents researchers from improving models. As an alternative, this study introduces Housing GANs, a data-driven modeling approach inspired by the recent success of generative adversarial networks (GANs). The Housing GANs include a generator and discriminator function utilizing Wasserstein GAN with gradient penalty and mitigate original housing datasets, including continuous and discrete data. The generator function predicts the real data distribution and generates realistic housing data. The empirical analysis highlights that the Housing GANs successfully learns the distribution and generate realistic housing data in high fidelity.

Keywords Generative adversarial networks · Machine learning · Housing market · Synthetic data generation

1 Introduction

Housing price plays a significant role in shaping the economy. Housing renovation and construction boost the economy by increasing the house sales rate, employment, and expenditures. The traditional housing price prediction methods are based on the sales price comparison, which hardly achieves valuable accuracy. However, there are various advanced methods commonly used to model the housing market and predict housing prices, such as multiple regression (Yusof & Ismail, 2012), hedonic regression (Boyle & Kiel, 2001; Clapp & Giaccotto, 1998a, b; Colwell & Dilmore, 1999; Coulson & Bond, 1990) support vector regression (Chen et al., 2017), time-series

✉ Bilgi Yilmaz
bilgiyilmaz07@gmail.com

[1] Mathematics, RPTU Kaiserslautern-Landau, Gottlieb-Daimler-Strasse, 67663 Kaiserslautern, Germany

[2] Mathematics, Kahramanmaras Sutcu Imam University, Bati Cevreyolu Bulvari, 46050 Kahramanmaras, Turkey

analysis (Yilmaz & Selcuk-Kestel, 2020), stochastic calculus (Korn & Yilmaz, 2022; Yilmaz & Selcuk-Kestel, 2018; Yilmaz et al., 2022a, b), spatial econometrics (Huang et al., 2010), machine learning (Gu et al., 2011; Khamis & Kamarudin, 2014; Raj & Ananthi, 2019), micro-econometric models (Meen, 2011; Van Leuvensteijn & Koning, 2004), and macro-econometric models (Al-Homoud et al., 2009; Puri & Van Lierop, 1988). The choice of method depends on the research question, the data availability, and the specific goals of the analysis.

In the housing market analysis and modeling, property characteristics (e.g., the size, age, location, and other attributes of each property), neighborhood characteristics (e.g., the quality of schools, crime rates, and access to amenities), market conditions (e.g., supply and demand, interest rates, and economic indicators), time-series data (data covering a period), price data (the sale prices of properties) consumer preferences (e.g., proximity to amenities or specific features) are essential for analyzing housing market and developing and validating housing price models.

Various government agencies and organizations provide housing-related data, such as census data, property records, and housing market indicators, that are publicly available. These datasets offer valuable insights, but generally, they have limited granularity or lack specific details needed for advanced analysis. Some companies also offer comprehensive housing market data that includes property characteristics, transaction history, and market trends. However, such datasets often come with a cost, and access is generally limited to specific regions or countries. Furthermore, the data may not cover niche markets or emerging trends. Multiple listing services (MLS) databases contain information on properties available for sale or rental housing. They summarize detailed property listings, including prices, square footage, and location. However, accessing MLS data can be challenging, as it is typically restricted to licensed real estate agents and brokers. Real estate portals (online platforms) like Zillow, Trulia, and Redfin gather housing data from various sources, including MLS and public records. Such platforms provide a wealth of housing information but generally need complete coverage, especially in certain areas or for specific property types. Some academic and research institutions compile housing datasets for specific studies or research purposes. These datasets can be valuable but limited in scope or not readily accessible to the broader public.

The limited data on the housing market is one of the major problems in modeling since the advanced modeling methods require detailed data on the various attributes of a house, which is challenging to obtain or inconsistent and expensive. A sufficient amount of high-quality data impacts housing market analysis and housing price prediction and forecasting (e.g., it may solve the problems of inaccurate results, limited scope, difficulty in comparison, unreliable predictions and forecasting, and limited understanding of the market). Consequently, the existing historical data of housing markets typically includes the average transaction housing price and has yet to be weighted or estimated using a model, encompasses a wide range of information, including property attributes, market conditions, and transaction details, which can be further analyzed and modeled to derive meaningful insights and predictions. Although some modeling methods are present, they need more data for efficient housing market analysis and housing price forecasting (Stevenson, 2008). This problem can be solved by using artificially generated realistic data sets. Therefore, the

study offers to use Generative Adversarial Networks (GANs) to generate synthetic data for housing markets in the light of recent developments in synthetic data generation using GANs. Generating a synthetic data set offers several advantages over relying solely on existing datasets. Synthetic data generation can help overcome limitations in existing data sources by providing the following:

1. Data privacy: Synthetic data can be anonymized, mitigating privacy concerns associated with real-world data and facilitating sharing and collaboration without compromising individuals' personal information. Thus, synthetic data allows researchers to work with realistic data without compromising the privacy of real individuals.
2. Data scarcity: In cases where comprehensive and high-quality datasets are scarce or expensive to obtain, synthetic data can be generated to augment the sample size and improve the accuracy of the analysis.
3. Model validation: Synthetic data can be used to validate and compare different models by generating data that follows specific patterns or distributions.
4. Improved understanding: Synthetic data can be used to explore hypothetical scenarios and better understand the impact of different factors on housing markets.
5. Fairness and bias testing: Synthetic data can be used to evaluate and test algorithms for fairness and bias, ensuring that the results are unbiased and representative of the real-world population.

Therefore, depending on the specific research goals, the time and effort invested in creating a synthetic data set can be worthwhile, especially when existing datasets have significant shortcomings or must fulfill the analysis requirements.

The concept of GANs was initially proposed by Goodfellow et al. (2014). Furthermore, they have since established themselves as a versatile and effective tool for various generative tasks such as image manipulation and generation, natural language processing, and video prediction. Their ability to generate synthetic samples with similar statistical properties to real data has made GANs a popular choice among researchers (Arjovsky et al., 2017; Galteri et al., 2019; Gulrajani et al., 2017; Radford et al., 2015; Wu et al., 2016; Yu et al., 2017; Zhu et al., 2017). Although initially designed for image processing and computer vision, GANs' success in producing realistic images has sparked widespread interest and applications in various research fields (Silva et al., 2021). In recent years, GANs' superior performance compared to other generative models has also attracted researchers in sequential data domains such as medical data (Esteban et al., 2017), finance (Wiese et al., 2020), and electricity load data (Yilmaz & Korn, 2022). For instance, Wiese et al. (2020) used GANs to approximate a realistic asset price simulator for financial markets, while Yilmaz and Korn (2022) generated synthetic individual electricity consumption data using various GANs, and Bendaoud et al. (2021), Gu et al. (2019), Fekri et al. (2019), Tian et al. (2019), Zhou et al. (2020), Wang and Hong (2020), Chen et al. (2018), Yuan et al. (2021) applied GANs to generate synthetic load data and generating power generation scenarios for renewable energy sources.

Given the shortcomings above, including limited granularity, incomplete coverage, and data scarcity, this study recognizes the potential benefits of synthetic data generation in addressing these issues. By leveraging Conditional Tabular GAN (CTGAN) introduced by Xu et al. (2019), this study aims to generate synthetic tabular data specifically tailored to the housing market, called Housing GANs. By utilizing synthetic data, the study aims to overcome existing dataset limitations, improve the accuracy and robustness of housing market analysis, and enable insights into hypothetical scenarios and the effect of various factors on the housing market.

The benefits of the Housing GANs are: It allows the generation of realistic datasets without data privacy concerns since the data can be anonymized. Once the model is built, it can generate samples as the users request with the help of conditions. It can improve the efficiency of housing market modeling since the housing data at hand can be enlarged by combining real and synthetic data. It can generate data even for local housing markets where the trade is rare. It can be used for housing market scenario analysis.

This study introduces empirical analysis based on house price data from Zillow (1999), freely available online, to show the accuracy of the Housing GANs. The Zillow platform gathers housing data from various sources, including MLS and public records. It provides data, an index value that measures typical house values and market changes across a given region and housing type. It mirrors the typical house values in the 35th to 65th percentile range. However, it still suffers from certain drawbacks. For instance, it relies on publicly available data, which can be incomplete or inaccurate. Hence, it is incomplete coverage, particularly in specific areas or for certain housing types. Consequently, the data may capture only some of the housing market, leaving gaps in understanding localized trends and dynamics. Furthermore, it does not consider local market conditions, such as a sudden increase or decrease in supply or demand. It is also worth mentioning that Zillow's data is only an estimate of housing prices. It is not an appraisal.

The study is structured as follows: Sect. 2 introduces the intuition behind the Housing GANs. The empirical analysis of Housing GANs is summarized in Sect. 3, along with the statistical evaluations. Finally, Sect. 4 concludes the study.

2 Methodology

2.1 Generative Adversarial Networks (GANs)

GANs aim to train two neural networks: a generator (G) and a discriminator (D). The generator maps random noise or latent space to real data samples. At the same time, the discriminator attempts to distinguish between real and fake (generated) samples by assigning high probabilities to real samples and low probabilities to generated samples. The ultimate goal of GANs is to generate random variables with the same distribution as the real data, as initially defined by Goodfellow et al. (2014).

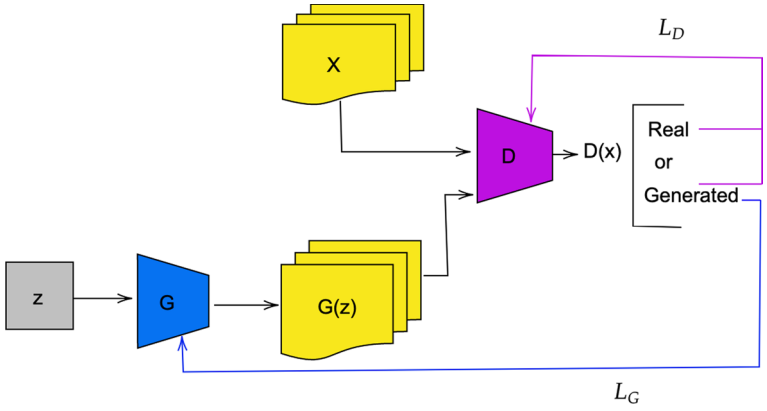

Fig. 1 Illustration of a standard generative adversarial network (GANs) architecture

Definition 1 Let $\{x_i : i = 1, 2, \ldots, n\}$ denotes real data samples having an unknown distribution $p_X(x)$. Let the neural networks generator G_θ and discriminator D_ω be parameterized with θ and ω. Let G_θ be a mapping from the latent space $z \in \mathbb{R}^K$ to data samples $x \in \mathbb{R}^D$, while D_ω be a mapping from both real data samples and generated samples to distinguished values. Then, GANs are trained to solve the following minimax problem

$$\min_{\theta} \max_{\omega} \left(\mathbb{E}_{x\sim\mu}\left[\log\left(\sigma(D_\omega(x))\right)\right] + \mathbb{E}_{z\sim\theta}\left[\log\left(1 - \sigma(D_\omega(G_\theta(z)))\right)\right]\right),$$

where $\sigma(\cdot)$ is the sigmoid function.

Definition highlights that the discriminator and generator are interdependent. The discriminator aims to optimize the problem by controlling only its parameters (ω), while the generator aims to minimize the problem by adjusting its parameters (θ) (Yilmaz, 2021). Both the discriminator and generator are implemented as neural networks, which can be designed as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or Autoencoders. The discriminator and generator require loss functions L_D and L_G, respectively, to update their parameters (see Fig. 1). Both loss functions are are typically based on binary cross-entropy loss. The generator updates its parameters only through feedback from the discriminator based on the fake output. Meanwhile, the discriminator updates its parameters using both real and fake outputs. During training, the generator only uses random samples and never sees real data, and its output is updated only based on information from the discriminator. In this sense, GANs are purely data-driven models.

During the training process, GANs utilize both real or training data and the synthetic data generated by the generator. The discriminator's role is to differentiate between the training and synthetic data sets. The generator's objective is to create artificial samples that resemble the training samples, effectively deceiving the discriminator into believing they are real. The generator starts by transforming random noise from a known distribution, such as Gaussian or uniform, into synthetic samples that should reflect the

training data distribution. The objective is to win the zero-sum game by successfully deceiving the discriminator. When the discriminator correctly identifies a synthetic sample, the generator's loss is recalculated. The discriminator continues to learn and improve its ability to distinguish synthetic samples from real data, assigning a negative loss value if it fails to identify a synthetic sample correctly.

Figure 1 illustrates the general architecture of standard GANs. In the illustration, X represents the real data or training data, including samples that the generator attempts to learn and replicate to produce synthetic data, typically in the form of a batch. z is a random noise vector originating from the latent space sampled from a known distribution, such as a multivariate normal (MVN) or uniform distribution. It serves as the input for the generator to create synthetic samples. The generator, represented by G, is a neural network trained to imitate the distribution of X and produce artificial samples that resemble the samples from X. The generator uses the vectors from z to produce synthetic samples ($G(z)$), which should have a distribution indistinguishable from the distribution of the samples drawn from X. The discriminator, represented by D, is a neural network trained to distinguish between samples drawn from X and $G(z)$. The inputs to the discriminator are X and $G(z)$, and it produces a binary output for each sample, indicating whether it is real or synthetic.

The GANs given by Goodfellow et al. (2014) is extended to Wasserstein GAN (WGAN) by Arjovsky et al. (2017) to solve the mode collapse and vanishing gradient challenges by optimizing the Wasserstein-1 or earth-moving distance. However, the computation of the Wasserstein-1 distance is intractable. Thus, Arjovsky et al. (2017) approximate the Wasserstein-1 distance by reorganizing the optimization problem as

$$\min_G \max_D \mathbb{E}_{x\sim p_{data}}\Big[D(x)\Big] - \mathbb{E}_{z\sim p_z}\Big[D(G(z))\Big],$$

where D belongs to a k-Lipschitz function family. The k-Lipschitz constraint is guaranteed by weight-clipping so that it lies in the compact space $[-c, c]$, e.g., $c = 0.01$. However, the weight-clipping limits discriminator capacity, pushes its weights to the two extreme values of the compact space, and leads to vanishing gradients. Therefore, Gulrajani et al. (2017) proposed the WGAN with gradient penalty (WGAN-GP) as a solution to such problems. It replaces weight clipping with a gradient penalty parameter and takes advantage of the fact that a differentiable function is 1-Lipschitz if and only if its gradients have a norm of no more than one at all points. As a result, the Lipschitz constraint can be enforced by penalizing the discriminator with a gradient penalty, which then makes the objective of the GAN

$$\min_G \max_D \mathbb{E}_{x\sim p_{data}}\Big[D(x)\Big] - \mathbb{E}_{z\sim p_z}\Big[D(G(z))\Big] - \lambda\mathbb{E}_{\hat{x}\sim p_{\hat{x}}}\Big[\|\nabla_{(\hat{x}}D(\hat{x})\|_2 - 1)^2\Big],$$

where λ is the penalty coefficient.

2.2 Housing GANs

The Housing GANs also requires training a generator G learned from a real data table T and generates a synthetic data table T_{syn} using the trained G. The real data table (T) may contain both continuous columns $\{C_1, \ldots, C_{N_c}\}$ (e.g., prices, distance to the city center, etc.) and discrete columns $\{D_1, \ldots, D_{N_d}\}$ (e.g., number of rooms, housing type, floor, heating type, etc.). In this setting, each column is assumed to be a random variable. The parameters N_c and N_d correspond to the number of continuous and discrete variables, respectively. These random variables have an unknown joint distribution denoted by $P(C_1 : N_c, D_1 : N_d)$. One row $r_j = \{c_{1,j}, \ldots, c_{N_c}, j, d_{1,j}, \ldots, d_{N_{d,j}}\}$, $j \in \{1, \ldots, n\}$, represents an observation from the joint distribution. After training G on T, T_{syn} is constructed by independently sampling rows using G. The efficiency of G is evaluated by comparing the distributional behavior of real data table T and synthetic data table T_{syn}.

Remark 1 Real data table T combines discrete and continuous data types. Therefore, to simultaneously produce both types of columns, GANs must utilize both the softmax and tanh functions on the output.

Remark 2 The real data contains multiple modes in its distribution, causing multimodal distributions, so the kernel density estimation method is utilized to determine the number of modes in the distribution.

Accurately representing the real/training data is crucial for successfully training neural networks. Discrete values can easily be represented as one-hot vectors. However, representing continuous values with complex distributions is a challenging task. In previous studies, the models usually normalized continuous values to the interval $[-1, 1]$ using the standard min-max normalization transform (Park et al., 2018). However, this study utilizes a mode-specific normalization approach to handle columns with complicated distributions, as in Xu et al. (2019). The study processes each column independently, representing each value as a combination of a one-hot vector indicating the mode and a scalar indicating the value within the mode. The method has three steps (Xu et al., 2019):

- For each continuous column C_i, it uses a variational Gaussian mixture model (VGM) (Bishop, 2006) to estimate the number of modes (m_i) and fits a Gaussian mixture distribution.
- For every $c_{i,j}$ in column C_i, it evaluates the probability of $c_{i,j}$ originating from every mode m_i.
- Randomly select one mode from the given probability density and use the selected mode to normalize the value.

Then, a row can be represented as the concatenation of continuous and discrete columns

$$r_j = \alpha_{1,j} \otimes \beta_{1,j} \otimes \cdots \otimes \alpha_{N_c,j} \otimes \beta_{N_c,j} \otimes d_{1,j} \otimes \cdots \otimes d_{N_d,j},$$

where $d_{i,j}$ denotes a one-hot representation of a discrete value.

Traditionally, the generator in a GAN is trained using a vector from a standard MVN distribution. When trained with a Discriminator or Critic neural network, a deterministic transformation maps the standard MVN to the data distribution. However, this method needs to account for imbalances in categorical columns, where under-represented minor categories can result in incorrect training of the generator. Resampling the training data can also lead to learning a different distribution rather than the real data distribution exacerbated by the "class imbalance" issue commonly found in discriminatory modeling. The challenge is further compounded as there is no single column to balance, and the real data distribution must be preserved. The objective is to perform resampling efficiently such that all categories of discrete attributes are sampled evenly (but not necessarily uniformly) during training while retaining the real data distribution during testing.

Remark 3 Let k^* be the value from i^*th discrete column D_{i^*} that has to be matched by the generated samples $\hat{r}$. Then, G can be interpreted as the conditional distribution of rows given that particular value at that particular column, i.e., $\hat{r} \sim \mathbb{P}_G(row \mid D_{i^*} = k^*)$.

The integration of a conditional generator into the architecture of a GAN involves addressing three key challenges: first, developing a representation for the condition and a corresponding input, second ensuring that the generated samples maintain the specified condition, and third teaching the conditional generator to learn the real data's conditional distribution, in order to reconstruct the original distribution Xu et al. (2019) introduces

$$\mathbb{P}(row) = \sum_{k \in D_{i^*}} \mathbb{P}_G(row \mid D_{i^*} = k^*)\mathbb{P}(D_{i^*} = k).$$

Definition 2 A vector called *cond* is the way to indicate the condition $(D_{i^*} = k^*)$. Recall that all discrete columns $D_1, \ldots, D_{N_d}$ end up as one-hot vectors $d_1, \ldots, d_{N_d}$ such that the ith one-hot vector is $d_i = [d_i^{(k)}]$, for $k = 1, \ldots, |D_i|$. Let $m_i = [m_i^{(k)}]$, for $k = 1, \ldots, |D_i|$ be the ith mask vector associated with ith one-hot vector d_i. Then, the condition can be expressed in terms of these mask vectors as

$$m_i^{(k)} = \begin{cases} 1, & \text{if} \quad i = i^* \text{ and } \ k = k^*, \\ 0, & \text{otherwise.} \end{cases}$$

Then, define the vector *cond* as $cond = m_1 \otimes \cdots \otimes m_{N_d}$. For instance, for two discrete columns, $D_1 = 1, 2, 3$ and $D_2 = 1, 2$, the condition $(D_2 = 1)$ is expressed by the mask vectors $m_1 = [0, 0, 0]$ and $m_2 = [1, 0]$; so $cond = [0, 0, 0, 1, 0]$.

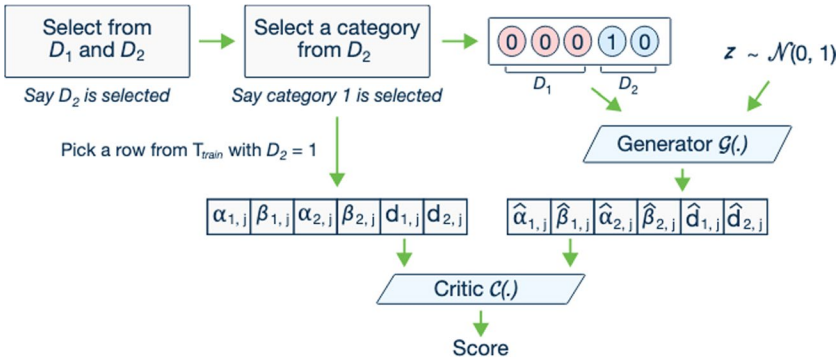

Fig. 2 The Housing GANs generates synthetic rows while considering one of the discrete columns as a condition. With training-by-sampling, the condition and training data are sampled based on the log frequency of each category, allowing Housing GANs to evenly cover all possible discrete values [gathered from (Xu et al., 2019)]

Table 1 A snapshot of the tabular data representing housing market data

Price	RegionID	RegionName	State	Metro	CountyName
773,584	6181	New York	NY	New York-Newark-Jersey City	Queens County
279,195	39051	Houston	TX	Houston-The Woodlands-Sugar Land	Harris County
306,835	17426	Chicago	IL	Chicago-Naperville-Elgin	Cook County
300,227	6915	San Antonio	TX	San Antonio-New Braunfels	Bexar County

During the training process, the conditional generator can generate any set of one-hot discrete vectors $\{\hat{d}_1, \ldots, \hat{d}_{N_d}\}$. Remarkably, considering the condition $(D_{i^*} = k^*)$ like *cond* vector form, nothing in the feed-forward pass prevents from outputting either $\hat{d}_{i^*}^{(k^*)} = 0$ or $\hat{d}_{i^*=1}^{(k)}$ for all $k \neq k^*$. The mechanism enforces the conditional generator to generate $\hat{d}_{i^*} = m_{i^*}$ is to penalize its loss by adding the cross entropy between m_{i^*} and $\hat{d}_{i^*}$, averaged over all instances of the batch. Therefore, while the training advances, G learns to mitigate the given m_{i^*} and $\hat{d}_{i^*}$.

The performance of the synthetic data generated by the conditional generator G must be evaluated by the critic (D), which measures the discrepancy between the learned conditional distribution $\mathbb{P}_G(row \mid cond)$ by G and the conditional distribution in the training data $\mathbb{P}(row \mid cond)$. To ensure an accurate estimation by the critic, it is crucial to properly sample the training data and construct the *cond* vector. Sampling the *cond* vector and training data in an appropriate manner can allow the model to equally explore all possible values in the discrete columns. The study follows these steps:

1. Build N_d zero-filled mask vectors $mi = [m_i^{(k)}]_{k=1,\ldots,|D_i|}$, for $i = 1, \ldots, N_d$, so the ith mask vector corresponds to the ith column, and each component is associated to the category of that column.

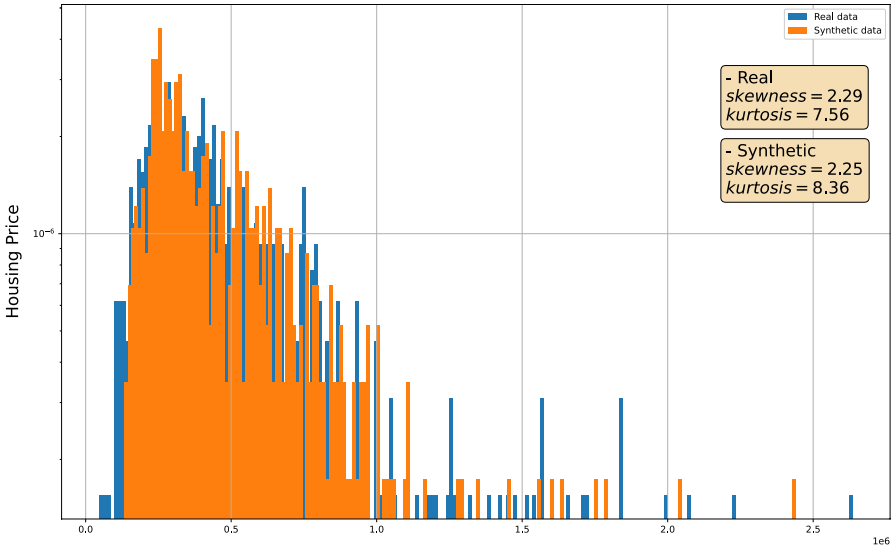

Fig. 3 Visualization of a Housing GANs realization along with the real data

Table 2 The descriptive statistics of the real and synthetic house price

	Real data	log(Real Data)	Synthetic data	log(Synthetic Data)
μ	4.960815e+05	12.933332	4.705569e+05	12.918963
σ	3.364630e+05	0.596959	2.850618e+05	0.518371
min	4.666900e+04	10.750835	1.334240e+05	11.801287
25%	2.790200e+05	12.539038	2.717808e+05	12.512748
50%	4.126010e+05	12.930236	3.862260e+05	12.864174
75%	6.090412e+05	13.319641	5.909738e+05	13.289525
max	2.634933e+06	14.784368	2.438491e+06	14.706890

2. Randomly select a discrete column D_i out of all N_d discrete columns, with equal probability. Let * be the index of the column selected (e.g., in Fig. 2, the selected column is D_2, thus $i^* = 2$).
3. Construct a probability mass function (PMF) across the range of values of the column selected in 2, D_{i^*}, such that the probability mass of each value is the logarithm of its frequency in that column.
4. Let k^* be a randomly selected value according to the PMF above (e.g., in Fig. 2, the range D_2 has two values, and the first one is selected as $k^* = 1$).
5. Set the k^*th component of i^*th mask to one, i.e., $m_{i^*}^{(k^*)} = 1$.
6. Compute the vector $cond = m_1 \otimes, \cdots, \otimes m_{i^*} \otimes m_{N_d}$ (e.g., in Fig. 2, the masks are $m_1 = [0, 0, 0]$ and $m_{2^*} = [1, 0]$; hence $cond = [0, 0, 0, 1, 0]$).

3 Empirical Analysis

To assess the accuracy of the Housing GANs, we retrieved the data from Zillow (1999). Table 1 presents a snapshot of the data used in the empirical analysis. The data includes house prices recorded on 31/12/2022 for various states, metros within those states, and counties within those metros. More precisely, the size of the training data is 4000×4.

In the training of Housing GANs, the Neural Networks uses the following hyperparameters: The number of Epochs is 500, the number of Batch size is 100, the dimensions of Generator and Discriminator are (256, 256, 256), the generator and discriminator learning rates are both $2e{-}4$, the discriminator weight decay for the AdAm optimizer is $1e{-}6$, and the number of discriminator updates to do for each generator update is 1. The gradient penalty of WGAN-GP is a function of $L2$-norm of the real data.

Figure 3 compares synthetic housing prices generated by Housing GANs and recorded housing price data. The figure graphs the logarithm of housing prices (y-axis) and the ranges or intervals of the logarithm of housing prices (x-axis) given in Table 1 to visualize better the empirical densities of the real and synthetic housing price data. It shows that the Housing GANs generates synthetic housing prices, which mimic the distributional behavior of the real housing prices. Even though the illustrated results are promising, the Housing GANs fail to capture some housing prices in the tails. On the other hand, the statistical measures of skewness and kurtosis values of the real and generated housing price are relatively close, which are also summarized in the figure. Hence, we can conclude that the Housing GANs learned the distributional behavior of the real housing price data and successfully generated synthetic housing prices. It is crucial to emphasize that the figure represents an illustration based on a single generated dataset. It is crucial to acknowledge that the Housing GANs have the potential to generate various synthetic datasets, some of which may demonstrate improved fidelity to the real data, while others may exhibit less accuracy. The figure serves as a demonstration of the general capabilities of the Housing GANs, but further analysis and evaluation are necessary to assess the overall performance and reliability of the synthetic data generated by the model.

Table 2 summarizes the basic statistical properties of the real and synthetic housing prices and their logarithmic values' statistical properties. As Fig. 3 also reveals, the realization of the synthetic data is not capturing the maximum and minimum housing prices. However, those values are relatively close to the real data. Furthermore, in the real data and synthetic data, the region and county of maximum

Table 3 The statistical significance of the difference between the real and synthetic data distributions

Test	Statistics	p value	Null hypothesis
KS	0.072	0.1497	The two distributions are identical
Student t	0.406	0.684	No difference in means of the real and synthetic data
Mann–Whitney U	128,357.0	0.4623	The distribution of the real and synthetic data the same

and minimum housing prices are consistent. The maximum and minimum housing prices are in Zionsville region of Yuma County, Arizona, and Abbeville region of Ada County, Idoha, respectively. Even though the minimum and maximum housing prices are mathematically not identical, the Housing GANs can generate synthetic data with minimum and maximum housing values in the same region. However, we should emphasize that the synthetic housing price data summarized here is a single realization. Hence, the Housing GANs may generate synthetic housing prices better or worse than this realization. We illustrated an example in "Appendix" to show this conclusion. More specifically, we graphed the empirical distribution of three realizations of Housing GANs in Fig. 4 along with their distributional statistics. Figure 4 shows that one may generate various realistic housing price scenarios using the Housing GANs for the housing market. However, each realizations will have different descriptive statistical while having identical distributional properties to the real data.

Figure 3 and Table 2 give some intuition about the distributional properties of the real and synthetic house prices. However, we should be more rigorous and assess the statistical significance of the difference between the distributions of these data sets. Hence, we compute the two-sample Kolmogorov–Smirnov (KS) test, student t test, and Mann–Whitney U-test to assess the statistical significance of the difference between these two distributions. Table 3 shows these tests' statistical results. The KS test indicates no significant difference between the real and synthetic data distributions, with a confidence level of 95% due to the p value being greater than 0.05. The student t test confirms this, as the p-value of 0.684 indicates no difference in the means of the real and synthetic data. The Mann–Whitney U-test also suggests that the two data sets have a comparable distribution shape, as the p-value of 0.4623 supports the null hypothesis. As a result, we can conclude that the synthetic data distribution is indistinguishable from the real data distribution, as none of these tests produced significant results.

4 Conclusion

The study presents GANs as a solution for modeling the distribution of columns with complex distributions and offers Housing GANs to generate synthetic housing market data sets that are distributionally identical to the real housing market data. The Housing GANs utilizes mode-specific normalization to transform continuous values into a compact vector representation that neural networks can process. The results demonstrate that neural networks can effectively be utilized within an adversarial framework to model housing market data with discrete and continuous variables. While training these models can be challenging, the advancements in GANs have produced promising results for modeling housing market data and hold the potential for even better performance in the future with more detailed datasets.

The Housing GANs have demonstrated their ability to generate realistic housing price data by closely capturing the distributional properties observed in the Zillow

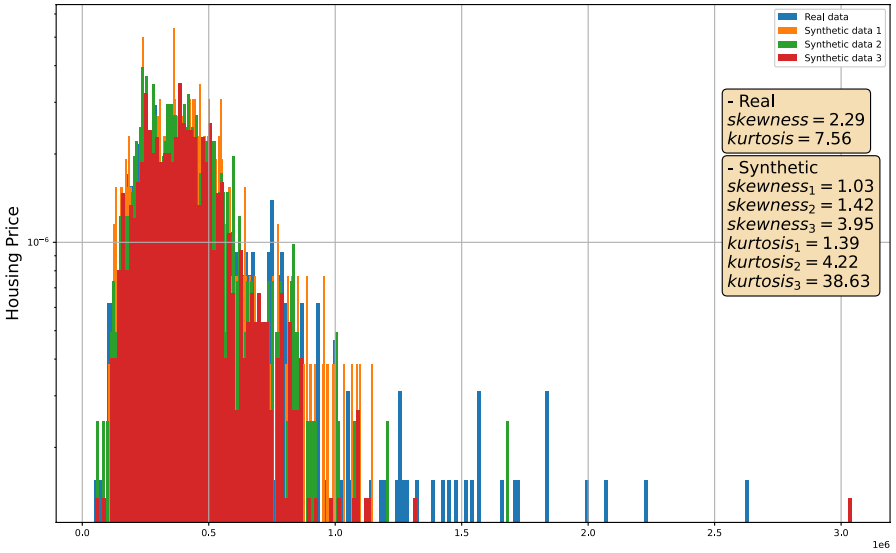

Fig. 4 Visualization of Housing GANs realizations along with the real data

housing data. While the synthetic data is not an exact replica of the Zillow housing data, it is generated from the same underlying distribution. This characteristic allows the Housing GANs to be utilized for scenario analysis by sampling from the learned distribution, effectively addressing the need for detailed housing data as described in the introduction. Moreover, the application of Housing GANs holds substantial benefits for companies and market regulators. By incorporating private data as inputs, the Housing GANs have the potential to enhance their efficiency and effectiveness in generating synthetic housing data. This integration of additional data sources, such as property characteristics, neighborhood characteristics, market conditions, time-series data, price data, and consumer preferences, can significantly augment the richness and accuracy of the synthetic data. Consequently, it enables stakeholders in the housing market to conduct more comprehensive and granular analyses, leading to valuable insights and informed decision-making.

The study's major limitation is the availability of detailed housing market data. With access to more comprehensive data, the performance of the Housing GANs is expected to improve, offering a data-driven approach that surpasses traditional financial models.

Appendix: Visualization of the Housing GANs Realizations

To show the accuracy of the Housing GANs, we visualized three realization of the synthetic house price and real house price data in Fig. 4. All three realizations had housing prices close the real housing prices. The third realization captured

the housing price in both left and right tails. However, it has the highest kurtosis value while the first realization has the lowest skewness value.

Funding The author declares that no external funding was received for this research. The study was conducted without financial support from any organization or institution.

Declarations

Conflict of interest The author declares that there are no conflicts of interest or competing interests associated with this manuscript. I confirm that I have no financial or personal relationships with individuals or organizations that could inappropriately influence or bias the content of this work.

Ethics approval This study does not involve the use of human or animal related data. Therefore, no ethics approval was required for this research. The study strictly adheres to ethical guidelines and regulations regarding the handling and analysis of non-human, non-sensitive data.

References

Al-Homoud, M., Al-Oun, S., & Al-Hindawi, A. M. (2009). The low-income housing market in Jordan. *International Journal of Housing Markets and Analysis, 2*, 233–252.

Arjovsky, M., Chintala, S., Bottou, L. (2017). Wasserstein generative adversarial networks. In: International conference on machine learning, PMLR, pp 214–223

Bendaoud, N. M. M., Farah, N., & Ahmed, S. B. (2021). Comparing generative adversarial networks architectures for electricity demand forecasting. *Energy and Buildings, 247*(111), 152.

Bishop, C. M. (2006). *Pattern recognition and machine learning (information science and statistics)*. Berlin: Springer.

Boyle, M., & Kiel, K. (2001). A survey of house price hedonic studies of the impact of environmental externalities. *Journal of Real Estate Literature, 9*(2), 117–144.

Chen, J. H., Ong, C. F., Zheng, L., et al. (2017). Forecasting spatial dynamics of the housing market using support vector machine. *International Journal of Strategic Property Management, 21*(3), 273–283.

Chen, Y., Wang, Y., Kirschen, D., et al. (2018). Model-free renewable scenario generation using generative adversarial networks. *IEEE Transactions on Power Systems, 33*(3), 3265–3275.

Clapp, J. M., & Giaccotto, C. (1998a). Price indices based on the hedonic repeat-sales method: Application to the housing market. *The Journal of Real Estate Finance and Economics, 16*, 5–26.

Clapp, J. M., & Giaccotto, C. (1998b). Residential hedonic models: A rational expectations approach to age effects. *Journal of Urban Economics, 44*(3), 415–437.

Colwell, P. F., & Dilmore, G. (1999). Who was first? An examination of an early hedonic study. *Land Economics*, *75*(4), 620–626.

Coulson, N. E., & Bond, E. W. (1990). A hedonic approach to residential succession. *The Review of Economics and Statistics*, *72*(3), 433–444.

Esteban, C., Hyland, S. L., Rätsch, G. (2017). Real-valued (medical) time series generation with recurrent conditional GANs.

Fekri, M. N., Ghosh, A. M., & Grolinger, K. (2019). Generating energy data for machine learning with recurrent generative adversarial networks. *Energies, 13*(1), 130.

Galteri, L., Seidenari, L., & Bertini, M., et al. (2019). Towards real-time image enhancement GANs. In *International conference on computer analysis of images and patterns* (pp. 183–195). Springer.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., et al. (2014). Generative adversarial networks. *Advances in Neural Information Processing Systems*, 2672–2680.

Gu, J., Zhu, M., & Jiang, L. (2011). Housing price forecasting based on genetic algorithm and support vector machine. *Expert Systems with Applications, 38*(4), 3383–3386.

Gu, Y., Chen, Q., Liu, K., et al. (2019). Gan-based model for residential load generation considering typical consumption patterns. In *2019 IEEE power & energy society innovative smart grid technologies conference (ISGT)* (pp. 1–5). IEEE.

Gulrajani, I., Ahmed, F., Arjovsky, M., et al. (2017). Improved training of Wasserstein GANs. *Advances in Neural Information Processing Systems, 30*.

Huang, B., Wu, B., & Barry, M. (2010). Geographically and temporally weighted regression for modeling spatio-temporal variation in house prices. *International Journal of Geographical Information Science, 24*(3), 383–401.

Khamis, A. B., & Kamarudin, N. (2014). Comparative study on estimate house price using statistical and neural network model. *International Journal of Scientific & Technology Research, 3*(12), 126–131.

Korn, R., & Yilmaz, B. (2022). House prices as a result of trading activities: A patient trader model. *Computational Economics, 60*(1), 281–303.

Meen, G. (2011). A long-run model of housing affordability. *Housing Studies, 26*(7–8), 1081–1103.

Park, N., Mohammadi, M., Gorde, K., et al. (2018). Data synthesis based on generative adversarial networks. *Proc VLDB Endow, 11*(10), 1071–1083.

Puri, A. K., & Van Lierop, J. (1988). Forecasting housing starts. *International Journal of Forecasting, 4*(1), 125–134.

Radford, A., Metz, L., Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434

Raj, J. S., Ananthi, J. V., et al. (2019). Recurrent neural networks and nonlinear prediction in support vector machines. *Journal of Soft Computing Paradigm (JSCP), 1*(01), 33–40.

Silva, V. L. S., Heaney, C. E., Li, Y., et al. (2021). Data assimilation predictive GAN (DA-PredGAN): Applied to determine the spread of COVID-19. CoRR abs/2105.07729. https://arxiv.org/abs/2105.07729.

Stevenson, S. (2008). Modeling housing market fundamentals: Empirical evidence of extreme market conditions. *Real Estate Economics, 36*(1), 1–29.

Tian, C., Li, C., Zhang, G., et al. (2019). Data driven parallel prediction of building energy consumption using generative adversarial nets. *Energy and Buildings, 186*, 230–243.

Van Leuvensteijn, M., & Koning, P. (2004). The effect of home-ownership on labor mobility in the Netherlands. *Journal of Urban Economics, 55*(3), 580–596.

Wang, Z., & Hong, T. (2020). Generating realistic building electrical load profiles through the Generative Adversarial Network (GAN). *Energy and Buildings, 224*(110), 299.

Wiese, M., Knobloch, R., Korn, R., et al. (2020). Quant GANs: Deep generation of financial time series. *Quantitative Finance, 20*(9), 1419–1440.

Wu, J., Zhang, C., Xue, T., et al. (2016). Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in Neural Information Processing Systems, 29*.

Xu, L., Skoularidou, M., Cuesta-Infante, A., et al. (2019). Modeling Tabular data using Conditional GAN. In *Advances in neural information processing systems*.

Yilmaz, B. (2021). Understanding the mathematical background of generative adversarial neural networks (GANs). Available at SSRN 3981773.

Yilmaz, B., & Korn, R. (2022). Synthetic demand data generation for individual electricity consumers: Generative Adversarial Networks (GANs). *Energy and AI, 9*(100), 161.

Yilmaz, B., & Selcuk-Kestel, A. S. (2018). A stochastic approach to model housing markets: The US housing market case. *Numerical Algebra Control and Optimization, 8*(4), 481–492.

Yilmaz, B., & Selcuk-Kestel, A. S. (2020). Forecasting house prices in Turkey: GLM, VaR and time series approaches. *Journal of Business Economics and Finance, 9*(4), 274–291.

Yilmaz, B., Hekimoglu, A. A., & Selcuk-Kestel, A. S. (2022a). Default and prepayment options pricing and default probability valuation under VG model. *Journal of Computational and Applied Mathematics, 399*(113), 724.

Yilmaz, B., Korn, R., Selcuk-Kestel, A. S. (2022b). The impact of large investors on the portfolio optimization of single-family houses in housing markets. *Computational Economics* 1–19.

Yu, L., Zhang, W., Wang, J., et al. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*.

Yuan, R., Wang, B., Mao, Z., et al. (2021). Multi-objective wind power scenario forecasting based on PG-GAN. *Energy, 226*(120), 379.

Yusof, A. M., & Ismail, S. (2012). Multiple regressions in analysing house price variations. *Communications of the IBIMA, 2012*, 1.

Zhou, D., Ma, S., Hao, J., et al. (2020). An electricity load forecasting model for Integrated Energy System based on BiGAN and transfer learning. *Energy Reports, 6*, 3446–3461.

Zhu, J. Y., Park, T., Isola, P., et al. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223–2232).
Zillow. (1999). Zillow home value forecast (zhvf). https://www.zillow.com/.