applied sciences

MDPI

Article

Taxi Demand and Fare Prediction with Hybrid Models: Enhancing Efficiency and User Experience in City Transportation

Ka Seng Chou [1,2,*], Kei Long Wong [1,2], Boliang Zhang [1], Davide Aguiari [3], Sio Kei Im [4], Chan Tong Lam [1], Rita Tse [1], Su-Kit Tang [1] and Giovanni Pau [1,2,3,5]

[1] Faculty of Applied Sciences, Macao Polytechnic University, Macao SAR 999078, China; keilong.wong@mpu.edu.mo (K.L.W.); p1807471@mpu.edu.mo (B.Z.); ctlam@mpu.edu.mo (C.T.L.); ritatse@mpu.edu.mo (R.T.); sktang@mpu.edu.mo (S.-K.T.); giovanni.pau@unibo.it (G.P.)
[2] Department of Computer Science and Engineering, University of Bologna, 40126 Bologna, Italy
[3] Autonomous Robotics Research Center, Technology Innovation Institute (TII), Abu Dhabi P.O. Box 9639, United Arab Emirates; davide.aguiari@tii.ae
[4] Macao Polytechnic University, Macao SAR 999078, China; marcusim@mpu.edu.mo
[5] Samueli Computer Science Department, University of California, Los Angeles, CA 90095, USA
* Correspondence: kaseng.chou@mpu.edu.mo

Abstract: An essential part of a city's transportation infrastructure, taxis allow for regular encounters between drivers and customers. Nevertheless, there are issues with efficiency since there is an imbalance in the supply and demand for taxis. This study describes the creation of a platform that serves both customers and taxi drivers by offering immediate forecasts of demand and fare. Root mean squared error (RMSE) of 3.31 and a negative log-likelihood of −3.84 , the long short-term memory recurrent neural network (LSTM-RNN) with the mixture density network (MDN) is employed to forecast taxi demand. The best RMSE of 3.24 is obtained for fare prediction via an ensemble learning model that integrates linear regression (LR), ridge regression (RR), and multilayer perceptron (MLP). To ensure peak performance, the models are systematically created, implemented, trained, and improved. By integrating these models into a web application interface, the taxi service system offers a better overall user experience, which improves urban mobility.

Keywords: taxi demand; taxi fare; LSTM-MDN; ensemble learning; intelligent transportation

Citation: Chou, K.S.; Wong, K.L.; Zhang, B.; Aguiari, D.; Im, S.K.; Lam, C.T.; Tse, R.; Tang, S.K.; Pau, G. Taxi Demand and Fare Prediction with Hybrid Models: Enhancing Efficiency and User Experience in City Transportation. *Appl. Sci.* **2023**, *13*, 10192. https://doi.org/10.3390/app131810192

Academic Editor: Pauline Ong

Received: 23 August 2023
Revised: 6 September 2023
Accepted: 8 September 2023
Published: 11 September 2023

1. Introduction

Taxi service systems within large cities exhibit complexity due to the intricate interplay and self-organization between taxi drivers and passengers. Two significant inefficiencies in this complex system stand out: excessive numbers of vacant trips and prolonged passenger wait times [1]. These challenges primarily arise from an imbalance between supply and demand, which could be mitigated through the establishment of a reliable mechanism for forecasting taxi demand. Central to this challenge is the fare structure. It directly influences the economic viability of taxi trips with varying distances and destinations [2]. In densely populated areas, there is often an oversupply of taxis, while remote areas suffer from insufficient availability [3]. Employing real-time point-to-point taxi pricing data alongside route information enhances the accuracy of demand prediction models, offering valuable insights into taxi demand [4]. Likewise, by providing fare projections, it is possible to create a closer approximation of pricing between taxis and third-party service providers, thereby reducing wait times and increasing overall efficiency.

Giving taxi drivers access to a predicted taxi demand system is the main way to handle this issue [5]. By forecasting future taxi demand and proactively positioning their vehicles in various places in advance, they can balance operational efficiency and profitability [6]. Urban traffic forecasting poses unique geographical and temporal challenges due to the

intricate interplay between these variables. The characteristics of nodes (locations) and edges (relationship between locations) subtly influence these correlations [7]. Traditional approaches fall short when relying solely on basic Global Positioning System (GPS) coordinates to describe location information [8]. Leveraging a fusion of deep learning models with varying levels of complexity and adaptiveness shows promise based on available data characteristics [9–12]. However, certain factors, such as time of day, day of the week, weather conditions, and special events, introduce regularities and patterns in taxi demand, making the widely used classic models, such as autoregressive integrated moving average (ARIMA) model, less accurate [13,14]. A more convincing model comes in the form of an MDN and recurrent neural network (RNN) fusion instead of offering a broad city-wide average. In this setting, the RNN model predicts temporal demand fluctuations, whereas the MDN model prognosticates pickup and dropoff demand distribution in each area [15].

Significantly improving the accuracy of traditional taxi fare models, such as the conventional LR model, holds substantial potential [16]. Taxi fares depend on various factors beyond distance, including drop charges and duration charges. While duration costs can be predicted, drop charges remain constant. Yang et al. (2005) found that variable taxi fares linearly increase with trip length due to traffic [17]. However, the accuracy of the non-linear results is still unknown in spite of this [18]. In order to learn non-linear patterns, researchers have investigated ensemble models for improved precision, including stacking classifiers and utilizing location-based social networks [19,20]. Among these, the ensemble model with a voting mechanism emerges as a particularly compelling approach [15].

This project aims to build a complete system that uses deep learning methods and historical taxi trip data to forecast demand trends and taxi fare estimates for the benefit of both passengers and taxi drivers. Accompanying this system is a user-friendly web application bridging the gap between theoretical research and practical execution. Visualizations, such as heatmaps and route maps, are anticipated to aid end users in comprehending the prediction results [21]. The platform also enables in-depth data analysis for research purposes, potentially facilitating future comparisons of fare structures among different taxi service providers.

This work centers on two primary objectives: prediction and visualization. The key contributions can be summarized as follows:

1. Development of a hybrid model intertwining LSTM-RNN with MDN to predict high-demand zones for customers across various time intervals, thereby optimizing taxi drivers' efficiency and income potential.
2. Creation of an ensemble model amalgamating LR, RR, and MLP to estimate taxi fares for point-to-point trips, while also identifying the nearest taxi pickup locations.
3. Establishment of a user-friendly interface facilitating seamless interaction between end users and the platform, enabling clear and effective visualization of the predicted outcomes.

The subsequent sections of this paper are structured as follows: in Section 2, the methodology employed in this study is expounded upon, encompassing aspects such as dataset utilization, preprocessing techniques, fare and demand modeling, as well as the development of the web application. The outcomes of the proposed approach are showcased in Section 3, unveiling the empirical results. Lastly, Section 4 serves as a comprehensive wrap-up, encapsulating the core findings of this paper and offering concluding insights.

2. Methodology

2.1. Dataset

The primary data source used in this study is taxi data from New York City (NYC), and it includes precisely vetted trip parameters such as pickup and dropoff locations, location IDs, fares, and speeds. This dataset, which focuses on the period from June to August 2018, takes up an enormous 5.31 GB of storage [22]. Comprising 14 distinct data elements and housing over 1.5 million rows of records, these three months encapsulate a wealth of information. It is worth noting that the location ID attribute assigns positive

integers from 1 to 258 to represent zones defined by encrypted GPS coordinates. To facilitate the processing of location IDs, we leverage the TLC-sourced taxi zone shape file as a crucial reference.

In Figure 1a, we showcase the relationship between trip distance and taxi fare for a specific month (June 2018) within the dataset. This analysis reveals a consistent linear trend, with occasional anomalies like the fixed fare of approximately USD 52 for trips to JFK airport. This consistency justifies the development of a foundational model based on LR. The use of a heatmap to visualize feature correlations, as seen in Figure 1b, underscores the strong connections between fare, time, and distance while also highlighting intricate relationships between other dataset elements. However, integrating these models into the front-end and back-end of the web application necessitates consideration of functional characteristics accessible from the front-end, which may lead to a reduction in the initial set of 14 features.

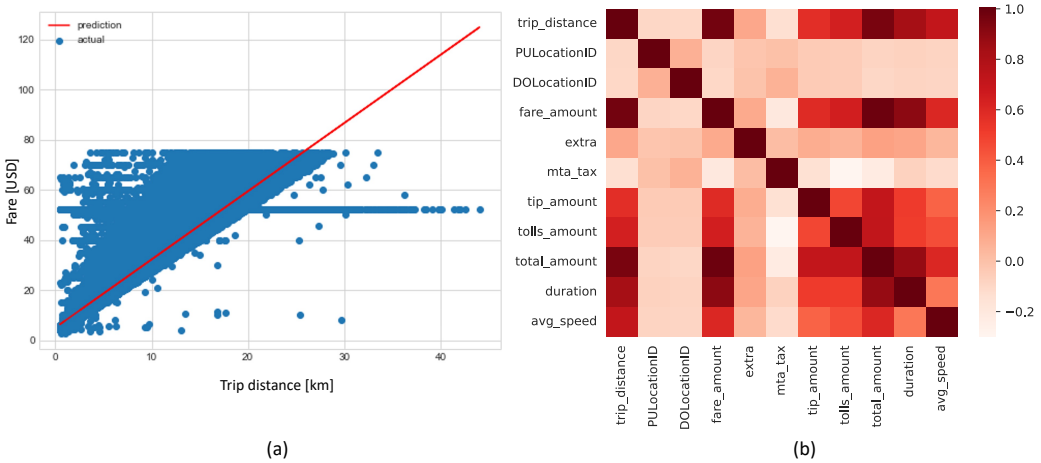


Figure 1. Dataset at a glance: (**a**) Illustrates a linear trend analysis encompassing trip distance and fare within a range of 30 miles/USD 80 and a statistical ordinary least squares (OLS) LR model. (**b**) Offers a heatmap visualizing the relationships among dataset features, showcasing their correlational dynamics.

2.2. Demand Prediction Model

Numerous studies have investigated the complex patterns and influencing elements of NYC's fluctuating taxi demand. Contrary to a random process, the prevailing consensus in the academic literature suggests that taxi demand exhibits a complex yet often predictable interplay driven by temporal factors. Spatial resolution analyses [23] reinforce this viewpoint. To address this complexity, researchers have created a model arsenal that includes conventional techniques: historical average, ARIMA, LR, RR, MLP, and XGBoost. Convolutional neural networks (CNNs) together with a long short-term memory (LSTM) model have even been used in recent advancements to solve image-based problems and improve prediction accuracy [24–26]. Furthermore, the adoption of large deep learning models like transformers or BERT has ushered in a new era of data-driven precision in understanding and predicting urban mobility patterns [27,28]. Despite the challenges, significant strides have been made in developing accurate models for forecasting taxi demand.

Contemporary models perform noticeably worse than middle-level deep models (such as LSTM/CNN or ensemble learning). In comparison, the large deep models, including transformers, BERT, and text-based models, are computationally expensive for applications predicting commercial taxi demand. This work applies hybrid and ensemble learning techniques to balance the computational cost in a real-time application. To circumvent the time-independent nature of conventional neural networks, we leverage the LSTM-RNN model, known for its ability to maintain contextual states across sequential inputs [29]. Unlike fixed-frequency time-series data, such as stock prices, the domain of travel demand presents a continuously fluctuating probability distribution over time. To effectively address

this inherent variability, we employ a downsampling preprocessing strategy, transforming the data into fixed-interval pickups, specifically on an hourly basis. This approach facilitates the many-to-one time-series prediction. In particular, we utilize a default input length of 10 records for the LSTM-RNN forecasting model, corresponding to a temporal range of 10 h. Figure 2 illustrates the architecture of the LSTM network employed for forecasting the number of pickups in each zone (i.e., location ID).

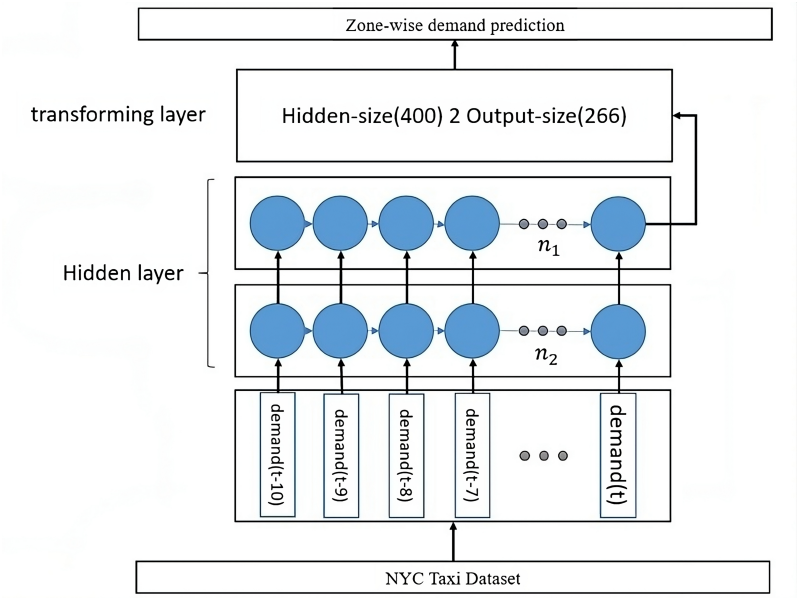


Figure 2. The LSTM-RNN layout with the tanh activation function on each RNN layer and the softmax output layer.

To address the limitations of a standalone LSTM-RNN model, we propose an innovative approach that integrates a multilayer perceptron with mixture density network (MLP-MDN) model. Traditional time-series neural networks often yield results that align closely with dataset means. However, forecasting taxi demand requires a deeper understanding of location distributions, which presents challenges due to excluded areas and a lack of distance correlation. Relying solely on LSTM-RNN output undermines meaningful visualization. The MLP-MDN model is designed to overcome these data limitations. It leverages the neural network's output layer (MDN), to shape distributions using carefully selected Gaussian weights. The output of the MDN layer combines different Gaussian distributions, each characterized by a unique mean and standard deviation. This MLP-MDN model accurately addresses the complexities of predicting taxi demand by predicting a wide range of outputs from given inputs.

The effectiveness of the MDN lies in its ability to construct a probability distribution function (PDF) for each input. This PDF, emerges as a weighted summation of discrete Gaussian probability distributions by using Equation (1). The distribution parameters are derived from the neural network's LSTM output. The sum of $\Pi(x)$ is confined to 1, guaranteeing a PDF sum of 1. Equation (2) defines the softmax formula for Π_k. The objective is to minimize the loss, as stated in Equation (3). By accurately representing distributional nuances, our combined model outperforms conventional LSTM-RNN models, revolutionizing the field of taxi demand prediction.

$$P(Y = y|X = x) = \sum_{k=0}^{K-1} \Pi_k(x)\Phi(y, \mu_k(x), \sigma_k(x)), \sum_{k=0}^{K-1} \Pi_k(x) = 1 \quad (1)$$

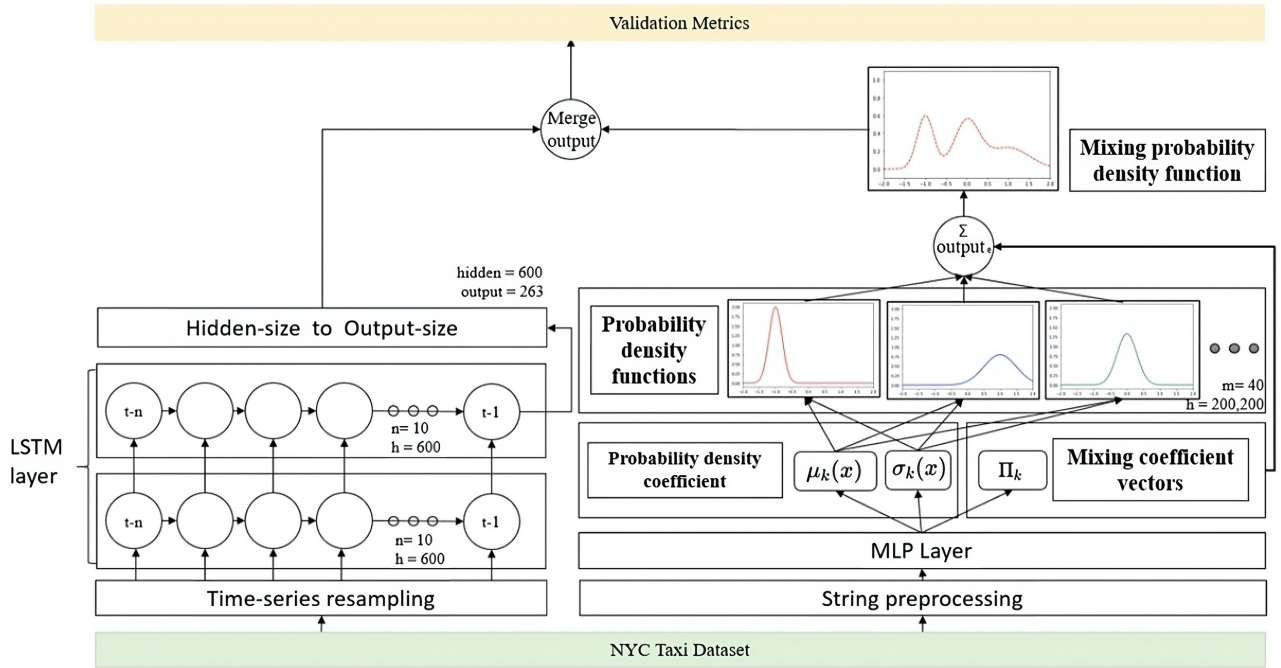
$$SoftMax(\Pi_k) = \frac{e^{\Pi_k}}{\sum_{k=1}^{n} e^{\Pi_k}} \quad (2)$$

$$E_t = -\ln\left\{\sum_{k=1}^{M} \Pi_k(x) g_k(y_t|x)\right\} \tag{3}$$

As shown in Figure 3a, the final model is a combination of the two networks described earlier. The model initially extracts essential information from the historical records to forecast temporal data at time t. Before entering the LSTM-RNN, this input undergoes a resampling processing layer. Subsequently, the model produces projected demand values for each location ID in the upcoming period. The MLP-MDN model simultaneously creates a probability distribution representing demand across the map at the specified time. This two-pronged approach not only aids in forecasting demand beyond predetermined zones but also equips the front-end with the capability for two-dimensional visualization.

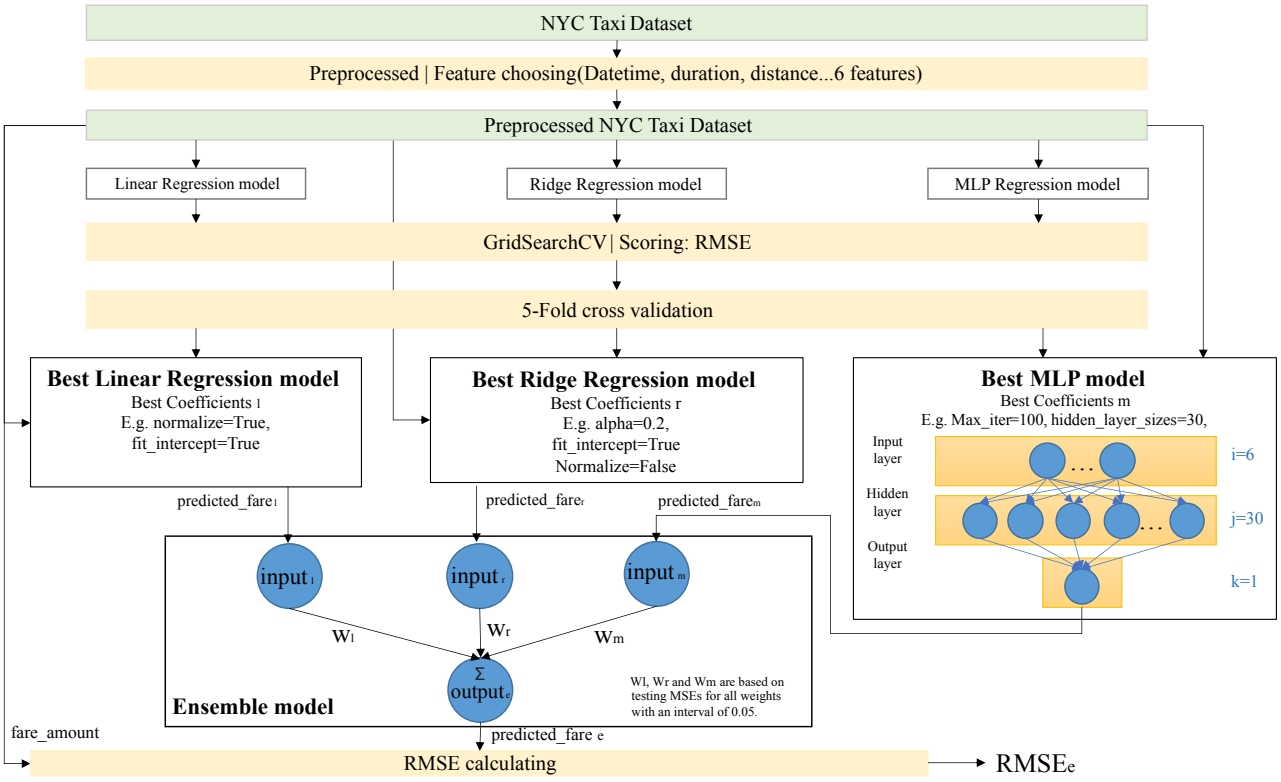
2.3. Fare Prediction Model

The key is building a model that can capture linear and non-linear patterns in taxi fare prediction. To achieve this, we employ ensemble learning to enhance forecast precision while minimizing the risk of making suboptimal decisions. This approach seamlessly transitions from complex deep models to more simplified alternatives, striking a balance between accuracy and efficiency, often outperforming individual algorithms. In our ensemble model, we combine three machine learning (ML) models, LR, RR, and MLP, with each assigned varying weights to account for linear and non-linear fare prediction characteristics. The resulting ensemble model is designed to provide a robust and versatile approach to fare prediction. Figure 3b depicts the model's condensed architecture. The symbols *Wl*, *Wr*, and *Wm* in the final model ensemble stand for weights. For each model in the combination, testing weights in the range of 0.05 to 0.95 in 0.05 increment steps is required. The mean squared error (MSE) is calculated, and the best combination with the lowest MSE for the forecast is chosen.



(a)

Figure 3. *Cont.*



(b)

Figure 3. The ML model architecture. (**a**) Optimal hybrid model architecture for demand prediction: combining LSTM-RNN and MLP-MDN models. (**b**) Optimal ensemble model architecture for fare prediction: combining LR, RR, and MLP models.

2.4. Web Application

To facilitate efficient city transportation and ensure positive user experiences, it is essential to bridge the gap between users and ML. Full-stack applications, however, require highly complex software development and deployment processes. We provide a comprehensive, standalone or cloud-based solution that simplifies the development, testing, and deployment processes. Figure 4a shows the application's front-end and back-end components. This approach integrates the data flow from data to ML models to the front-end by utilizing Django, which is a suitable back-end framework with Python as the fundamental language for ML [30]. Django's extensive community support surpasses that of alternative Python-based frameworks like Flask [31]. The front-end uses a website-like design, as seen in Figure 4a. The user interface makes important tasks including feature selection, timesteps, location selection, and forecasts easier. Visualizing historical data and ML model results is achieved using Google Maps [32], Apex Chart [33], and Chartjs [34], three essential tools.

Python, along with the Django web framework, constitutes critical back-end components. Figure 4b shows that the application server architecture follows the current model–view–template design. In order to streamline the creation of complex systems and guarantee continuing application design, implementation, and maintainability, this architecture incorporates module separation, demarcating data access, logical operations, and interface template composition. It coordinates communication between ML models, which may be stored in the application server's file system or on a separate server for ML model predictions in the back-end server. Clients receive websites with traceable HTTP-session objects from the Django view. Figure 4d shows online interfaces with specific sites for demand, fare, and data uploads (model retrain automatically) that are tailored

to NYC taxi drivers, passengers, and companies. The use of sophisticated self-adaptive Bootstrap components makes this design usable on both desktop and mobile devices [35].

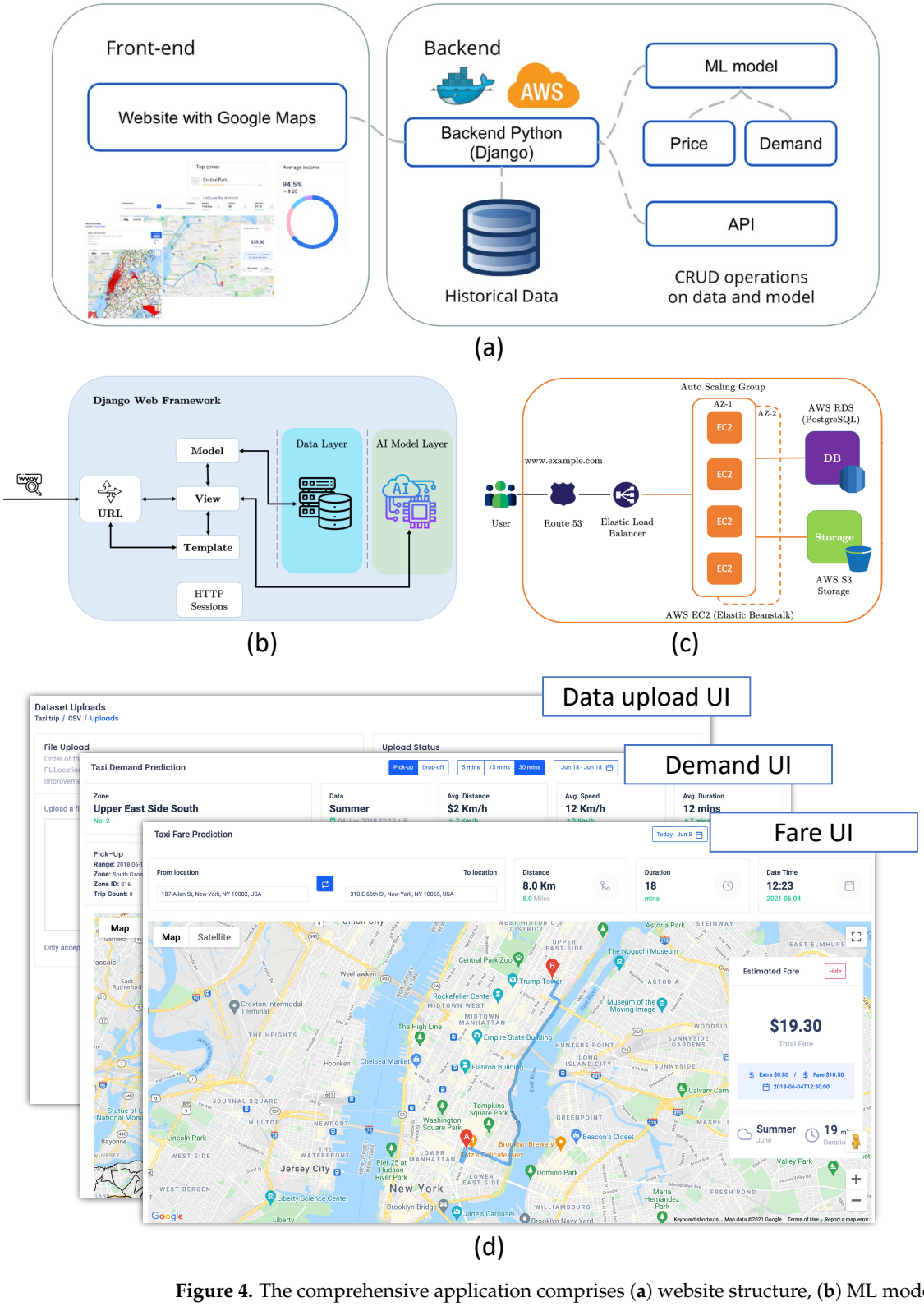


Figure 4. The comprehensive application comprises (**a**) website structure, (**b**) ML model-embedded framework, (**c**) one-click deploy cloud architecture, and (**d**) interfaces utilizing standard web technologies like HTML, JavaScript, CSS, and Bootstrap, optimized for desktop and portable devices.

The design allows for cloud-based deployment, providing increased functionality (data upload, storage, visualization, etc.) at a low price while maintaining data security. An architecture that allows for disaster recovery and backup strategies to guarantee data integrity during long-term deployment is shown in Figure 4c. The overall architecture is built on top of Amazon Web Services (AWS) [36] while also supporting Microsoft Azure and Google App Engine as additional cloud service providers. AWS Route 53, elastic load balancing, EC2 instances, RDS database, and S3 storage are the five parts that make up the back-end. One-click deployment supports all five of these components [37].

3. Result And Discussion

3.1. Predictive Models

In this section, we conduct quantitative analyses to evaluate the performance of all the trained models and assess the effectiveness of the proposed predictive models.

3.1.1. Evaluation Metric

To comprehensively assess the efficacy of our predictive methodology, we incorporate four widely recognized error metrics. For fare prediction, the employed loss functions encompass the MSE (4) and its derivative, the RMSE (5), representing the square root of the MSE. In these equations, $y_{predict}$ signifies the predicted fare amount, while y_{actual} represents the actual fare amount.

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(y^i_{actual} - y^i_{predict})^2 \tag{4}$$

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y^i_{actual} - y^i_{predict})^2} = \sqrt{\text{MSE}_{target}} \tag{5}$$

$$\text{MAE} = \frac{\sum_{i=1}^{m}|y^i_{test} - y^i_{predict}|}{m} \tag{6}$$

$$\max_{\theta} -\ln\left\{\sum_{k=1}^{M} w_k(x)g_k(y_t|x)\right\} \tag{7}$$

In the realm of demand prediction, the model's performance evaluation employs an array of metrics: MSE, RMSE, mean absolute error (MAE) (6), and negative log-likelihood (7). The negative log-likelihood is germane to the loss function of the MDN. In employing the maximum likelihood estimation, this loss function operates on the premise that the sampled values possess the highest probability of occurrence. In alignment with ML norms, the focus lies on minimizing, not maximizing. Thus, a negative sign is appended, effectively transposing the objective to minimize the negative log-likelihood. Notably, the summation of likelihood functions from each distribution (M) underpins this process which is the total number of areas in the city.

The RMSE, MAE, and negative log-likelihood are justified evaluation metrics for our predictive models. The RMSE provides an overall measure of the prediction accuracy, while MAE captures average prediction errors. They possess intuitiveness and interpretability for numerical prediction. Moreover, the negative log-likelihood assesses the probabilistic predictions by measuring the logarithmic loss between the predicted probabilities and the actual outcomes. This metric is particularly useful when dealing with probabilistic predictions Using these metrics, we gain insights into the accuracy, precision, and probabilistic performance.

3.1.2. Fare Prediction

We evaluated the performance of the fare prediction model across various training and testing set sizes. To achieve this, we divided the dataset into different proportions for training and testing purposes. For instance, we reserved 30% of the entire dataset for testing, utilizing the remaining 70% for training. We repeated this process with nine

different testing set proportions: 10%, 20%, ..., 90%. Figure 5 displays the testing error in RMSE corresponding to these different testing set sizes.

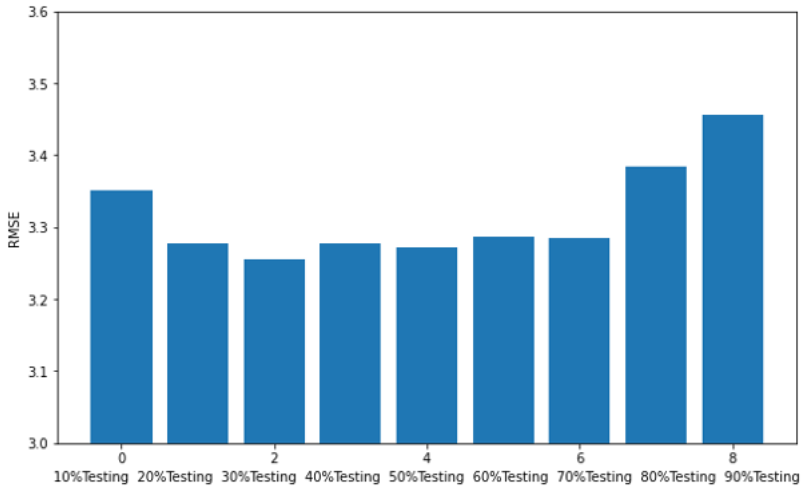


Figure 5. Fare prediction testing results under different testing set sizes.

The model performs best under 30% of the data being used as the testing set, as shown in the figure. The testing with 10%, 80%, and 90% testing sets yielded the worst results. In general, the overall performance of the fare prediction model is acceptable as the worst situations come from extreme cases.

We further evaluate our proposed model by comparing it to three baseline models, namely, LR, random forest (RF), and MLP. Table 1 shows the performance comparison of our proposed model and the baseline models.

Table 1. Performance comparison of the proposed model and baseline models.

Model	**Baseline**			**Proposed**
	LR	RF	MLP	**Ensemble Model**
RMSE for 14 features	0.32	0.59	0.25	0.098
RMSE for 6 features	3.61	3.44	3.33	3.26

It is important to note that the 14-feature model utilizes the original raw data, whereas the 6-feature model represents the data features our model can access after connecting to the front-end.

As shown in Table 1, our proposed ensemble model demonstrates enhanced performance in comparison to the baseline models when evaluating the 14-feature model. Additionally, a vertical comparison reveals that models focusing on only 6 features demonstrate inferior performance in contrast to those utilizing all 14 features. This observation suggests that the exclusion of certain features leads to an increase in errors and a decrease in the accuracy of the models' predictions. These findings strongly imply that the features within the dataset exhibit interdependencies, which significantly impact the accuracy of the models. In summary, our ensemble model consistently outperforms the other models in various analyses, solidifying its status as the top-performing model.

3.1.3. Demand Prediction

The effectiveness of the demand prediction model is presented below. In our experiment, the dataset is split into three groups: training, validation, and testing, with a ratio of 6:3:1. The training set is used during the training phase; the validation set is used for model selection and hyper-parameter tuning; the testing set is used for evaluation purposes.

Here, we conduct an independent evaluation of the proposed model, examining its performance with respect to both the LSTM and MLP-MDN models. For LSTM, we tested

three structures, namely, one layer, two layers, and three layers, with different numbers of hidden neurons and dropouts. Based on the testing results, the LSTM with a two-layer structure had the best performance. Therefore, we evaluate different numbers of hidden neurons and dropouts in the two-layer model. Table 2 shows the comparison of LSTM under different setups.

Table 2. Performance comparison of the LSTM model.

Method	**2-Layer** LSTM **without Dropout**		**2-Layer** LSTM **with 0.1 Dropout**	
	RMSE	MAE	RMSE	MAE
$n_1, n_2 = 100, 100$	4.7018	0.3104	4.8328	0.3474
$n_1, n_2 = 200, 200$	3.5504	0.2438	3.4462	0.3359
$n_1, n_2 = 400, 400$	3.3746	0.2413	3.4227	0.3370
$n_1, n_2 = 600, 600$	3.3106	0.2305	3.3106	0.3206
$n_1, n_2 = 800, 800$	3.6732	0.2598	3.6488	0.3335

As shown in the table, the LSTM with the $n1, n2 = 600, 600$ setup has the best performance. Furthermore, the inclusion of a dropout layer has minimal impact on the accuracy of the model. Here, we further compare our proposed LSTM model with four baseline models, namely, multilayer perceptron, ridge regression, lasso, and elastic net. We use the LSTM with $n_1, n_2 = 600, 600$ setup for the comparison as it has the best performance. Table 3 presents the comparison of them.

Table 3. Performance comparison between the LSTM model and baseline models.

Method	**RMSE**	**MAE**
LSTM (proposed)	3.3106	0.2305
Multilayer perceptron	3.4890	0.2962
Ridge regression	3.4224	0.2137
Lasso	3.4249	0.2164
Elastic net	3.4388	0.2283

Based on the presented results, it is evident that our proposed model exhibits superior performance when evaluated using the RMSE metric. What is more, our model shows comparable performance in terms of the MAE metric compared to the other baseline models. Despite not attaining the highest performance in the MAE metric, the noteworthy achievement in the RMSE underscores the resilience of our proposed model in handling outlier scenarios. This advantage is expected to be further amplified with the inclusion of additional data from diverse time ranges.

For MLP-MDN, we tested different numbers of hidden neurons. Table 4 presents a comparison of MDN under different setups. It is found that the model with the $n_1, n_2 = 200, 200, M = 40$ setup has the best performance.

Table 4. Performance comparison of the MDN model.

Method	**Neg. Log-Likelihood**
$n_1, n_2 = 100, 100, M = 20$	-3.7557
$n_1, n_2 = 100, 100, M = 40$	-3.7552
$n_1, n_2 = 100, 100, M = 80$	-3.7999
$n_1, n_2 = 200, 200, M = 20$	-3.7533
$n_1, n_2 = 200, 200, M = 40$	-3.8394
$n_1, n_2 = 200, 200, M = 60$	-3.7533

3.2. Interface and Use Cases

The demand interface is developed for NYC taxi drivers with three key features (i.e., identify the most demanded areas, highest payment zones, customer pick up and drop off trends). First, using the heatmap provided by the map, taxi drivers can pinpoint the areas with the most demand. The most popular zones were predicted historically in direct relationship to the number of pickups from travel data. The zone information at the top of the page on the right side of Figure 6A gives further summaries of the taxi ride for the zones. Figure 6B's line graph, which uses the minute as its unit of measurement, depicts the total demand for the following timestep as determined by users. Finally, Figure 6C shows a list of the top five demanded zones and a payment analysis to help taxi drivers determine the zones with the highest revenue levels. Taxi drivers can choose which zone to travel in using the interfaces in the following use cases:

1. Simply using a heatmap, point to the neighboring zone with the highest demand in the upcoming timestep. By moving in the direction of greater need, this situation is regarded as one of the quickest methods to gain a new client.
2. When neighboring zones have comparable demand, the top five demand zones are listed along with a payment analysis for the higher-income zone. Extending from above, a taxi driver could choose to travel to the area with the highest revenue while the other regions have a comparable demand. Consequently, integrating the heatmap, top zones list, and payment analysis can result in superior decision support with historical income for taxi drivers.
3. With a thorough evaluation of the demand heatmap, payment analysis, and trip summary for the maximum zone. Suppose there was a demand scenario on a Sunday between 8 and 8:30 p.m. Penn Station/Madison Sq West is the highest demand zone, and it has the information (designated in Figure 6A) of the average journey distance, speed, and duration. On Monday, the zone's average speed was around 10 miles per hour higher than typical. Combining the demand line chart (labeled as B in Figure 6), which displays the decreasing trend in the number of pickups and the higher trend in the number of dropoffs, provides proof that the taxi service is over-served. Label C denotes the highest earning zone, suggesting that drivers could be able to earn more money during the over-served hour.
4. By comparing the trip information in various timesteps, it is possible to estimate traffic jams using historical data (label A in Figure 6). The area's typical speed is revealed by the high average speed at night. There is presumed traffic congestion when the average speed of the trips in a particular zone is low and the average length is high.

In order to provide protection for passengers by providing a straightforward fare for all types of taxi services, including yellow taxis, green taxis, and Uber, a fare prediction interface has been created. After the user picks the destinations and the precise time, the fare forecast result is shown in Figure 4d. To choose the origin and destination, the user can enter them manually or click on the map. The nearest day with a 30 min interval in the provided DateTime in the history data or by the ML fare prediction is used to obtain the fare. The query also considers the duration and distance generated in real-time by the Google Maps API. Additionally, the uploading of taxi datasets is taken into account for the ease of data analysis for machine learning models and potential expansions. The dataset upload interface, which supports the format of CSV files with cleaned data, is also shown in Figure 4d. The CSV file is also placed into the database for real-time access to facilitate taxi demand and fare searching.

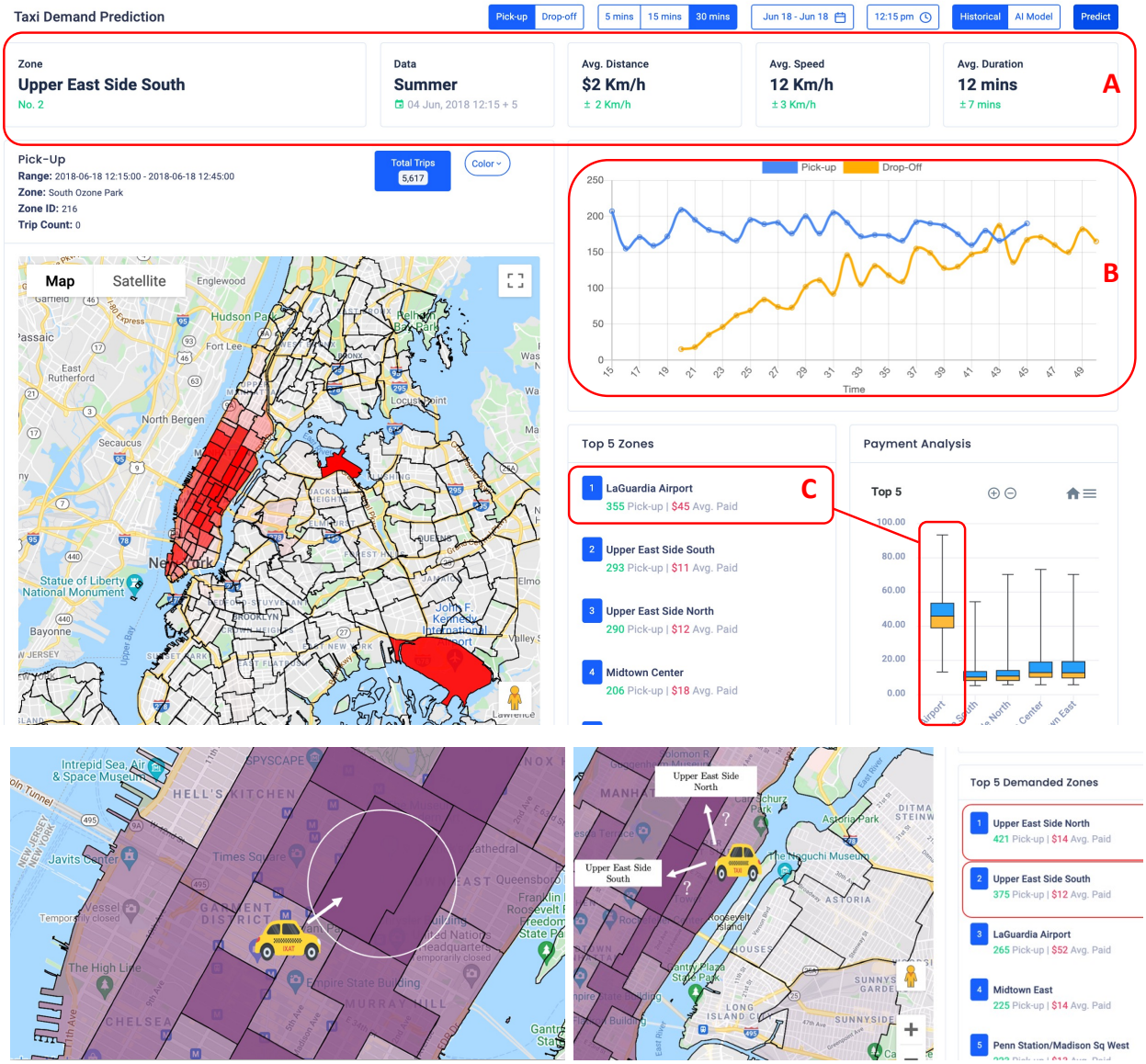


Figure 6. The demand prediction interface and use cases. The demand areas and income ratings to support the decision making for taxi drivers are stated on the left-hand side of the figure. (**A**) general information at a specific zone. (**B**) the pick and drop trend of a zone. (**C**) payment analysis of the top 5 zones.

4. Conclusions

In this research, we have constructed prediction models and successfully linked them with a web application to establish a comprehensive platform for viewing real-time taxi information and easing interactions between passengers and drivers in NYC. This taxi service system has an intuitive and interactive user interface, which has substantial consequences for three important areas.

Firstly, the platform offers drivers useful insights to maximize their profit and cut down on journey time. Drivers can utilize our system to gain access to vital data such as passenger demand trends (which pinpoints locations with strong demand at any given moment), proximity to potential passengers, estimated travel times to each customer, and projected profitability for each trip. With this information at hand, drivers are better equipped to choose passenger pickups wisely, maximizing their revenue effectively. Secondly, the platform also helps passengers by giving them access to crucial data that influences their choice of taxi service. They can locate platforms with reduced rates, decide

on the best pickup locations to reduce waiting periods (finding places where it is most likely that a taxi would arrive promptly), and obtain rough estimations of the cost of the trip. With this knowledge at hand, customers can decide which taxi to take with confidence, improving their overall experience and convenience. Furthermore, the platform gives government officials helpful insights on a larger scale. The government can learn more about a variety of facets of the city's taxi sector, including profit margins, current traffic conditions, and regions prone to regular traffic jams, by utilizing the system's richness of data. These revelations can help in the decision-making process for upcoming upgrades to city road infrastructure, resulting in more effective transportation systems and better urban planning.

In summary, our study has effectively created an interactive platform that is user-friendly and includes forecasting models for taxi demand and fare calculation. With the use of this platform, drivers, passengers, and government agencies can all make educated decisions, allocate resources more effectively, and improve overall efficiency in NYC's taxi ecosystem.

Author Contributions: Conceptualization, K.S.C.; demand and fare model, K.S.C. and K.L.W.; web application, K.S.C.; taxi data analysis, K.L.W.; formal analysis, K.S.C. and K.L.W.; system and model investigation, K.S.C. and K.L.W.; resource, R.T., S.-K.T. and G.P.; writing—original draft preparation, K.S.C., K.L.W. and B.Z.; writing—review and editing, K.S.C., K.L.W., B.Z., D.A., S.K.I., C.T.L., R.T., S.-K.T. and G.P.; supervision, S.K.I., C.T.L., R.T., S.-K.T. and G.P.; project administration, C.T.L., S.K.I., R.T., S.-K.T. and G.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported in part by the Macao Polytechnic University—Edge Sensing and Computing: Enabling Human-centric (Sustainable) Smart Cities (RP/ESCA-01/2020) and by the H2020 project titled "European Bus Rapid Transit of 2030: Electrified, Automated, Connected" EBRT—Grant Agreement No. 101095882.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhan, X.; Qian, X.; Ukkusuri, S. A Graph-Based Approach to Measuring the Efficiency of an Urban Taxi Service System. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2479–2489. [CrossRef]
2. Yang, H.; Fung, C.C.; Wong, K.F.; Wong, S.K. Nonlinear pricing of taxi services. *Transp. Res. Part A Policy Pract.* **2010**, *44*, 337–348. [CrossRef]
3. Zhang, R.; Ghanem, R. Demand, Supply, and Performance of Street-Hail Taxi. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 4123–4132. [CrossRef]
4. Chelliah, B.J.; Singh, J.; Chaturvedi, D.; Singh, A.K. Taxi fare prediction system using key feature extraction in artificial intelligence. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2021**, *12*, 3803–3808.
5. Santi, P.; Resta, G.; Szell, M.; Sobolevsky, S.; Strogatz, S.; Ratti, C. Quantifying the benefits of vehicle pooling with shareability networks. *Proc. Natl. Acad. Sci. USA* **2014**, *111*, 13290–13294. [CrossRef] [PubMed]
6. Liu, L.; Qiu, Z.; Li, G.; Wang, Q.; Ouyang, W.; Lin, L. Contextualized Spatial–Temporal Network for Taxi Origin-Destination Demand Prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 3875–3887. [CrossRef]
7. Yamaki, S. Study on Taxi Demand Prediction Using Context and Spatio-Tamporal Data. 2020. Available online: https://core.ac.uk/download/pdf/288814152.pdf (accessed on 30 March 2021).
8. Grinberg, J.; Jain, A.; Choksi, V. Predicting Taxi Pickups in New York City. 2014. Available online: http://www.vivekchoksi.com/papers/taxi_pickups.pdf (accessed on 5 August 2023).
9. Liu, T.; Wu, W.; Zhu, Y.; Tong, W. Predicting taxi demands via an attention-based convolutional recurrent neural network. *Knowl. Based Syst.* **2020**, *206*, 106294. [CrossRef]
10. Miao, F.; Han, S.; Lin, S.; Stankovic, J.; Zhang, D.; Munir, S. Taxi Dispatch with Real-Time Sensing Data in Metropolitan Areas: A Receding Horizon Control Approach. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 463–478. [CrossRef]

11. Moreira-Matias, L.; Gama, J.; Ferreira, M.; Mendes-Moreira, J.; Damas, L. Predicting Taxi–Passenger Demand Using Streaming Data. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1393–1402. [CrossRef]
12. Yuan, J.; Zheng, Y.; Zhang, L.; Xie, X.; Sun, G. Where to find my next passenger. In Proceedings of the 13th International Conference on Ubiquitous Computing, Beijing, China, 17–21 September 2011.
13. Zhang, C.; Zhu, F.; Wang, X.; Sun, L.; Tang, H.; Lv, Y. Taxi Demand Prediction Using Parallel Multi-Task Learning Model. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 794–803. [CrossRef]
14. Zhang, C.; Zhu, F.; Lv, Y.; Ye, P.; Wang, F.Y. MLRNN: Taxi Demand Prediction Based on Multi-Level Deep Learning and Regional Heterogeneity Analysis. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 8412–8422. [CrossRef]
15. Xu, J.; Rahmatizadeh, R.; Boloni, L.; Turgut, D. Real-Time Prediction of Taxi Demand Using Recurrent Neural Networks. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2572–2581. [CrossRef]
16. Wu, C.H.; Ho, J.W.; Lee, D.H. Travel-Time Prediction with Support Vector Regression. *IEEE Trans. Intell. Transp. Syst.* **2004**, *5*, 276–281. [CrossRef]
17. Yang, H.; Ye, M.; Tang, W.H.; Wong, S. Regulating taxi services in the presence of congestion externality. *Transp. Res. Part A Policy Pract.* **2005**, *39*, 17–40. .: 10.1016/j.tra.2004.05.004. [CrossRef]
18. Antoniades, C.; Fadavi, D.; Foba Amon, A., Jr. Fare and Duration Prediction: A Study of New York City Taxi Rides. 2016. Available online: http://cs229.stanford.edu/proj2016/report/AntoniadesFadaviFobaAmonJuniorNewYorkCityCabPricing-report.pdf (accessed on 5 August 2023).
19. Rossi, A.; Barlacchi, G.; Bianchini, M.; Lepri, B. Modelling Taxi Drivers' Behaviour for the Next Destination Prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 2980–2989. [CrossRef]
20. Upadhyay, R.; Lui, S. Taxi Fare Rate Classification Using Deep Networks. 2017. Available online: https://www.researchgate.net/publication/324706525_Taxi_Fare_Rate_Classification_Using_Deep_Networks (accessed on 5 August 2023).
21. Ferreira, N.; Poco, J.; Vo, H.; Freire, J.; Silva, C. Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2149–2158. [CrossRef]
22. TLC Trip Record Data. Available online: https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page (accessed on 10 August 2023).
23. Zhao, K.; Khryashchev, D.; Vo, H. Predicting Taxi and Uber Demand in Cities: Approaching the Limit of Predictability. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 2723–2736. [CrossRef]
24. Shu, P.; Sun, Y.; Zhao, Y.; Xu, G. Spatial-temporal taxi demand prediction using LSTM-CNN. In Proceedings of the 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), Hong Kong, China, 20–21 August 2020; pp. 1226–1230.
25. Guo, X. Prediction of taxi demand based on CNN-BiLSTM-attention neural network. In Proceedings of the Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, 23–27 November 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 331–342.
26. Duan, Z.T.; Kai, Z.; Yun, Y.; Ni, Y.Y.; Bajgain, S. Taxi demand prediction based on CNN-LSTM-ResNet hybrid depth learning model. *J. Transp. Syst. Eng. Inf. Technol.* **2018**, *18*, 215.
27. Cao, D.; Zeng, K.; Wang, J.; Sharma, P.K.; Ma, X.; Liu, Y.; Zhou, S. BERT-Based Deep Spatial-Temporal Network for Taxi Demand Prediction. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 9442–9454. [CrossRef]
28. Li, Y.; Moura, J.M. Forecaster: A graph transformer for forecasting spatial and time-dependent data. *arXiv* **2019**, arXiv:1909.04019.
29. Kim, T.; Sharda, S.; Zhou, X.; Pendyala, R.M. A stepwise interpretable machine learning framework using linear regression (LR) and long short-term memory (LSTM): City-wide demand-side prediction of yellow taxi and for-hire vehicle (FHV) service. *Transp. Res. Part C Emerg. Technol.* **2020**, *120*, 102786. [CrossRef]
30. Django. The Web Framework for Perfectionists with Deadlines. 2019. Available online: https://www.djangoproject.com/ (accessed on 5 August 2023).
31. Flask. Welcome to Flask—Flask Documentation (2.3.x). Available online: https://flask.palletsprojects.com/en/2.3.x/ (accessed on 5 August 2023).
32. Google Maps Platform-Location and Mapping Solutions. Available online: https://mapsplatform.google.com/ (accessed on 5 August 2023).
33. ApexCharts.js—Open Source JavaScript Charts for Your Website. Available online: https://apexcharts.com/ (accessed on 5 August 2023).
34. Chart.js | Open Source HTML5 Charts for Your Website. 2019. Available online: https://www.chartjs.org/ (accessed on 5 August 2023).
35. Otto, M. Bootstrap. 2022. Available online: https://sites.google.com/view/bootstrap2022/ (accessed on 5 August 2023).
36. Amazon. Amazon Web Services (AWS)—Cloud Computing Services. 2023. Available online: https://pages.awscloud.com/AWS-Innovators-Amazon.html (accessed on 5 August 2023).
37. Deploying a Django Application to Elastic Beanstalk. AWS Elastic Beanstalk. Available online: https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/create-deploy-python-django.html (accessed on 5 August 2023).