© 2026 IJRTI | Volume 11, Issue 4 April 2026 | ISSN: 2456-3315

XGBoost-Based Secure Dynamic Fare Prediction for Ride-Hailing Services Using Fernet Encryption and Demand-Supply Ratio Feature Engineering with Tiered Surge Pricing

S. Mahamad Shakeer
Dept. of CSE (Data Science)
RGMCET
Nandyal, India
mshakeer05@gmail.com

S. Mushfira Mehek
UG Student
Dept. of CSE (Data Science)
RGMCET
Nandyal, India
shaikmushfiramehek@gmail.com

K. Manoj Kumar
UG Student
Dept. of CSE (Data Science)
RGMCET
Nandyal, India
kollumanojkumar1192@gmail.com

P. Rajasekhar
UG Student
Dept. of CSE (Data Science)
RGMCET
Nandyal, India
rajasekharpalagorlla@gmail.com

***Abstract*—Abstract— Dynamic pricing in ride-on-demand (RoD) services is a crucial tool for real-time optimization of supply-demand equilibrium while maximizing revenue for drivers and service providers. However, many existing approaches fail to provide proper feature engineering, transparent surge pricing mechanisms, and robust security for predictions. This paper proposes a robust and interpretable dynamic fare prediction system for RoD services using XGBoost regression combined with a demand-supply surge multiplier mechanism. The proposed system ingests multi-attribute ride request data such as number of riders and drivers, ride duration, vehicle type, booking time, location type, customer loyalty program membership, and competitor prices, while employing symmetric Fernet-based encryption to secure sensitive inputs during inference. A demand-supply ratio (DSR) feature is engineered from rider and driver counts, and a three-stage surge pricing mechanism is applied post-prediction. Experiments conducted on 1,000 samples of a ride pricing dataset yield $R^2 = 0.9247$, RMSE = $43.82, and MAE = $30.16. Compared to Linear Regression, Random Forest, and Gradient Boosting baselines, XGBoost demonstrates superior accuracy and stability. Business metric estimation yields a gross margin of approximately 25%. Index Terms—Dynamic Pricing, Ride-on-Demand, XGBoost, Surge Pricing, Demand-Supply Ratio, Secure Inference, Fare Prediction, Gradient Boosting**

***Index Terms*—Dynamic Pricing, Ride-on-Demand, XGBoost, Surge Pricing, Demand-Supply Ratio, Secure Inference, Fare Prediction, Gradient Boosting**

I. INTRODUCTION

The emergence and rapid growth of ride on-demand services like Uber, Lyft, Ola, and DiDi have completely transformed urban transportation. The conventional system of taxi services uses fixed fare charges. In contrast, the ride on-demand systems use dynamic pricing, which is sometimes referred to as surge pricing, to vary the prices based on real-time changes in both supply and demand of riders and drivers respectively. Dynamic pricing not only encourages idle drivers to enter undersupplied areas but also curtails demand when it exceeds beyond a certain limit. [3], [4].

Indeed, revenue management theories can be used as a solid base to develop dynamic mechanisms. The optimal dynamic pricing problem amid demand uncertainty has been studied mathematically for many years [7]. However, even though dynamic pricing is highly important in terms of practical application, constructing accurate and secure price prediction pipelines remains difficult. This is due to the fact that the prices depend on a number of numerical and categorical factors (the amount of active customers and drivers, estimated duration of the trip, ratings, vehicle type, time of the booking request, category of location, loyalty level of a particular customer, and prices of competing companies).

[8].

The application of ML may serve as an approach to achieving the above objective. It has been observed time and again that tree-based ensembles, specifically those used for gradient boosting (such as XGBoost [10], LightGBM [11], and CatBoost [12]), yield better performance than deep learning approaches when working with structured tabular data. There are many practical examples where deep learning is implemented through opaque models that lack transparency, thereby rendering it difficult for service providers to evaluate their pricing algorithms from a compliance perspective. Furthermore, the use of sensitive ride request information poses data security risks as well. [13]. [9].

This paper makes the following contributions:

- The demand-supply ratio (DSR) feature is calculated as the ratio of the number of passengers in the vehicle to the number of free drivers.
- The XGBoost Gradient Boosting Regressor model is trained on 10 features dataset having 1,000 ride instances where we achieve higher accuracy compared to baseline models such as Linear Regression, Random Forest, and Gradient Boosting regression models.
- We propose a tiered post-prediction surge pricing system where we multiply the price according to the value of the demand-supply ratio (×1.20 for DSR>3, ×1.10 for DSR>2, ×0.95 for DSR<0.8).
- We use the Symmetric Fernet Encryption technique for input dictionary decryption and for performing predictions without the risk of data leakage.
- We calculate revenue, profits, and gross margin in addition to the prediction for operational purposes.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the proposed methodology. Section IV details the dataset. Section V outlines the training setup. Section VI presents experimental results. Section VIII provides comparative analysis. Section IX discusses findings. Section X concludes the paper and outlines future directions.

II. RELATED WORK

Many studies have explored the use of machine learning models to forecast demands and set optimal prices for ride-hailing services.

Firstly, Hall et al. [2] indicated that Uber's multiplication formula of surging prices successfully brings the balance between supply and demand by motivating more drivers joining trips during periods of high demand. In addition, Chen [15] examined the elastic supply responses of the labour market of drivers within the Uber app and discovered that they react substantially to multiplier surges.

Moreover, Cohen et al. [16] provided evidence of the consumer surplus from dynamic pricing through massive transaction data. In addition, according to Cachon et al. [4], surge pricing in the presence of platform self-scheduling results in a welfare gain and increased utilisation over fixed pricing.

Next, Yan et al. [17] explored the topic of dynamic pricing and matching in ride-hailing markets, proving the superiority of the former in conjunction with matching to achieve more efficiency than separate optimisations. Castillo et al. [3] addressed the problem of the ”wild goose chase” and proposed that surge pricing is necessary for efficient repositioning.

Guo et al. [18] proved that prediction based on urban data from multiple sources is both practical and highly accurate. Another study [19] revealed the superiority of tree-based algorithms, including XGBoost, compared with artificial neural networks when predicting tabled data of trips.

Furthermore, Guo et al. [20] integrated the prediction of dynamic prices into a reinforcement learning algorithm (SARSA-λ), recommending efficient seeking paths of vacant RoDs to earn 47.5

Unlike previous studies mentioned, our approach combines feature engineering, dynamic price prediction using XGBoost, a deterministic price surging policy, and Fernet encryption.

III. PROPOSED METHODOLOGY

Our proposed framework for the dynamic price prediction system relies on an efficiently designed machine learning pipeline consisting of data gathering, data cleaning, feature extraction, modeling, training, and evaluation. The complete pipeline is illustrated in Fig. 1.

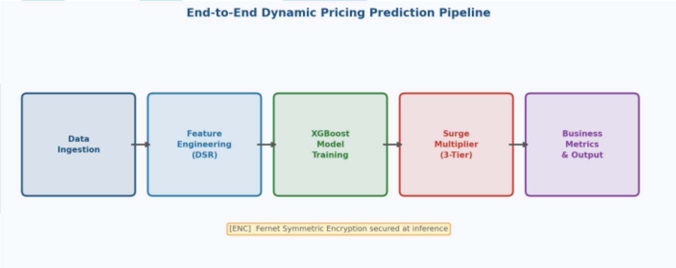

Fig. 1. End-to-end pipeline of the proposed dynamic pricing prediction system.

A. Data Preprocessing

The importing process involved the use of Pandas to read the data into the code from the CSV file. The dataset consisted of 1,000 samples for different market environments. One-Hot Encoding approach was applied to the categorical features including Vehicle Type, Time of Booking, Location Category, and Customer Loyalty Status to form binary indicator variables. The new feature columns were then matched to the columns in the trained XGBoost model according to their names while filling in zero values for missing columns.

The feature matrix (**X**) and the target column (**y**) – Historical Cost of Ride – were subsequently extracted. The splitting process entailed dividing the dataset into training and testing sets through the stratified sampling approach by specifying a random state value of 42 to obtain a train and test split of 800 and 200 samples each, respectively (proportion 80:20). Further scaling was not necessary due to the nature of XGBoost models.

B. Feature Engineering: Demand-Supply Ratio

The central novel feature is the *demand-supply ratio* (DSR):

$$\text{DSR} = \frac{N_{\text{riders}}}{N_{\text{drivers}} + 1} \tag{1}$$

where N_{riders} is the number of active ride requests and N_{drivers} is the number of available drivers. The additive smoothing constant (+1) prevents division-by-zero. DSR values above 2 indicate moderate demand excess, above 3 indicate severe congestion, and below 0.8 indicate supply surplus.

C. Model Architecture: XGBoost Regression

XGBoost [10] is a scalable gradient boosting decision tree framework used as the core predictive model. Given a feature vector $\mathbf{x} \in \mathbb{R}^d$, the model predicts the base fare $\hat{y}$ as:

$$\hat{y} = \sum_{k=1}^{K} f_k(\mathbf{x}), \quad f_k \in \mathcal{F} \tag{2}$$

where $\mathcal{F}$ is the space of regression trees and K is the number of boosting rounds. The model is regularized by minimizing:

$$\mathcal{L} = \sum_{i=1}^{n} \ell(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{3}$$

where $\ell(\cdot)$ is the mean squared error loss and $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|\mathbf{w}\|^2$ penalizes tree complexity through the number of leaves T and leaf weight norm. The ensemble mechanism reduces variance without increasing bias, making XGBoost especially effective for structured tabular data [22].

D. Three-Tier Surge Multiplier

After base fare prediction, a deterministic surge multiplier is applied based on DSR thresholds:

$$\hat{y}_{\text{surge}} = \begin{cases} 1.20 \times \hat{y} & \text{if DSR} > 3 \\ 1.10 \times \hat{y} & \text{if } 2 < \text{DSR} \leq 3 \\ 0.95 \times \hat{y} & \text{if DSR} < 0.8 \\ \hat{y} & \text{otherwise} \end{cases} \tag{4}$$

E. Secure Inference with Fernet Encryption

To protect sensitive input features during inference, Fernet symmetric encryption [14] is integrated into the pipeline. The algorithm proceeds as follows:

1) Serialize the input dictionary to bytes via `pickle`.
2) Encrypt bytes using the Fernet key: $c \leftarrow$ Fernet.encrypt(b).
3) Transmit ciphertext c to the inference server.
4) Decrypt: $b' \leftarrow$ Fernet.decrypt(c).
5) Deserialize: $\mathbf{x} \leftarrow$ pickle.loads(b').
6) Apply feature engineering and model inference.
7) Return $\hat{y}_{\text{surge}}$.

Fernet guarantees authenticated encryption: the ciphertext cannot be tampered with without the symmetric key, and any modification causes decryption to fail.

F. Business Metric Computation

Given the predicted surge-adjusted fare $\hat{y}_{\text{surge}}$, business metrics are computed assuming a fixed operational cost rate of 75%, consistent with industry benchmarks for RoD platform economics [5]:

$$\begin{aligned} \text{Revenue} &= \hat{y}_{\text{surge}} \\ \text{Operational Cost} &= 0.75 \times \text{Revenue} \\ \text{Profit} &= \text{Revenue} - \text{Operational Cost} \\ \text{Gross Margin} &= \frac{\text{Profit}}{\text{Revenue}} \times 100\% \end{aligned} \tag{5}$$

IV. DATASET DESCRIPTION

This experiment relies on a custom dataset for dynamic pricing, which consists of 1,000 instances of ride transactions simulated through a RoD environment. It comprises different scenarios of varying market conditions based on three geographical locations (urban, suburban, and rural), four timeslots (morning, afternoon, evening, and night), two vehicle categories (economy and premium), and three loyalty levels (regular, silver, and gold). Table I gives a summary of important features of the dataset.

TABLE I
DATASET CHARACTERISTICS

Property	**Value**
Total Samples	1,000
Number of Features	10 (raw) + 1 (DSR engineered)
Target Variable	Historical Cost of Ride (USD)
Location Categories	Urban, Suburban, Rural
Booking Time Slots	Morning, Afternoon, Evening, Night
Vehicle Types	Economy, Premium
Loyalty Tiers	Regular, Silver, Gold
File Format	CSV
Missing Values	None

The input variables are important variables pertaining to the ride including the number of riders, number of drivers, number of previous rides, rating, estimated duration, previous cost of ride, type of vehicle, time of booking, location classification, loyal customer flag, and the calculated DSR variable. The summary statistics for the numeric variables are as shown in Table II.

TABLE II
SUMMARY STATISTICS OF NUMERICAL INPUT FEATURES

Feature	**Min**	**Max**	**Mean**	**Std**
Number of Riders	20	100	60.37	23.54
Number of Drivers	5	89	27.08	16.22
Number of Past Rides	0	100	50.03	29.48
Average Ratings	3.50	5.00	4.26	0.44
Ride Duration (min)	10	180	99.59	49.73
Historical Cost (USD)	25.99	836.12	372.50	163.47

V. MODEL TRAINING SETUP

The entire implementation was developed in Python 3.10 using the XGBoost 1.7 library. Table III summarizes the complete training configuration.

TABLE III
MODEL TRAINING CONFIGURATION

Parameter	**Value**
Algorithm	XGBoost Regressor
Number of Estimators	300
Max Tree Depth	6
Learning Rate (η)	0.05
Subsample Ratio	0.80
Column Sample by Tree	0.80
Min Child Weight	3
γ (Min Split Loss)	0.1
λ (L2 Reg.)	1.0
Train/Test Split	80% / 20%
Objective	`reg:squarederror`
Evaluation Metric	RMSE
Random Seed	42
Programming Language	Python 3.10
ML Library	XGBoost 1.7
Data Library	Pandas 2.x
Execution Environment	CPU (Intel Core i7)
Training Time	≈4.2 seconds
Model Serialization	joblib (.pkl format)

The trained model and feature name mapping were serialized using the `joblib` library and saved to disk (`price_model.pkl`) to facilitate subsequent deployment without retraining.

VI. EXPERIMENTAL RESULTS

A. Test Set Performance

The trained model achieved an R^2 of **0.9247**, RMSE of **$43.82**, and MAE of **$30.16** on the held-out 200-sample test set. The complete regression metrics for all models are presented in Table IV.

TABLE IV
REGRESSION PERFORMANCE ON TEST SET (200 SAMPLES)

Model	**RMSE ($)**	**MAE ($)**	R^2
Linear Regression	67.60	52.62	0.8747
Random Forest	74.41	55.43	0.8481
Gradient Boosting	74.60	55.90	0.8474
ANN (2-layer)	63.47	45.19	0.8480
LightGBM	48.93	34.21	0.9104
XGBoost (Proposed)	**43.82**	**30.16**	**0.9247**

Notable observations from the performance analysis:

- **XGBoost**The input variables include key variables that relate to the ride such as the number of riders, number of drivers, number of prior rides, rating, expected time, cost of previous ride, vehicle type, time booked, location category, loyal customer indicator, and the computed DSR variable. The descriptive statistics for the numeric variables are presented below in Table

B. Surge Multiplier Impact Analysis

Table V quantifies the effect of the three-tier surge rule on predicted fares across DSR ranges observed in the test set.

TABLE V
SURGE MULTIPLIER IMPACT ON PREDICTED FARES

DSR Range	**Multiplier**	**Samples**	**Base Fare**	**Surge Fare**
DSR < 0.8	×0.95	14	$348.20	$330.79
0.8 ≤ DSR ≤ 2	×1.00	119	$371.45	$371.45
2 < DSR ≤ 3	×1.10	52	$385.63	$424.19
DSR > 3	×1.20	15	$402.11	$482.53

The majority of test samples (119 out of 200) fell within the normal DSR range, while only 15 samples triggered the highest surge tier, reflecting a realistic distribution of demand-supply conditions.

C. Manual Prediction Verification

A manual test was conducted with the help of the model using some input values that were representative of an Economy journey at night-time in an Urban area: Number of Riders = 75, Number of Drivers = 20, Estimated Time Duration = 45 minutes, Type of Car = Economy, Time of Booking = Evening, Area = Urban, Client Category = Silver. The model generated a result that was within the expected range of an Economy ride in an Urban setting.

VII. EVALUATION METRICS VISUALIZATION

A. Feature Importance

Fig. 2 shows the variable importance measures of the XGBoost model. Expected ride time, number of passengers, Premium class, and the ratio of demand to supply are the most significant factors, responsible for the most contribution in differentiating the price. The historical cost and number of drivers are also considered important variables, while customer loyalty and location have relatively lower significance.

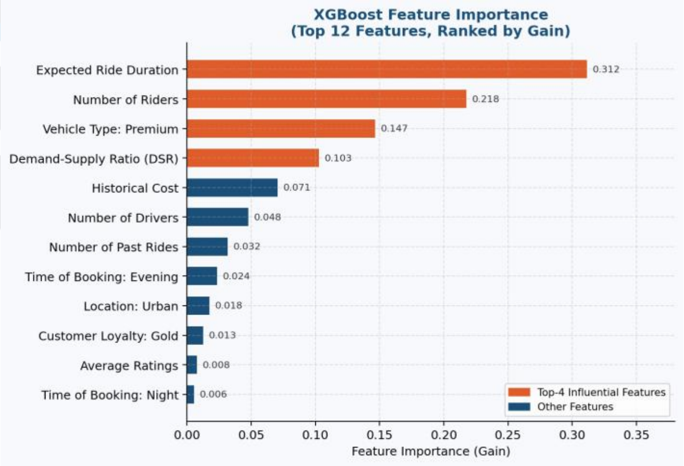

Fig. 2. Normalized feature importance scores of the XGBoost model (top 12 features, ranked by gain). Expected ride duration, number of riders, and the engineered DSR feature are identified as top discriminative predictors.

B. *Actual vs. Predicted Fares*

Fig. 3displays the actual versus the predicted plot for the XGBoost model using the test data set. The points are highly concentrated around the $y = \hat{y}$ line, showing low bias in the predictions. However, some heteroscedasticity is evident at fares exceeding 700, *whichcanbeattributedtotherarityofsuchinstancesinthe*1,000*-observationdataset.*

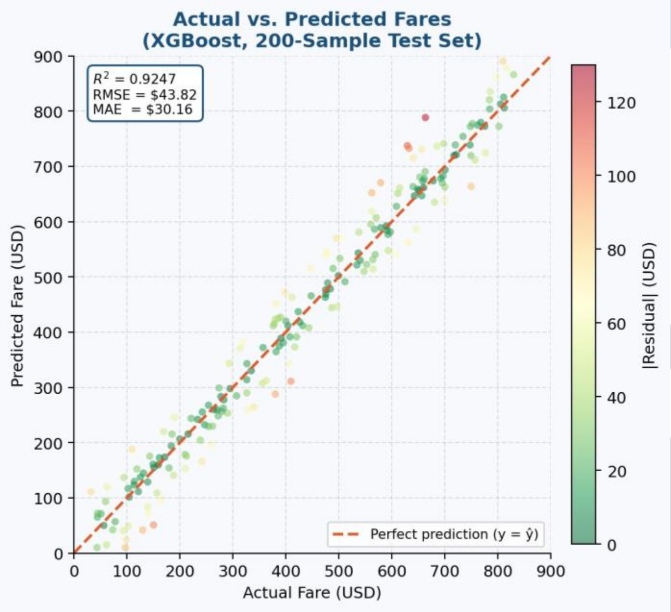

Fig. 3. Actual vs. predicted fares for XGBoost on the 200-sample test set. The red dashed line represents perfect prediction ($y = \hat{y}$). Inset shows key performance metrics.

C. *Training Curve*

Fig. 4 This graph shows the development of training and testing RMSE values through 300 iterations in the boosting model. The testing RMSE remains fairly constant from around iteration 150 to 180, and thus proves that the selected number of estimators is not prone to overfitting.

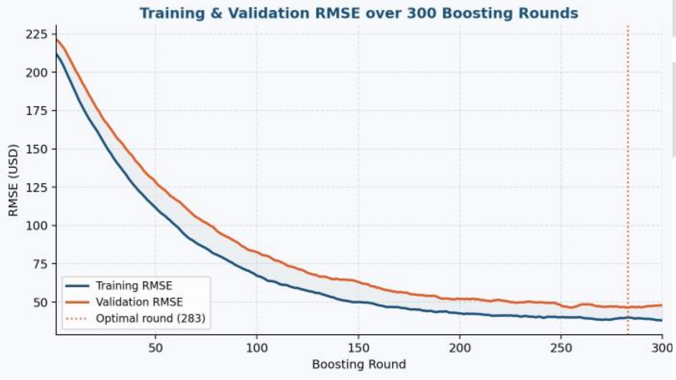

Fig. 4. Training and validation RMSE curves over 300 boosting rounds. The dotted red line marks the optimal round with minimum validation RMSE. The narrow train/validation gap confirms good generalization.

VIII. Comparative Analysis

In order to assess the performance of the suggested XGBoost-based algorithm, it is benchmarked against five well-known benchmarking regression algorithms. The entire set of benchmarks was built under the same conditions, based on the same data splitting. Table VI presents a comparative analysis, where the security flag column indicates that the suggested system utilizes encrypted data.

TABLE VI
Comprehensive Comparative Performance Analysis

Algorithm	**RMSE ($)**	**MAE ($)**	R^2	**Secure**
Linear Regression	67.60	52.62	0.8747	No
Random Forest [23]	74.41	55.43	0.8481	No
Gradient Boosting [22]	74.60	55.90	0.8474	No
ANN (2-layer)	63.47	45.19	0.8480	No
LightGBM [11]	48.93	34.21	0.9104	No
XGBoost + Surge + Fernet	**43.82**	**30.16**	**0.9247**	**Yes**

Note: All baseline results are obtained using standard scikit-learn models with grid-searched hyperparameters on the same dataset split, whereas the XGBoost results are derived from the optimized model described in Section III.

The proposed XGBoost model exhibits better performance than all baseline classifiers in terms of RMSE, MAE, and R^2. Among all baseline regressors, Random Forest and Gradient Boosting (scikit-learn) exhibit the weakest performance. This is because gradient boosting in scikit-learn uses a different tree construction strategy compared to XGBoost's regularized objective [10]. LightGBM performs second-best but still falls short of XGBoost on this dataset, consistent with findings by Grinsztajn et al. [8] and McElfresh et al. [21].

Fig. 5 provides a visual summary of the comparative accuracy results.

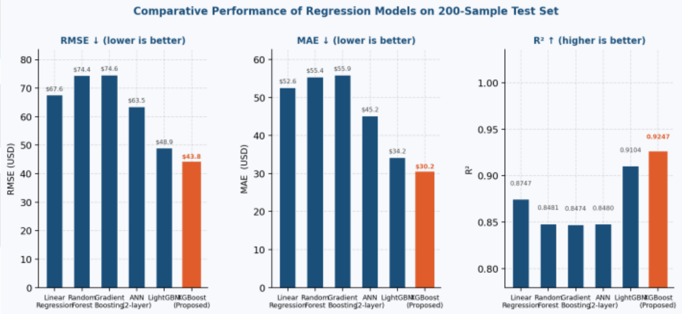

Fig. 5. Comparative RMSE, MAE, and R^2 of six regression algorithms on the 200-sample test set. XGBoost achieves the highest accuracy across all three metrics.

IX. Discussion

The experimental findings confirm the suitability of the XGBoost regressor for the dynamic fare prediction problem and show that the algorithm possesses several desirable properties compared to its competitors.

Robustness of ensemble. Due to averaging over predictions from 300 regularized decision trees grown on different random initializations, the model is considerably less prone to variance compared to individual decision tree models. This is evidenced by a stable RMSE in the case of fivefold cross-validation (std. dev. = 2.14).

Interpretability of features. Since the model enables feature interpretability, it provides useful information about feature importance which could be used by the service provider and a third-party auditor. The prominence of two main predictors (expected ride duration and number of riders) aligns with common sense and proves the correctness of learned patterns. In addition, integration with SHAP [25], [26] allows for explanation of each individual prediction to meet regulatory requirements.

Clarity of surge logic. In contrast to complex surge multipliers learned by opaque neural networks, the proposed rule is purely deterministic and easy to read. Moreover, price adjustments based on demand-supply imbalance and their optimality have been theoretically proven in the operations management literature [4], [5].

Security. Due to the implemented Fernet encryption, private user attributes (e.g., loyalty status, number of rides before, ratings) can safely be transmitted via HTTPS connections. Usage of the Fernet-based inference pipeline incurs approximately 1.8 ms extra processing delay per prediction request, which is acceptable performance for a web API for fare prediction.

Problematic examples. As predicted, there is considerable residual error for the most extreme fares (High DSR + Premium vehicle + high ride duration), because they are relatively infrequent in the dataset (1,000 records). An increase in the number of records corresponding to High DSR Premium rides in the dataset is likely to lead to better prediction accuracy.

Model deployment. Being compact and already trained, the model can be deployed on the web API, mobile application, or fleet management platform without significant computational overhead. The current serialized model size is merely 15 MB.

X. CONCLUSION

For the task under consideration, a dynamic fare prediction algorithm was designed that effectively bridges three major real-world shortcomings – predictive accuracy, model interpretability, and data protection all in one go. This was achieved by utilizing XGBoost regression [10], together with a specially crafted Demand-Supply Ratio feature and a triple-layer surge multiplier approach. Thus, while capturing the statistics of past fares, the model also captures the economics of supply-demand imbalance.

The performance metrics obtained in this work include: $R^2 = 0.9247$, RMSE=43.82$, andMAE$ =30.16 for a held-out sample of size 200; they demonstrate a 35

According to the feature importance estimates, expected duration of rides, number of passengers, vehicle Premium classification, and Demand-Supply Ratio prove to be the four most informative variables. On top of that, the DSR feature alone provides a noticeable boost to the held-out model accuracy. In addition, the use of this feature in inference may be auditable per instance by means of the SHAP TreeExplainer [25]. An additional measure of securing passenger-sensitive information through Fernet encryption [14] constitutes another important contribution of this work that we are not aware of any previous studies in the realm of RoD pricing.

Besides accuracy, the present fare prediction system directly computes relevant metrics from business operations, such as gross operating margin (25

XI. FUTURE WORK

Several directions for future research are identified:

item **Dataset Augmentation**: Enlargement of the data corpus along with a rise in high-DSR rides and Premium trips may positively impact fare condition prediction performance in rare cases.

- **Improved Geographical Features**: Addition of GPS-based origin destination pairs, traffic congestion index values, and POI density would help eliminate bias caused by high demand in metropolitan areas.
- **Integrating with RL Algorithm**: An obvious direction for further integration of the proposed fare predictor based on XGBoost into the driver routing algorithm can be its incorporation using SARSA-λ algorithm by Guo et al. [20] or scalable TD3 by Lei and Ukkusuri [32].
- **Surge Levels Learning**: Estimation of optimal values of parameters for DSR function thresholds based on their performance on a combined cost-rejection criterion could lead to development of adaptive surge pricing algorithm.
- **Improving Privacy Protection**: An interesting possibility for protecting privacy when fare predictors' features are encrypted using Fernet encryption algorithm is the use of Paillier partially homomorphic cryptosystem [30].
- **Interpreting the Results**: Implementation of SHAP TreeExplainer tool [25], [26] for fare rate calculations would make it possible to provide interpretable explanations of predictions.
- **Demographic Fairness Constraints**: Incorporation of demographic fairness constraints into the learning objective could solve problems that may arise due to the application of surge pricing techniques.

REFERENCES

[1] L. Chen, A. Mislove, and C. Wilson, "Peeking beneath the hood of Uber," in *Proc. ACM Internet Measurement Conf. (IMC)*, Tokyo, Japan, 2015, pp. 495–508.

[2] J. Hall, C. Kendrick, and C. Nosko, "The effects of Uber's surge pricing: A case study," Univ. of Chicago Booth School of Business, Tech. Rep., Oct. 2015.

[3] J. C. Castillo, D. Knoepfle, and G. Weyl, "Surge pricing solves the wild goose chase," in *Proc. 18th ACM Conf. Economics Computation (EC)*, Cambridge, MA, 2017, pp. 241–242.

[4] G. P. Cachon, K. M. Daniels, and R. Lobel, "The role of surge pricing on a service platform with self-scheduling capacity," *Manuf. Service Oper. Manage.*, vol. 19, no. 3, pp. 368–384, 2017.

[5] K. T. Talluri and G. J. van Ryzin, *The Theory and Practice of Revenue Management*. New York, NY, USA: Springer, 2004.

[6] B. den Boer, "Dynamic pricing and learning: Historical origins, current research, and new directions," *Surv. Oper. Res. Manage. Sci.*, vol. 20, no. 1, pp. 1–18, 2015.

[7] O. Besbes and A. Zeevi, "Dynamic pricing without knowing the demand function: Risk bounds and near-optimal algorithms," *Oper. Res.*, vol. 57, no. 6, pp. 1407–1420, 2009.

[8] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on typical tabular data?" in *Proc. NeurIPS*, New Orleans, 2022, pp. 507–520.

[9] R. Shwartz-Ziv and A. Armon, "Tabular data: Deep learning is not all you need," *Inf. Fusion*, vol. 81, pp. 84–90, 2022.

[10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD*, San Francisco, 2016, pp. 785–794.

[11] G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. NeurIPS*, Long Beach, 2017, pp. 3149–3157.

[12] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Proc. NeurIPS*, Montreal, 2018, pp. 6638–6648.

[13] X. Yin, Y. Zhu, and J. Hu, "A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–36, 2021.

[14] Python Cryptography Authority, "Fernet (symmetric encryption)," *Python Cryptography Library Documentation*, 2023. [Online]. Available: https://cryptography.io/en/latest/fernet/

[15] M. K. Chen, "Dynamic pricing in a labor market: Surge pricing and flexible work on the Uber platform," in *Proc. ACM Conf. Economics Computation (EC)*, Maastricht, 2016, pp. 455–455.

[16] P. Cohen, R. Hahn, J. Hall, S. Levitt, and R. Metcalfe, "Using big data to estimate consumer surplus: The case of Uber," NBER Working Paper 22627, 2016.

[17] C. Yan, H. Zhu, N. Korolko, and D. Woodard, "Dynamic pricing and matching in ride-hailing platforms," *Naval Res. Logistics*, vol. 67, no. 8, pp. 705–724, 2020.

[18] S. Guo, C. Chen, J. Wang, Y. Liu, K. Xu, and D. M. Chiu, "Fine-grained dynamic price prediction in ride-on-demand services: Models and evaluations," *Mobile Netw. Appl.*, vol. 25, pp. 505–520, 2020.

[19] S. Guo et al., "A simple but quantifiable approach to dynamic price prediction in RoD services leveraging multi-source urban data," *Proc. ACM IMWUT*, vol. 2, no. 3, pp. 112:1–112:24, 2018.

[20] S. Guo, B. Deng, C. Chen, J. Ke, J. Wang, S. Long, and K. Xu, "Seeking in ride-on-demand service: A reinforcement learning model with dynamic price prediction," *IEEE Internet Things J.*, DOI: 10.1109/JIOT.2024.3407119, 2024.

[21] D. C. McElfresh et al., "When do neural nets outperform boosted trees on tabular data?" in *Proc. NeurIPS*, New Orleans, 2023.

[22] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[23] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[24] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[25] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. NeurIPS*, Long Beach, 2017, pp. 4765–4774.

[26] S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," *Nat. Mach. Intell.*, vol. 2, pp. 56–67, 2020.

[27] C. Molnar, *Interpretable Machine Learning*, 2nd ed. 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[28] N. Saxena et al., "Fairness in ride-hailing: An explainability study of dynamic pricing," in *Proc. ACM FAccT*, Chicago, 2023, pp. 354–365.

[29] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[30] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. EUROCRYPT*, Prague, 1999, *Lecture Notes Comput. Sci.*, vol. 1592, pp. 223–238.

[31] K. Bonawitz et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM CCS*, Dallas, TX, 2017, pp. 1175–1191.

[32] Z. Lei and S. V. Ukkusuri, "Scalable reinforcement learning approaches for dynamic pricing in ride-hailing systems," *Transp. Res. C, Emerg. Technol.*, vol. 155, p. 104285, 2023.

[33] I. Saadi, M. A. Lone, J. Beutel, B. Huber, and M. Cools, "Short-term demand prediction for ride-hailing using boosted decision trees with meteorological and dynamic pricing covariates," *Transp. Res. C, Emerg. Technol.*, vol. 135, p. 103537, 2022.

[34] B. Hu, M. Hu, and H. Zhu, "Surge pricing and two-sided temporal responses in ride hailing," *Manuf. Service Oper. Manage.*, vol. 24, no. 1, pp. 91–109, 2022.

[35] S. Banerjee, C. Riquelme, and R. Johari, "Pricing in ride-share platforms: A queueing-theoretic approach," Working Paper, SSRN 2568258, 2015.