Expert Systems With Applications 264 (2025) 125838

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

GAN-based synthetic time-series data generation for improving prediction of demand for electric vehicles

Subhajit Chatterjee [a], Debapriya Hazra [a], Yung-Cheol Byun [b,*]

[a] Department of Computer Engineering, Jeju National University, Jeju 63243, South Korea
[b] Department of Computer Engineering, Major of Electronic Engineering, Jeju National University, Institute of Information Science & Technology, Jeju 63243, South Korea

ARTICLE INFO

Keywords:
Machine learning
Time-series data
Demand prediction
Shared electric kickboards
Synthetic data
Generative adversarial networks
WGAN-GP
CGAN
Regression

ABSTRACT

Demand forecasting is essential for any business to grow and manage its different business activities. With the basic needs of customers, it is hard to predict the future demand using traditional techniques. The popular approach to overcoming the difficulties faced by startup businesses is to use machine learning techniques for demand prediction. Another constraint to train machine learning algorithms for accurate prediction requires a considerable amount of data. For a new startup business, vast data acquisition is a very problematic issue. To overcome the data scarcity problem data enhancement techniques are mainly used for expanding the existing data. Synthetic data generation to balance the existing data which can lead to increased prediction model accuracy. In this study at the beginning, we found that the accuracy of the proposed clustering-based ensemble regression model was bad because of the small size of the data. To overcome this issue, we proposed a modified Conditional Wasserstein Generative Adversarial Network with a Gradient Penalty (CWGAN-GP) for generating synthetic time-series data according to the original data distribution. This generated synthetic data was further added to the original data to train the model and additionally enhanced the demand prediction accuracy for shared electric kickboards. Improved performance was noticed after the model was trained with combined data. Using a range of evaluation measures and graphical representations, we evaluated the performance of our approach against that of other ensemble models. For the production of synthetic data, our GAN model converged more quickly than other GAN models and solved the mode collapse problem. We have contrasted our suggested approach with other cutting-edge models. This study can be helpful for companies to meet the user's demand for a better quality of service.

1. Introduction

The urban transportation system is undergoing a significant transformation. Shared electric kickboards (EVs) transportation services are becoming a more significant competitor to traditional public transportation modes like buses, taxis, and cabs. This rise in popularity is in line with the recent expansion of shared electric kickboard programs. These systems are a strong substitute for conventional modes of transportation, giving city residents a flexible and environmentally friendly choice for getting around. This approach could have several positive social effects, such as a decline in the use of automobiles, a reduction in greenhouse gas emissions, and an improvement in traffic congestion, particularly in urban areas. Shared electric kickboards encourage a more eco-friendly and effective urban transportation system by providing a convenient carbon-free option and encouraging smart riding options. A complete approach to reducing air pollution is to deploy shared electric kickboards (EVs). Reports indicate that global EV sales in 2018 increased 72% over 2017, and market share increased to 2.1% (Amirkhani et al., 2019). With the growth of the global low-carbon movement and the rise in the number of private cars, Jeju province is planning to shift the focus to electric vehicles, mainly because traffic congestion and environmental pollution problems are becoming more and more serious concerns, and shared electric kickboards are as the new public transportation system to flourish in the world. The shared electric kickboards can provide short-distance travel play the flexible and efficient driving advantages (Hong et al., 2016). To thrive in the transportation industry, a developing enterprise must contend with rivals offering shared electric kickboard services. The business must provide better customer service and match demand if it wants to become more well-known. For a more sophisticated approach, they must use AI techniques to forecast client demand in advance and create a system that will deliver that service. Demand

* Corresponding author.
E-mail addresses: subhajitchatterjee@stu.jejunu.ac.kr (S. Chatterjee), debapriyah@jejunu.ac.kr (D. Hazra), ycb@jejunu.ac.kr (Y.-C. Byun).

https://doi.org/10.1016/j.eswa.2024.125838
Received 19 April 2024; Received in revised form 19 July 2024; Accepted 16 November 2024
Available online 23 November 2024

forecasting aids businesses in enhancing sales and productivity (Hulot et al., 2018). Human needs are endless, driving constant innovations across industries. Our research highlights the imperative for sustainable urban mobility, prompting a transition from conventional to electronic transportation modes. This surge in shared electric kickboard demand underscores the importance of efficient data processing and accurate forecasting for sustaining market competitiveness and profitability. Demand forecasting is looking at data that has been created in the past and analyzing different variables that are related to the goal variable. This is what we can use to create a cutting-edge predictive model that predicts future demand—a process known as demand forecasting.

For accurate demand forecasting a machine-learning model needs a considerable amount of data to train the model. One of the concerns in this field is the data imbalance issue, the most challenging problem in the machine learning domain. As a result, various oversampling strategies have been implemented to address data imbalance issues in multiple domains. The most simplistic approach, known as random oversampling methods, involves duplicating instances of minority class instances to increase the data size and balance the data labels. Before training the model, many studies have employed different iterations of the random oversampling approach. Time-series datasets, for instance, have few data points and wildly unbalanced data distributions (García-Jara et al., 2022). Imbalanced and small datasets make learning difficult because algorithms struggle to generalize data properties, and uneven distributions worsen the problem. These two qualities make the task a unique challenge for machine learning algorithms. This issue will be solved by data augmentation techniques routinely used to convert small, unbalanced datasets into balanced ones. As a result, time-domain augmentation techniques are still difficult and require more community effort (Wen et al., 2020). Jittering, dynamic window warping, dynamic time warping, and slicing are examples of traditional time-domain augmentation techniques that assume that these transformations occur naturally in the data and that the augmented samples will be valid time series with comparable properties to the original ones. GAN (Generative Adversarial Networks) are increasingly popular as a substitute strategy for handling imbalanced data. The GAN technique is used to generate synthetic data that enables the overcome the issue of data imbalanced issues. Since the generated samples will be used for regression, the generative model needs to understand how the data are distributed under different conditions. As a result, the model can simultaneously learn how to conserve data information and generate realistic samples.

In the study, authors (Mogren, 2016) proposed a model that combines RNN with GAN that is trained with adversarial training to model the entire joint probability of a sequence and to be able to generate data sequences because GAN plays a significant role in the field of time-series domain for solving the data imbalance problem. Esteban et al. (2017) coupled recurrent GAN (RGAN) and recurrent conditional GAN (RCGAN), adhering to the RNN idea, to create realistic real-valued multi-dimensional time series, placing particular emphasis on its use with medical data. A suitable generative model should preserve the temporal dynamics of time-series data because new sequences maintain the original associations between variables throughout time. To capture the time-series conditional temporal dynamics, Yoon et al. (2019) developed a jointly trained embedding network that blends a supervised autoregressive model with an unsupervised GAN framework. In the study, Fu et al. (2019) conditional GAN to learn time series data and simulate the synthetic data, where the condition would be both dependent and continuous variables containing different kinds of auxiliary information. Utilizing GAN as an oversampling technique, artificial data can be produced to help identify credit card fraud. The author (Ba, 2019) used WGAN and conditional GAN to stabilize the training to generate synthetic data for fraudulent transactions. Guo et al. (2021) created an end-to-end framework called ITCGAN for imbalance traffic classification data that can generate traffic samples for minority classes to adaptively rebalance the original traffic and train the best classifier at the same time. According to the article by Koivu et al. (2020), actGAN (activation-specific generative adversarial network), which can gather useful synthetic observations on data in terms of boosting prediction performance, was proposed as a solution to the minority oversampling ratio between the classes on unbalanced data. Li et al. (2018) introduced CS-GAN, a model combining RNN and GAN, to generate category sentences for dataset expansion. Tang et al. (2018) proposed a method using the Auxiliary Classifier GAN for programmatic data augmentation. Unlike image recognition or speech processing, building energy consumption datasets are often limited in size (Torres et al., 2017). GANs offer an effective approach for augmenting data and predicting building energy consumption (Grolinger et al., 2016), particularly in scenarios like sensor-based forecasting for event venues with fluctuating consumption patterns. In the study (Martínez-Álvarez et al., 2015), the application of these techniques to time series forecasting. To extract patterns from time-series datasets, Pérez-Chacón et al. (2016) proposed a pattern discovery strategy that uses the distributed version of the k-means algorithm within the Apache Spark framework.

It is difficult to meet customer needs when there is a shortage of both quantitative electric kickboards in the required station. A shared electric kickboard service provides its customers with unlimited access to use the service whenever and wherever they want. Another point is that external factors like weather conditions, calendar variables, etc drive shared electric kickboards. The project presents a study of different variables and their effect on the rental behavior of shared electric kickboards. The methods impact forecast shared electric kickboards availability in the station to ensure that the customer has just enough shared electric kickboards at the right time. The main contribution of this study is as follows:

- We have developed a synthetic data generation approach based on Generative Adversarial Networks (GAN) to overcome the limitations of small and imbalanced time-series data. We employed a modified version of the CWGAN-GP architecture to create a realistic synthetic time series that will be further combined with original data to train the machine learning model, ultimately enhancing the prediction performance of the proposed model.
- Our paper investigates the impact of spatial granularity on prediction accuracy in rental demand analysis. Recognizing that different zones within the dataset exhibit distinct rental patterns, we segmented the data into two and four sectors at the outset of our experiments. This segmentation allows for a detailed examination of how varying spatial detail levels affect our predictive models' accuracy.
- We propose a clustering-based ensemble regression blending technique to predict future demand for shared electric kickboards.
- To assess the effectiveness of synthetic data augmentation, we compared the prediction accuracy of our model with and without synthetic data, employing both graphical and statistical methods, and discussed the findings in the results section.

Finally, the paper is ordered as follows: Section 2 covers detailed methodology with information about the data set with preliminary data analysis, data cleaning, preprocessing, and step-by-step process of our experiment. Section 3 deals with all the machine learning models implemented are illustrated with their results in this section with the evaluation metrics used for testing the performance of the predictive models. Section 4 concludes with the conclusions.

2. Methodology

This section outlines the step-by-step procedure of our methodology for applying a data augmentation method to address the small and unbalanced data problem and investigate the impact of synthetic data on prediction performance. A detailed overview of our experiment

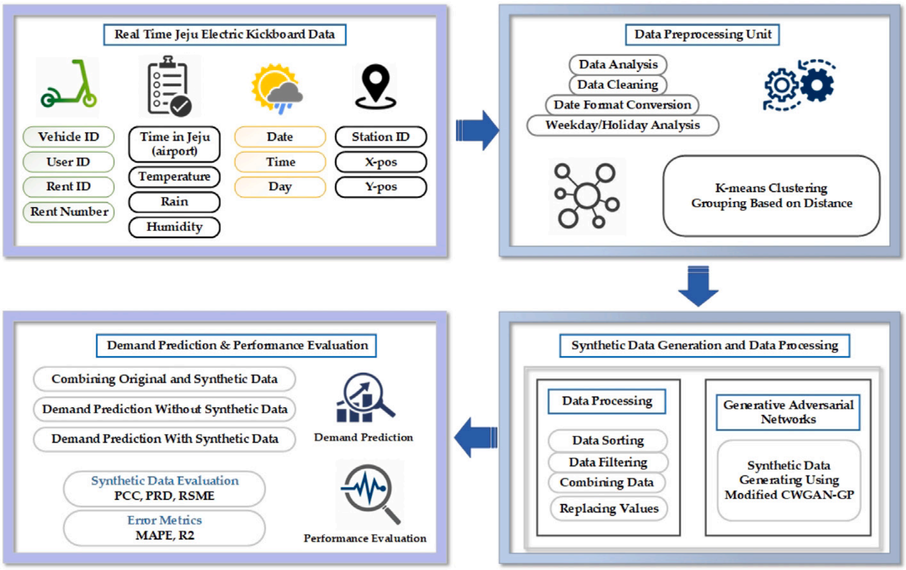

Real Time Jeju Electric Kickboard Data
Vehicle ID
User ID
Rent ID
Rent Number
Time in Jeju (airport)
Temperature
Rain
Humidity
Date
Time
Day
Station ID
X-pos
Y-pos
Data Preprocessing Unit
Data Analysis
Data Cleaning
Date Format Conversion
Weekday/Holiday Analysis
K-means Clustering Grouping Based on Distance
Synthetic Data Generation and Data Processing
Data Processing
Data Sorting
Data Filtering
Combining Data
Replacing Values
Generative Adversarial Networks
Synthetic Data Generating Using Modified CWGAN-GP
Demand Prediction & Performance Evaluation
Combining Original and Synthetic Data
Demand Prediction Without Synthetic Data
Demand Prediction With Synthetic Data
Demand Prediction
Synthetic Data Evaluation
PCC, PRD, RSME
Error Metrics
MAPE, R2
Performance Evaluation

Fig. 1. Workflow of the proposed framework.

Table 1
Structure and description of extracted input data.

Data field	Feature	Data type	Data format	Data description
Local timestamp	Date and Time	Datetime	yyyy-mm-dd hh:mm	Details of kickboard hiring
T	Temperature	Float	Celsius or Fahrenheit	The area's temperature
H	Humidity	Float	Percentage	Humidity in that region
WW	Rain	Float	Place it on a scale	Information about rain
rent_number	Rent_ID	Integer	Number of rent	Number of rented vehicles
rent_station	Station_ID	Varchar	Predefined string code with number	Vehicle unique ID number
day	Code for holiday	String (category)	Day	0 (Weekday), 1 (Weekend)
Xpos	Latitude	Float	Degree	Latitude angles
Ypos	Longitude	Float	Degree	Longitude angles

is provided in Fig. 1. We collected data from four different sectors within the company and processed it further. During the preprocessing stage, we removed underlying outliers and applied data cleaning and format conversion, ensuring the data structure remained unmodified. The preprocessed data was then analyzed to select relevant features. Following preprocessing and feature selection, we segmented the regions using the k-means clustering technique. The segmented data was then input into the proposed GAN model to generate synthetic data. We trained the proposed ensemble regression blending model with and without the augmented data, documenting the impact of synthetic data in the results section. Finally, we evaluated the performance, assessing the significance of the synthetic data and the effectiveness of demand prediction accuracy systems utilizing this augmented data.

2.1. Dataset and data preprocessing

The dataset contains real-time information on shared electric kickboard data from April 2019 to June 2021. We received four separate CSV files at the beginning of our experiment. Initially, we obtained data from various sources including the company's kickboard records, spatial coordinates, temporal details, and weather updates from the weather authority of Korea. The kickboard data encompasses vehicle IDs, user IDs, rental IDs, and rental numbers, each with its unique specifications. Spatial data encompasses station IDs alongside their corresponding x and y coordinates, while temporal data provides information on dates and times. Moreover, the raw meteorological data comprises timestamps along with temperature, humidity, and precipitation measurements. Refer to Table 1 for a detailed breakdown of the data utilized in our experiment.

We have obtained the input data from April 16, 2019, to June 11, 2021. We have combined four data source files; the final data consists of 11971 records. We incorporated supplementary criteria before finalizing the dataset, including factors like the average daily temperatures and precipitation. Our primary objective in this paper is to predict demand accurately. The second solution tackles the problem of imbalanced data within the dataset by implementing oversampling techniques and generating synthetic data using a GAN approach. We invested considerable effort into analyzing and preprocessing the data to uncover valuable and potentially instructive insights from the dataset. Fig. 2 portrays the demand according to the date and number of rent obtained. The graph's X-axis denotes the rental date, while the y-axis denotes the rental count.

The company's raw datasets have various characteristics. These traits were selected based on their significance and potential use in predicting kickboard demand. The data preparation involved several stages, and the data was meticulously analyzed for further preprocessing steps. Initially, 13 features were chosen for the experiment. In machine learning models, feature significance scores measure each feature's contribution to the model's prediction performance. Different methodologies are used to obtain these scores, depending on the model type. In tree-based models like Random Forest and XGBoost, feature importance is determined by the improvement each feature contributes to the splitting criterion (such as Gini impurity, entropy, or mean squared error) across all trees in the ensemble. The LightGBM (LGBM) technique was used to construct the feature importance scores shown in Fig. 3, as it automatically generates feature importance metrics. Specifically, the LGBMRegressor model was employed. The Gain

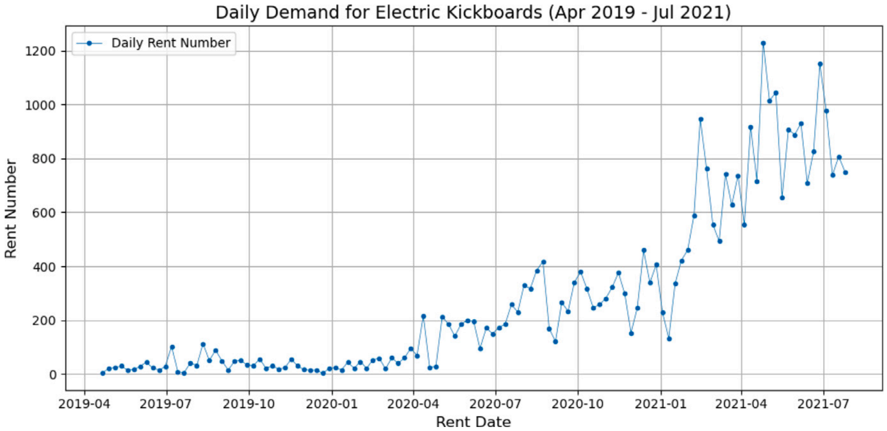


Fig. 2. Daily demand for shared electric kickboards (Arp 2019–Jul 2021), this graph illustrates the daily rental numbers for shared electric kickboards.

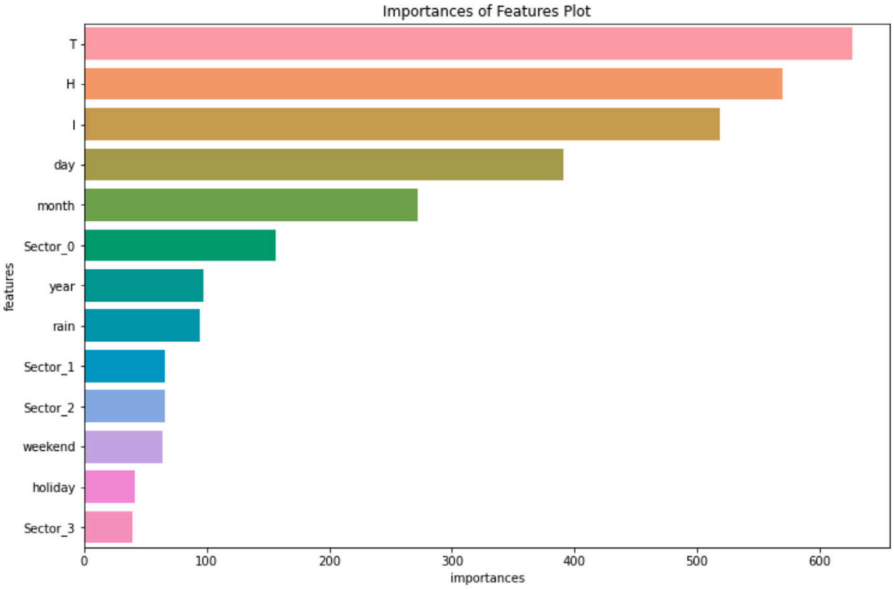


Fig. 3. Feature importance scores showing the contribution of each feature to the prediction performance.

criterion was used to generate the feature importance scores, indicating how much each feature contributes to the model's overall prediction accuracy. The LightGBM library includes functions for quickly extracting and visualizing these importance scores after training the model. For clarity, temperature is denoted by 'T', insulation by 'I', and humidity by 'H'. We also considered additional features such as day of the week, month, year, weekend indicators, rain events, and holidays. The feature importance scores are visualized on the graph's *x*-axis, while the corresponding feature names are listed on the *y*-axis. During data preprocessing, Saturday and Sunday were designated as weekends.

For separating the clusters, we have employed the k-means clustering algorithm. Unsupervised learning algorithm k-means clustering divides the unlabeled dataset into various groupings. Here, *k* specifies how many pre-defined clusters must be produced as part of the process; for example, if k = 2, there will be two clusters; if k = 3, there will be three clusters, and so on. In our study, we have applied k = 2 to get the two-sector data and k = 4 to get the four-sector data. The algorithm k-means was introduced by Hartigan and Wong (1979), an iterative technique that separates the unlabeled dataset into k distinct clusters, with each dataset belonging to just one group with identical characteristics. Fig. 4 displays the two-sector and four-sector data distribution before generating synthetic data. The red mark depicts the number of clusters, the latitude represents the rent_number of the specific sector, and the longitude represents the sector number.

2.2. *Synthetic data generation and enhancement using modified CWGAN-GP*

This section will discuss the synthetic data generated using the proposed modified CWGAN-GP model. To generate synthetic data and increase the data to enhance the performance of prediction accuracy. We have implemented modified CWGAN-GP architecture that helps us explicitly read the original data and generate the data according to the original data distribution, i.e., rent_number that has been less and pulling down the prediction accuracy to decrease the error rate.

2.2.1. *Overview of basic GAN architecture*

The basic idea behind the GAN is to set up a min–max game between two networks, first called the generator and another one the discriminator, introduced by Goodfellow et al. (2014) in 2014. The generator creates samples intended to come from the same distribution as the training data. The other network is the discriminator. The discriminator examines samples to determine whether they are real or fake means generated by the generator. The discriminator learns using traditional supervised learning techniques, dividing inputs into two classes (real or fake). Fig. 5 portrays the basic GAN framework.

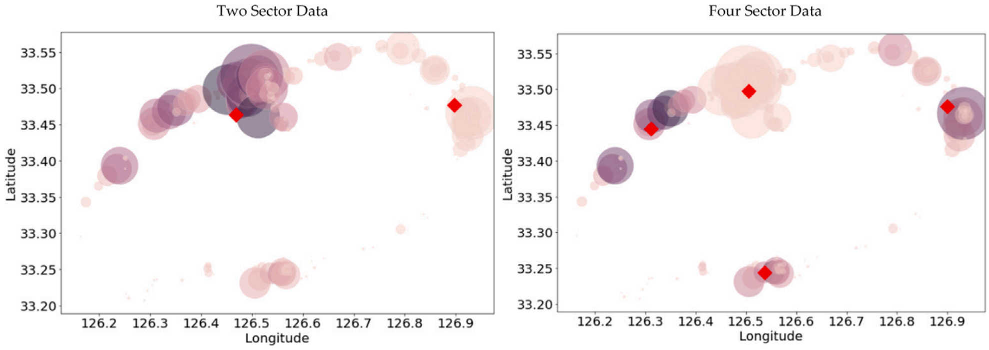


Fig. 4. Original data visualization of two-sector and four-sector data.

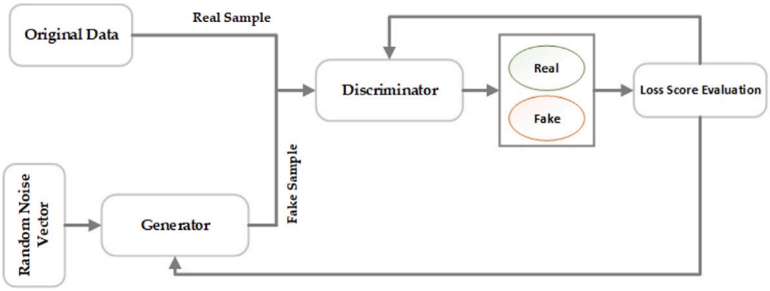


Fig. 5. Overview of basic GAN architecture.

A GAN consists of a generator G, tasked with learning the distribution of the data x using a random noise variable z. This learning process involves defining a prior P_g on the random noise variable. While the original GAN utilized a uniform distribution for P_g, recent studies commonly employ a Gaussian distribution. The generator $G(z;\theta g)$ employs a mapping function from z to the data space, with θg representing the parameters of a neural network. Additionally, a discriminator $D(x;\sigma_d)$ is constructed to assess the likelihood that input data x originated from the real data distribution P_r rather than the distribution produced by the generator $G(z)$. The discriminator's parameters are denoted by σ_d. Overall, the GAN framework involves a min–max game, as defined by Eq. (1), where P_g represents the prior noise, P_g represents a distribution of data produced by the generator, and P_r denotes the real data distribution.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_r}[log D(x)] + \mathbb{E}_{z\sim P_g}[log(1-D(G(z)))] \tag{1}$$

Where the objective behind this game is that the generator is trying to minimize the discriminator's reward or, in other words, maximize its loss. More precisely, generator G tries to fool the discriminator D into thinking its fake generated data is real. Thus, $D(x)$ is the output of the discriminator for a real input x, and $D(G(z))$ is the output of the discriminator for a fake generated data $G(z)$.

One issue researchers run across when attempting to use machine learning algorithms on time series data is overfitting. In a way, we could teach a model to generate new data that produced alternative realistic time series that share the same properties as the original data. Since this particular class of models has solid theoretical foundations and significantly enhances training stability, we adopted a Wasserstein GAN (WGAN) in our work. Additionally, the loss is correlated with the generator's convergence and the samples' quality, which is very advantageous since it eliminates the need for researchers to continuously review the samples generated to determine whether the model is improving. Finally, WGANs outperform ordinary GANs in terms of resistance to mode collapse and architectural alterations. A variant of the original WGAN model, the Wasserstein GAN with Gradient Penalty (WGAN-GP), was introduced by Gulrajani et al. (2017). To implement this upgraded version, one must compute the gradient penalty and modify the loss function to incorporate the Wasserstein distance. Since weight clipping can lead to capacity issues and necessitates the inclusion of additional parameters to specify the space in which the weights lie, WGAN-GP instead utilizes a "gradient penalty" (GP) technique. The discriminator's batch normalization needs to be changed to accommodate layer normalizing. The spectral normalization (SN) method has been utilized for routine training. Spectral Normalization (SN) essentially guarantees K-Lipschitz continuity of the discriminator (D). This is accomplished by methodically lowering the critic's Lipschitz constant on each of its layers. By developing a strong discriminator that produces a relevant gradient for the generator while ignoring its subpar performance, WGAN seeks to maximize the Wasserstein-1 distance. Eq. (2) illustrates how the Wasserstein-1 distance can be defined with a small change in the GAN objective.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_r}[D(x)] - \mathbb{E}_{z\sim n_z}[D(G(z))] \tag{2}$$

The discriminator's batch normalization needs to be changed to accommodate layer normalizing. The spectral normalization (SN) (Miyato et al., 2018) method has been utilized for routine training. In essence, SN ensures that D maintains K-Lipschitz continuity. It achieves this by progressively imposing constraints on the Lipschitz constant of your critic. The GP technique, on the other hand, offers more frequent training with essentially no hyper-parameter adjustment. Instead of clipping, it suggests a gradient penalty to satisfy the 1-Lipschitz requirement. Depending on how distant from 1-Lipschitz the gradient is, WGAN-GP either increases or lessens the gradient penalty. The change of loss functions of WGAN-GP that would combine the original critic loss and gradient penalty loss is defined as Eqs. (3) and (4).

$$OriginalCritic_{Loss} = \underbrace{\mathbb{E}_{x\sim P_r}[D(x)] - \mathbb{E}_{\tilde{x}\sim P_g}[D(G(\tilde{x}))]} \tag{3}$$

$$GradientPenalty_{Loss} = \underbrace{\lambda \cdot \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]} \tag{4}$$

The gradient norm for random samples, $\hat{x} \sim P_{\hat{x}}$, between pairs of points sampled from the data distribution P_r and the generator distribution P_g, $P_{\hat{x}}$ sampling is conducted uniformly along straight lines. λ has been set to 10. This has been empirically proven to work well with a wide range of architectures and datasets (Iwana & Uchida, 2021).

To generate new data, time-series data must undergo further processing. In some circumstances, we might want to regulate the column values to force the model to produce data of a particular data type. CGAN (Douzas & Bacao, 2018; Mirza & Osindero, 2014) performs the conditional data generation, which conditions the generator to produce a specified output. The order of the rows matters for the inter-row relationships present in the time-series datasets. A column with sortable values like integers, floats, or date times will likely serve as the indicator for this order. Before training the GAN model, we constrained the data about rent station IDs generated by the GAN to prevent the generation of new IDs.

In the Conditional GAN framework (CGAN), both the generator and discriminator receive additional information y. This information, which could be a class label or any other supplementary data, is denoted as y. Despite the inclusion of this additional information, the training process for CGAN remains identical to that of traditional GANs. The objective function is depicted in Eq. (5).

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_r}[D(x \mid y)] - \mathbb{E}_{z\sim P_g}[D(G(z \mid y))] \tag{5}$$

In CGAN, both P_r and P_g serve as inputs to the hidden layer, coupled with prior noise. Here, P_r and P_g retain the same definitions as in traditional GANs. The optimization process in CGAN mirrors that of GAN, ensuring comparability in performance.

To elaborate on the rationale behind creating WGAN-GP, it prepares the way for CWGAN-GP if the discriminator and generator in the CGAN version of GAN depend on extra auxiliary data. In this case, the variable y might stand for several kinds of data; in our study, y stands for the class label. While y is concatenated with $p(z)$ within the same representation in the generator, P_r and P_g are combined with y to generate a single hidden representation within the discriminator. Formally, Eq. (6) defines the minimax objective function between the generator and the discriminator.

$$\begin{aligned}\min_G \max_D V(D,G) = {} & \mathbb{E}_{x\sim P_r}[D(x \mid y)] - \mathbb{E}_{\tilde{x}\sim P_g}[D(G(\tilde{x} \mid y))] \\ & + \lambda \cdot \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x} \mid y)\|_2 - 1)^2]\end{aligned} \tag{6}$$

where $\hat{x}$ the sampling mode is the same as for WGAN-GP and is once more the gradient penalty coefficient λ. The proposed CWGAN-structure GP's and the loss functions of the discriminator and generator are minimized as following Eq. (7), (8).

$$D_{Loss} = -\mathbb{E}_{x\sim P_r}[D(x \mid y)] + \mathbb{E}_{\tilde{x}\sim P_g}[D(G(\tilde{x} \mid y))] + \lambda \cdot \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x} \mid y)\|_2 - 1)^2] \tag{7}$$

$$G_{Loss} = -\mathbb{E}_{\tilde{x}\sim P_g}[D(G(\tilde{x} \mid y))] \tag{8}$$

2.2.2. Synthetic data preprocessing

Following the creation of artificial time-series data utilizing the proposed GAN model, further preprocessing steps are imperative for forecasting demand. Generating synthetic data mirroring the distribution of the original data required 35 min with GPU processing. Essential steps include sorting, filtering, data combination, and value replacement. Employing a selection technique ensures alignment of composite dates with those in the original dataset, followed by the merging of original and synthetic data generated by the GAN model. Moreover, we replaced 0 values with 1 and converted negative values to their absolute counterparts in the target variable *rent_number*.

2.2.3. Regression based blend model

We used a blended ensemble model based on PyCaret, an open-source Python tool that streamlines machine learning procedures, to estimate demand. PyCaret is a comprehensive machine learning application that greatly increases productivity and expedites the experimentation process. Regression analysis was used to forecast continuous values that represent the demand for shared electric kickboards in the future.

For our experiment, several machine learning libraries must be imported, including pycaret (PyCaret:, 2020), seaborn, NumPy, and pandas. Recently for a regressing-based approach, extreme gradient Boosting has been used in many prediction tasks for better performance. In contrast to bagging techniques, which predict outcomes separately for each sample, Extreme Gradient Boosting (XGBoost) is unique in that it takes the weight of prior samples' results into account (Jung et al., 2020). It keeps learning from the results of earlier samples, which influences samples after that. XGBoost provides better model performance and a quicker learning rate when compared to other gradient boosting models (Ma & Yan, 2019; Torlay et al., 2017). Gradient boosting is prone to overfitting even though it usually concentrates only on training data results. However, by changing the hyperparameter settings, XGBoost reduces overfitting and makes the intended learning outcomes possible.

The Random Forest algorithm, which consists of several decision trees, functions based on the idea that the combined knowledge of numerous traditional algorithms is frequently greater than that of a single novel one (Zhan et al., 2018). Using this method, the ultimate forecast is obtained by combining data from multiple trees. In this approach, the final prediction is determined by aggregating results from numerous trees. Functioning as both a representative bagging method and a voting method, the Random Forest model leverages the collective predictions of multiple samples to determine the most prevalent outcome (Cao et al., 2021). Unlike a single decision tree, Random Forest employs learning across multiple decision trees, with the final prediction based on the most frequently occurring result. Similarly, the Extra Tree regressor employs a comparable learning approach to the Random Forest. Nonetheless, while Random Forest algorithms operate differently, Extra Trees opt for a strategy where multiple decision trees randomly select features to optimize outcomes (Meddage et al., 2021).

To construct our blended model, we utilized PyCaret's ensemble learning capabilities, which combine the strengths of multiple algorithms to enhance predictive performance. Our approach involved several steps: data preparation, model training, and model blending. We preprocessed the data, including data cleaning, handling missing values, and feature engineering to prepare it for modeling. Using PyCaret, we trained individual base models (XGBoost, Random Forest, and Extra Trees) on the prepared dataset. Consequently, the faster learning pace of Extra Trees, which utilizes this feature selection method, surpasses that of Random Forest. This leads us to favor Extra Trees as a meta-learner for robust prediction. This ensemble approach leverages the strengths of each model to improve overall accuracy. The impact of our blended ensemble model, incorporating synthetic data generated through our GAN approach, is discussed in detail in the results section. Our findings indicate that the proposed model significantly enhances demand prediction accuracy, as evidenced by reduced prediction errors.

3. Experimental results

This section delves deeply into both the trial outcomes and an exhaustive analysis of sales projections. It encompasses two distinct categories of studies derived from empirical results. We first looked at several existing GAN models to create synthetic time series data using our GAN model and provided performance comparisons with other GAN models. Second, the predictions made with and without synthetic data are compared using a blending regressors model.

3.1. Implementation details

The overall experimental environment of our system with the following specifications: Intel(R) Core(TM) i5-8500K CPU 3.70 GHz. Our system had a 64-bit operating system with 32 GB memory, 1 TB HDD. The experiments were conducted using an NVIDIA GeForce RTX 4080

Table 2
System components and specification used for the implementation.

Component name	Experimental parameters
Operating system	Windows 10, 64 bit
CPU	Intel(R) Core(TM) i5-8500K CPU @ 3.70 GHz
GPU	NVIDIA GeForce RTX 4080
RAM	32 GB
Development tool	Python 3.7.11
PyCaret	pycaret 2.3.6
Library	Tensorflow
IDE	Jupyter Notebook

GPU. For all our experiments we have utilized the system configuration mentioned in Table 2.

3.2. Experimental setup

We compared the performance of modified CWGAN-GP with existing time series synthetic data generation models like tabular-GAN-skip-WGAN-GP (Hazra & Byun, 2021), CTGAN, TGAN, TGAN-Skip, LSTM-AE, conditional tabular-GAN (Chatterjee & Byun, 2023). We conducted an evaluation comparing the generated synthetic data to assess its authenticity using similarity score metrics. Throughout the experiments, our proposed model consistently outperformed alternative models, leading to subsequent recommendations. The metrics presented demonstrated a strong correlation with sample quality, indicating their effectiveness as indicators of synthesizer performance. The discriminator architecture consists of a solitary dense layer featuring 50 nodes, accompanied by a learning rate fixed at 0.001. Furthermore, the random noise vector possesses a dimensionality of 200, with an L2-norm value of 0.00001. Training iterations spanned 800 epochs, with the number of steps determined by the ratio of N to batch size. Each batch comprised 200 rows of data.

Our model was trained and assessed using a 90–10 split of the original data. 10% of the data was set aside for testing, while the remaining 90% was used for training. In the next phase, synthetic data is combined with the original training data. It is important to note that the model was only evaluated using the original test dataset to ensure an unbiased assessment of the impact of synthetic data augmentation.

For the pycaret environment setting, we have to import the pycaret into our system. In the pycaret framework, the setup() function plays a pivotal role by initializing the environment and establishing the transformation pipeline necessary for data preparation, modeling, and deployment. As a fundamental step at the outset of any experiment, the setup function requires two mandatory parameters: a pandas frame and the designation of the target column, which in our case is denoted as *rent_number*. The pre-processing pipeline can be modified using the other optional options. Once the setup is finished, it is advisable to begin modeling by comparing all of the results to assess performance. This function trains every model in the model library and evaluates each one's performance using K-fold cross-validation. A scoring grid with the average values for MAE, MSE, RMSE, R2, RMSLE, and MAPE across all folds of all the models in the model library is printed in the output. We have used the k fold's default value of 10; however, we can change it according to our needs.

3.3. Synthetic time-series data

We assessed the quality of generated synthetic data using our suggested modified CWGAN-GP model along with several other GAN models, as shown in Table 4. The following criteria are applied to each model as a comparison: Time-series data created from synthetic data has been improved. We are evaluating the effectiveness of each model using a set of data. The results show that our suggested model can be used to create fake time series data because it has the highest correlation coefficient and much lower error than other models.

Table 3
Correlations values of synthetic vs. original data.

Correlation values	Relation
0	No correlation
0 to 0.2	Very weak correlation
0.21 to 0.4	Weak correlation
0.41 to 0.6	Moderate correlation
0.61 to 0.8	Strong correlation
0.81 to 1	Perfect correlation

Table 4
Comparison of GAN models performance based on Mean Correlation Coefficient, PRD, and RMSE.

Model	Mean correlation coefficient	PRD	RMSE
CTGAN	0.822	71.5	0.75
LSTM-AE	0.837	148.67	0.79
TGAN	0.788	76.4	0.74
TGAN-skip	0.842	71.5	0.68
TGAN-Skip-WGAN-GP	0.901	52.78	0.58
CWGAN-GP-PACGAN	0.932	53.2	0.50
Modified-CWGAN-GP	**0.961**	**52.9**	**0.43**

3.3.1. Pearson's correlation coefficient

The Pearson correlation coefficient, ranging from 0 to 1, is a statistical measure used by the Pearson R test (Benesty et al., 2009) to assess the strength and direction of the linear relationships between variables. A critical factor in synthetic data evaluation is the ratio of oversampled raw data to the generated data. Negative correlation coefficients indicate inverse relationships, while positive values suggest direct relationships. The absolute value of the correlation coefficient determines the strength of this connection, with larger values indicating stronger correlations. Eq. (9) defines the calculation of Pearson's correlation coefficient P_{CC} within our proposed methodology. Here, $Orig$ signifies the dataset of actual observations, while Syn represents the artificially generated dataset. Table 3 presents the correlation values.

$$P_{CC} = \frac{\sum_{i=1}^{n}(Orig_i - Orig)(Syn_i - Syn)}{\sqrt{[\sum_{i=1}^{n}(Orig_i - \overline{Orig})][\sum_{i=1}^{n}(Syn_i - \overline{Syn})]}} \tag{9}$$

3.3.2. Percent root mean square difference

To determine the degree of distortion between two signals—the original and the generated signals—the percent root means square difference, or PRD, has been employed. As Eq. (10), it has been defined.

$$PRD = \sqrt{100\frac{\sum_{i=1}^{n}(Orig_i - Syn_i)^2}{\sum_{i=1}^{n}(Orig_i)^2}} \tag{10}$$

3.3.3. Root Mean Square Error

In mathematics, Root Mean Squared Error (RMSE) is the square root of Mean Squared Error (MSE). MSE measures the difference between the predicted and actual values. Using the square root prevents errors from showing a negative sign since errors can be positive or negative. This relationship is represented by the following Eq. (11).

$$\mathrm{RMSE} = \sqrt{\mathrm{n}\sum_{i=1}^{\mathrm{n}}\left(\mathrm{YOrig_i} - \mathrm{YSyn_i}\right)^2} \tag{11}$$

According to Table 4, three metrics are compared to determine the performance of different GAN methods with Mean Correlation Coefficient, Precision and Recall Distance, and Root Mean Square Error. The models listed include CTGAN, LSTM-AE, TGAN, TGAN-skip, TGAN-Skip-WGAN-GP, CWGAN-GP-PACGAN, and our proposed Modified-CWGAN-GP. Higher values indicate better performance for a given pair of synthetic and real data based on the Mean Correlation Coefficient. In the PRD metric, models are ranked by their performance in terms of precision and recall. Low values indicate better performance. The RMSE measures the difference between the predicted

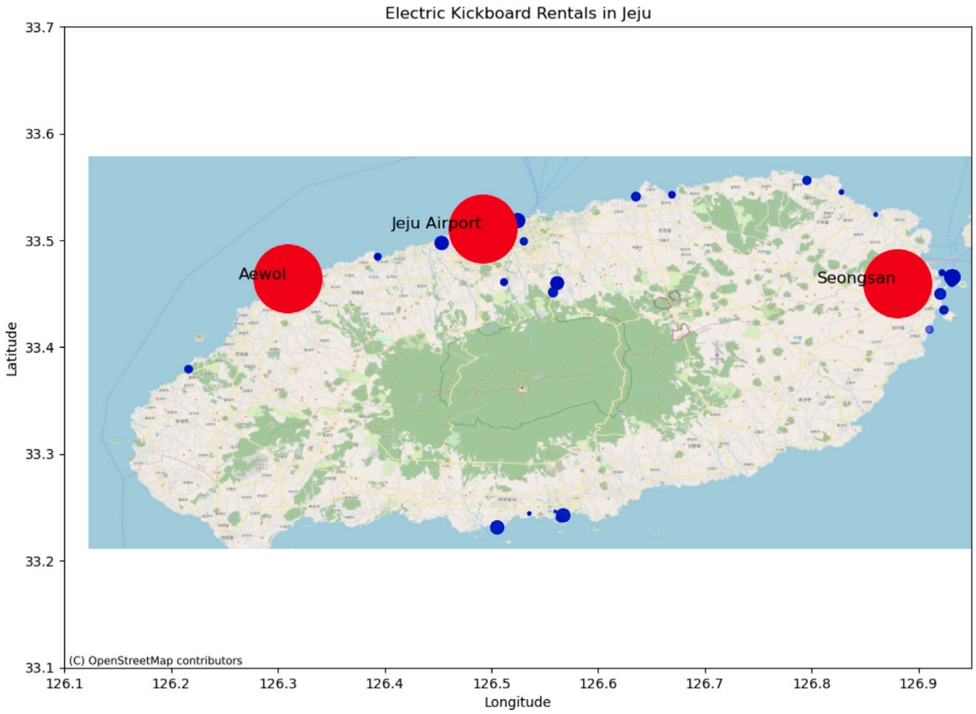


Fig. 6. Distribution of electric kickboard rental stations across Jeju Island with highlights of popular tourist destinations.

and actual values, and lower values correspond to higher accuracy. With the lowest PRD of 52.9, the lowest RMSE of 0.43, and the best Mean Correlation Coefficient of 0.961 among the assessed models, the Modified-CWGAN-GP performs better than the other models. This implies that the Modified-CWGAN-GP model produces synthetic data with enhanced precision–recall balance and high accuracy that closely reflects real data. Even if they perform well, some models fall short of the Modified-CWGAN-GP model's overall efficacy according to these measures.

3.4. Demand prediction

A visualization of the distribution of electric kickboard rental stations on Jeju Island is provided in Fig. 6, highlighting popular tourist sites and rental patterns based on geographic context.

Blue dots represent individual rental stations, with the size of each dot proportional to the number of rentals, illustrating rental activity density. Notable locations such as Jeju Airport, Aewol, and Seongsan are marked with red dots and labeled, reflecting their significance as high-activity areas due to tourist traffic. In addition to providing accurate geographic orientation, latitude and longitude are shown on the figure's axes. As a result of correlating rental patterns with tourist hotspots, this visualization can identify high-demand areas, plan additional rental stations, and enhance service infrastructure.

Fig. 7 illustrates the clustering of electric kickboard rental data across different sectors in Jeju Island, highlighting the characteristic differences between the two-sector and four-sector clustering approaches, as well as the impact of data augmentation. In the two-sector clustering, Jeju City is divided into three sub-sectors (Sectors 1, 2, and 3) due to its high rental demand, while Seogwipo City remains a single sector (Sector 4). Sector 1 in Jeju City stands out as a hotspot for rentals, particularly near Jeju Airport and popular tourist destinations like Aewol and Seongsan, whereas Sector 3 exhibits the lowest demand as shown in Fig. 6. The four-sector clustering provides a more granular view, revealing varied rental activities within Jeju City.

A clear discrepancy in rental numbers can be seen before augmentation, with the majority of rentals concentrated in Sector 1 and the majority in Sector 3. After the augmentation of the data, the rental distribution becomes more balanced, addressing data sparsity in low-demand areas and providing a comprehensive view of rental patterns. This augmentation enhances the dataset's robustness, providing a more comprehensive view of rental patterns and improving the accuracy of demand forecasting. The figure effectively demonstrates how sector-specific analysis and data augmentation can lead to better insights and strategic planning for rental services.

3.4.1. Mean Absolute Percentage Error

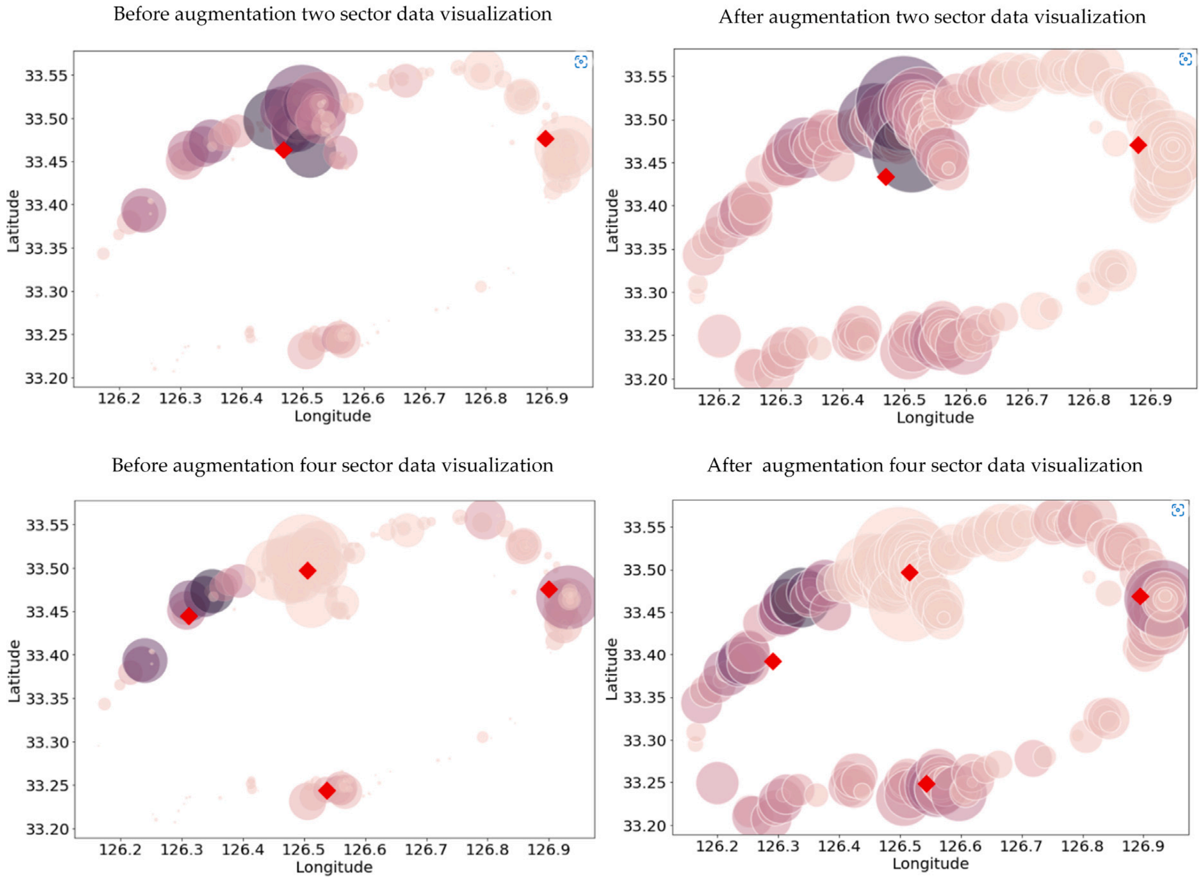
A statistical metric called Mean Absolute Percentage Error (MAPE) assesses a machine learning system's accuracy on a particular dataset. Like a loss function, MAPE measures the error that is seen when evaluating the model. We can assess the degree of accuracy in differences between estimated and actual values by utilizing MAPE. MAPE can also be calculated by expressing it as a percentage. A reasonably accurate prediction model is indicated by a low MAPE. The basic formula for MAPE computation is shown in Eq. (12).

$$MAPE = \frac{100\%}{n} \sum_{k=1}^{n} \left| \frac{\left(y_{\text{actual}} - \hat{y}_{\text{pred}}\right)}{y_{\text{actual}}} \right| \tag{12}$$

Here, y_{actual} = actual value, $\hat{y}_{\text{pred}}$ = predicted value, n = number of observations, and the vertical bars stand for absolute values.

3.4.2. R2 score

The R2 scores of regression-based machine learning models are used to evaluate them. It is sometimes referred to as the coefficient of determination and is expressed as R squared. It computes the degree of variation in the predictions explained by the dataset. It is the discrepancy between the model's predictions and the dataset's samples, as previously mentioned.


Before augmentation two sector data visualization

After augmentation two sector data visualization

Before augmentation four sector data visualization

After augmentation four sector data visualization

Fig. 7. Data visualization of two and four sectors without and with synthetic data.

$$R^2 = 1 - \frac{\sum \left(y_{actual} - y_{pred}\right)^2}{\sum \left(y_{actual} - \bar{y}_{pred}\right)^2} \tag{13}$$

To thoroughly evaluate the quality of synthetic time series data, we used several metrics offered by TimeGAN (Yoon et al., 2019), including discrimination scores, coverage metrics, and t-SNE plots. The discriminant score, which measures the classifier's ability to distinguish real data from synthetic data, was 0.54. This shows that the classifier can distinguish the two datasets with moderate accuracy, suggesting a reasonable similarity between the real and synthetic data.

The coverage metric, which measures how well the generated data covers the modes of the real data, was 0.963. This indicates that the real data mode is largely represented by the generated data. However, this must be interpreted in the context of other indicators. Additionally, the t-SNE visualization Fig. 8 shows some overlaps, but also clear clusters of real and synthetic data points. This indicates that synthetic data cannot fully capture the distribution of real data.

The demand prediction performance with and without synthetic data augmentation is compared in Figs. 9 and 10. The graphs on the left display predictions made solely with the original data, while the graphs on the right display forecasts made by combining the original data with synthetic data. There are two lines in both figures, one representing the actual number of rental units, and the other representing the predicted number of rental units. As evident from the graphs, by incorporating synthetic data alongside the original data, we achieved improved prediction accuracy. This is particularly significant when compared to the results obtained using the original data alone. The test data used spans from May 22nd, 2021, to June 11th, 2021. The x-axis represents the date, and the y-axis represents the shared electric kickboard rental count.

We have considered MAPE and R2-score for evaluation metrics. As we can see from Table 5 and 6 represents the proposed model's performance evaluation with two-sector data. We have presented both synthetic and real data model findings. The result shows that when the synthetic data was combined with real data, the model was trained using the combined data, and the MAPE score decreased when testing with the original test data. We demonstrate that when the synthetic data generated by our proposed model is combined with the original data, the prediction accuracy rises and the error rate falls. Table 7, 8, 9, and 10 represent the proposed model's performance evaluation with four-sector data. While considering four sector data, the third and fourth part of the four sector data is pretty small. At the beginning of our experiment using the original data, we faced problems getting better prediction results; the result improved a bit after we added original and synthetic data but still failed to get good results.

Compared to all other ensemble models tested and validated, the presented model outperforms them all. In addition to demonstrating the model's strength, the validation performance analysis shows that the model attains a low MAPE and a high R2 score compared to the baseline.

We compared the performance of our suggested GAN-based synthetic data generation approach to a number of common GAN models, such as TVAE (Tabular Variational AutoEncoder), TGAN (Tabular GAN), CTGAN, Wasserstein Generative Adversarial Network (WGAN), and our proposed GAN model, to verify its efficacy. The performance of these models is compiled in Table 11 using important metrics as R-squared, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Superior performance was shown by our suggested

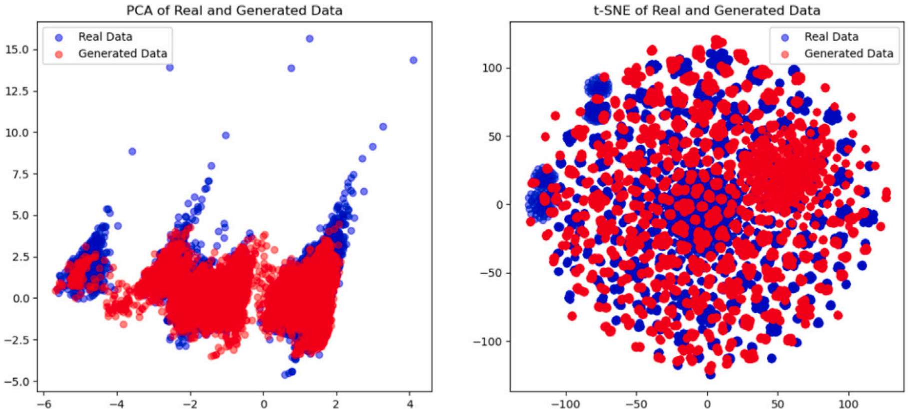
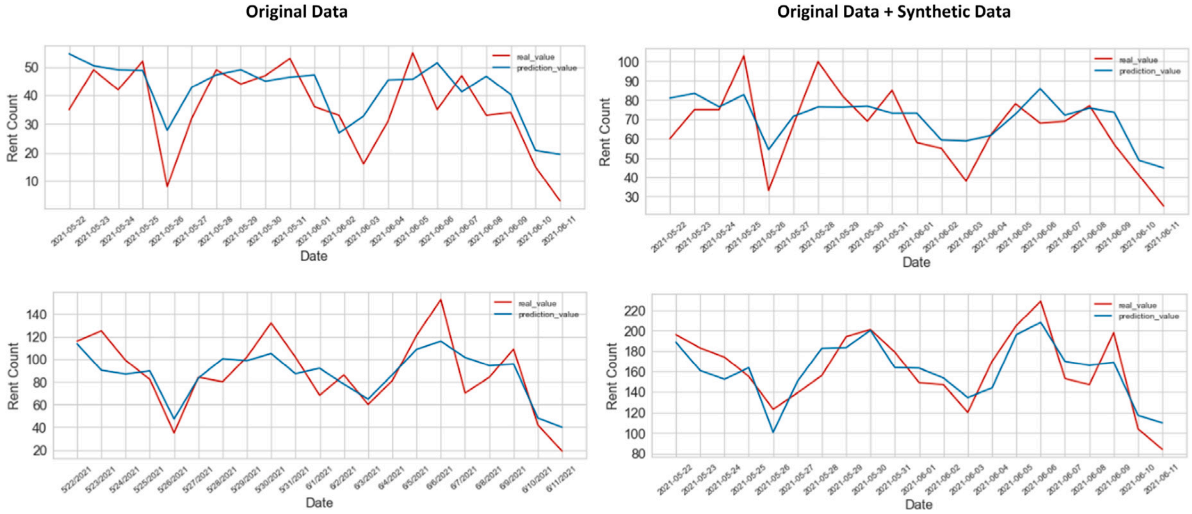


Fig. 8. t-SNE plot of real and generated data.


Original Data

Original Data + Synthetic Data

Fig. 9. Impact of data augmentation on two-sector prediction accuracy for electric kickboard demand.

Table 5
Ensemble model performance comparison with the first part of two-sector data.

Model Name	Original Data		Original + Synthetic Data	
	MAPE	R2	MAPE	R2
CatBoost+Extra Tree + Random Forest	61.11	0.5190	28.58	0.4916
AdaBoost+ Extra Tree + XGBoost	56.72	0.3822	37.39	0.4981
CatBoost + LGBM + Random Fores	64.92	0.3711	24.13	0.4624
Proposed Model	**62.68**	**0.4031**	**21.46**	**0.5099**

Table 6
Ensemble model performance comparison with the second part of two-sector data.

Model Name	Original Data		Original + Synthetic Data	
	MAPE	R2	MAPE	R2
CatBoost+Extra Tree + Random Forest	44.11	0.5190	22.58	0.5942
AdaBoost+ Extra Tree + XGBoost	36.02	0.4822	16.88	0.5481
CatBoost + LGBM + Random Fores	34.42	0.5981	21.43	0.6624
Proposed Model	**20.87**	**0.6820**	**10.88**	**0.7428**

Table 7
Ensemble model performance comparison with the first part of four-sector data.

Model Name	Original Data		Original + Synthetic Data	
	MAPE	R2	MAPE	R2
CatBoost+Extra Tree + Random Forest	34.22	0.3990	16.58	0.4911
AdaBoost+ Extra Tree + XGBoost	32.72	0.3722	15.78	0.4581
CatBoost + LGBM + Random Fores	31.42	0.3981	15.23	0.5124
Proposed Model	**31.11**	**0.4190**	**11.58**	**0.5416**

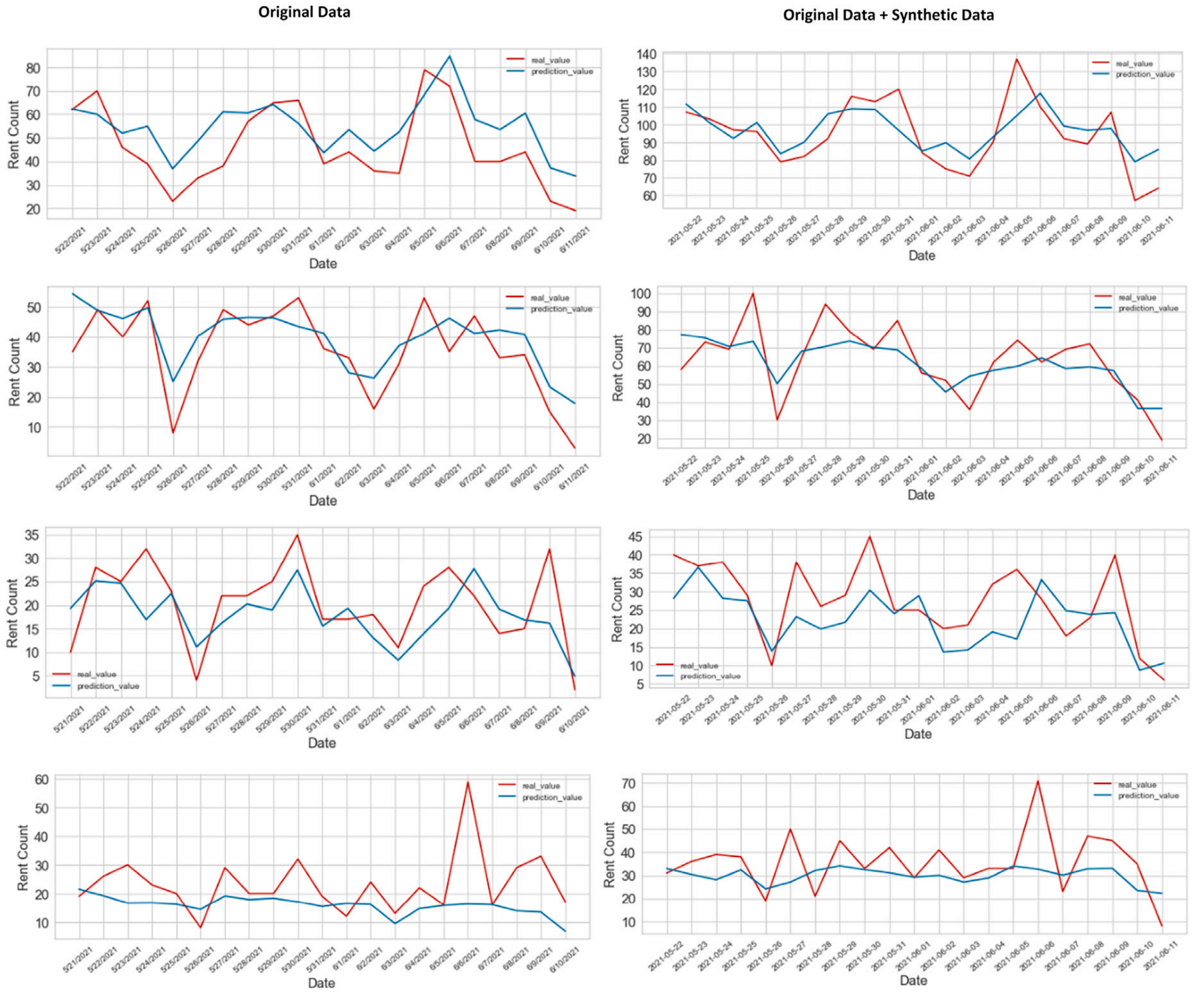


Fig. 10. Impact of data augmentation on four-sector prediction accuracy for electric kickboard demand.

Table 8
Ensemble model performance comparison with the second part of four-sector data.

Model Name	Original Data		Original + Synthetic Data	
	MAPE	R2	MAPE	R2
CatBoost+Extra Tree + Random Forest	34.22	0.3990	26.25	0.4911
AdaBoost+ Extra Tree + XGBoost	32.72	0.3722	31.78	0.4793
CatBoost + LGBM + Random Fores	58.37	0.5141	25.13	0.5304
Proposed Model	**53.66**	**0.5769**	**20.59**	**0.5843**

Table 9
Ensemble model performance comparison with the third part of four-sector data.

Model Name	Original Data		Original + Synthetic Data	
	MAPE	R2	MAPE	R2
CatBoost+Extra Tree + Random Forest	41.22	0.2399	29.77	0.2236
AdaBoost+ Extra Tree + XGBoost	42.91	0.3022	31.78	0.4793
CatBoost + LGBM + Random Fores	38.37	0.3381	28.13	0.3304
Proposed Model	**39.38**	**0.3434**	**28.64**	**0.3436**

Table 10
Ensemble model performance comparison with the fourth part of four-sector data.

Model Name	Original Data		Original + Synthetic Data	
	MAPE	R2	MAPE	R2
CatBoost+Extra Tree + Random Forest	34.12	0.0244	30.77	0.0158
AdaBoost+ Extra Tree + XGBoost	43.14	0.0402	31.18	0.0431
CatBoost + LGBM + Random Fores	38.37	0.0481	33.13	0.0544
Proposed Model	**33.60**	**0.0949**	**30.67**	**0.1126**

Table 11
Performance comparison of GAN models for demand prediction.

Model	MAE	RMSE	R2
TVAE	1.132	0.79	0.71
TGAN	1.104	0.74	0.78
CTGAN	1.088	0.59	0.84
WGAN	1.082	0.55	0.85
Proposed CWGAN-GP	**0.919**	**0.50**	**0.88**

CWGAN-GP on all criteria, demonstrating its resilience in producing realistic synthetic data and raising the accuracy of demand forecasts.

4. Discussion

One interesting approach to improving operational efficiency and customer satisfaction in the business world is the use of machine learning techniques for demand forecasting, especially in the context of shared electric kickboards. Because consumer behavior and market trends are dynamic, traditional methodologies frequently fail to effectively forecast future demand. In this study, we addressed this challenge by leveraging GAN to generate synthetic data for demand prediction. CWGAN-GP is the modified procedure used in this study for generating synthetic data that successfully addresses a limitation and imbalance in the shared electric kickboard rental data. Clustering-based ensemble regression model accuracy was significantly improved by the generated synthetic data, which demonstrates oversampling methods. Synthetic data augmentation is effective in solving data scarcity and imbalance problems commonly associated with demand prediction. However, further exploration is needed to assess the generalizability of our approach to other datasets. Additionally, a more in-depth comparison with various ensemble models and a deeper understanding of the limitations of our modified CWGAN-GP are valuable areas for future research. In summary, this study shows the potential use of synthetic data generation using GANs for enhancing the accuracy of demand prediction, which will help the shared electric kickboard rental industry better allocate resources and serve its customers.

5. Conclusion

We describe a synthetic data-generating generative adversarial network (GAN) strategy to improve demand prediction for a shared electric kickboard-based service provider. Unbalanced data is a common problem in theoretical research as well as real-world applications. Various resampling strategies, primarily oversampling methods, have been proposed to address the challenges posed by small and imbalanced data classification problems. Oversampling typically involves generating minority class data to balance an unbalanced dataset. While these methods improve performance, the generated data often lacks realism because traditional oversampling focuses on the local information of the minority class. Additionally, mode collapse and unstable training are persistent issues with GAN-based oversampling methods. To address these challenges, we propose a modified CWGAN-GP, an innovative oversampling method built on WGAN-GP and driven by the conditional form of GAN. Our approach generates synthetic data that follows the general distribution of the original data. This work explores the application of the GAN model to generate synthetic time series data for enhancing shared electric kickboard data. Machine learning approaches enable us to accurately predict the demand for shared electric kickboards, particularly in areas where companies struggle to meet customer demand at the right location. To improve prediction accuracy and reduce prediction error, our proposed GAN model generates synthetic data based on the original data distribution, which is then concatenated with the original data. We also illustrate how integrating our generated oversampling data with the original dataset reduces prediction errors. Our results section highlights how our modified CWGAN-GP produces superior-quality synthetic data, thereby enhancing demand forecasting capabilities. In the future, we plan to improve and evaluate our model using various publicly available datasets. It also includes evaluations using discrimination scores, coverage metrics, and t-SNE plots to further evaluate the quality of synthetic time series data. Our preliminary findings demonstrate that synthetic data encompasses the majority of real data types, but that finer features and variability can still be better captured. Furthermore, future research will include the comparison of our work with diffusion-based models to determine the effectiveness of the proposed model against autoregressive denoising diffusion models.

CRediT authorship contribution statement

Subhajit Chatterjee: Data curation, Conceptualization, Formal analysis, Methodology, Writing – review & editing. **Debapriya Hazra:** Conceptualization, Formal analysis, Methodology. **Yung-Cheol Byun:** Investigation, Funding acquisition, Resources, Project administration, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. RS-2024-00405278).

Data availability

The data that has been used is confidential.

References

Amirkhani, A., Haghanifar, A., & Mosavi, M. R. (2019). Electric vehicles driving range and energy consumption investigation: A comparative study of machine learning techniques. In *2019 5th Iranian conference on signal processing and intelligent systems* (pp. 1–6). IEEE.

Ba, H. (2019). Improving detection of credit card fraudulent transactions using generative adversarial networks. arXiv preprint arXiv:1907.03355.

Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing* (pp. 1–4). Springer.

Cao, D., Zeng, K., Wang, J., Sharma, P. K., Ma, X., Liu, Y., & Zhou, S. (2021). Bert-based deep spatial-temporal network for taxi demand prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Chatterjee, S., & Byun, Y.-C. (2023). A synthetic data generation technique for enhancement of prediction accuracy of electric vehicles demand. *Sensors*, *23*(2), 594.

Douzas, G., & Bacao, F. (2018). Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with applications*, *91*, 464–471.

Esteban, C., Hyland, S. L., & Rätsch, G. (2017). Real-valued (medical) time series generation with recurrent conditional gans. arXiv preprint arXiv:1706.02633.

Fu, R., Chen, J., Zeng, S., Zhuang, Y., & Sudjianto, A. (2019). Time series simulation by conditional generative adversarial net. arXiv preprint arXiv:1904.11419.

García-Jara, G., Protopapas, P., & Estévez, P. A. (2022). Improving astronomical time-series classification via data augmentation with generative adversarial networks. arXiv preprint arXiv:2205.06758.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, *27*.

Grolinger, K., L'Heureux, A., Capretz, M. A., & Seewald, L. (2016). Energy forecasting for event venues: Big data and prediction accuracy. *Energy and Buildings*, *112*, 222–233.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, *30*.

Guo, Y., Xiong, G., Li, Z., Shi, J., Cui, M., & Gou, G. (2021). Combating imbalance in network traffic classification using GAN based oversampling. In *2021 IFIP networking conference* (pp. 1–9). IEEE.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 100–108.

Hazra, D., & Byun, Y.-C. (2021). Generating synthetic fermentation data of shindari, a traditional jeju beverage, using multiple imputation ensemble and generative adversarial networks. *Applied Sciences*, *11*(6), 2787.

Hong, J., Park, S., & Chang, N. (2016). Accurate remaining range estimation for electric vehicles. In *2016 21st Asia and south Pacific design automation conference* (pp. 781–786). IEEE.

Hulot, P., Aloise, D., & Jena, S. D. (2018). Towards station-level demand prediction for effective rebalancing in bike-sharing systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 378–386).

Iwana, B. K., & Uchida, S. (2021). Time series data augmentation for neural networks by time warping with a discriminative teacher. In *2020 25th international conference on pattern recognition* (pp. 3558–3565). IEEE.

Jung, K., Park, J., Son, S., & Ahn, S. (2020). Position prediction of wireless charging electric vehicle for auto parking using extreme gradient boost algorithm. In *2020 IEEE wireless power transfer conference* (pp. 439–442). IEEE.

Koivu, A., Sairanen, M., Airola, A., & Pahikkala, T. (2020). Synthetic minority oversampling of vital statistics data with generative adversarial networks. *Journal of the American Medical Informatics Association*, *27*(11), 1667–1674.

Li, Y., Pan, Q., Wang, S., Yang, T., & Cambria, E. (2018). A generative model for category text generation. *Information Sciences*, *450*, 301–315.

Ma, F., & Yan, X. (2019). Research on the energy consumption estimation method of pure electric vehicle based on xgboost. In *2019 3rd international conference on electronic information technology and computer engineering* (pp. 1021–1026). IEEE.

Martínez-Álvarez, F., Troncoso, A., Asencio-Cortés, G., & Riquelme, J. C. (2015). A survey on data mining techniques applied to electricity-related time series forecasting. *Energies*, *8*(11), 13162–13193.

Meddage, D., Ekanayake, I. U., Weerasuriya, A., & Lewangamage, C. (2021). Tree-based regression models for predicting external wind pressure of a building with an unconventional configuration. In *2021 moratuwa engineering research conference* (pp. 257–262). IEEE.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957.

Mogren, O. (2016). C-RNN-GAN: Continuous recurrent neural networks with adversarial training. arXiv preprint arXiv:1611.09904.

Pérez-Chacón, R., Talavera-Llames, R. L., Martinez-Alvarez, F., & Troncoso, A. (2016). Finding electric energy consumption patterns in big time series data. In *Distributed computing and artificial intelligence, 13th international conference* (pp. 231–238). Springer.

PyCaret: (2020). An open source, low-code machine learning library in Python. URL https://pycaret.org/.

Tang, B., Tu, Y., Zhang, Z., & Lin, Y. (2018). Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks. *IEEE Access*, *6*, 15713–15722.

Torlay, L., Perrone-Bertolotti, M., Thomas, E., & Baciu, M. (2017). Machine learning–Xgboost analysis of language networks to classify patients with epilepsy. *Brain Informatics*, *4*(3), 159–169.

Torres, J. F., Fernández, A. M., Troncoso, A., & Martínez-Álvarez, F. (2017). Deep learning-based approach for time series forecasting with application to electricity load. In *International work-conference on the interplay between natural and artificial computation* (pp. 203–212). Springer.

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., & Xu, H. (2020). Time series data augmentation for deep learning: A survey. arXiv preprint arXiv:2002.12478.

Yoon, J., Jarrett, D., & Van der Schaar, M. (2019). Time-series generative adversarial networks. *Advances in Neural Information Processing Systems*, *32*.

Zhan, Y., Luo, Y., Deng, X., Grieneisen, M. L., Zhang, M., & Di, B. (2018). Spatiotemporal prediction of daily ambient ozone levels across China using random forest for human exposure assessment. *Environmental Pollution*, *233*, 464–473.