Order acceptance choice modeling of crowd-sourced delivery services: a systematic comparative study

Shixuan Hou

1Concordia University

October 31, 2023

Abstract

The efficiency of crowd-sourced delivery services (CDS) like UberEats and AmazonFlex highly depends on the decisions of individual shippers. Operating as freelancers, these shippers have the freedom to accept or decline orders from the CDS platform. Their decisions not only affect their earnings and the waiting times for orders but also influence the platforms' overall revenue and reputation. Understanding the factors that shape these decisions is thus crucial. In our study, we gather data from CDS shippers in Shanghai, China, using stated preference surveys. We then design a discrete choice model to predict shippers’ behaviors and compare its accuracy, computational efficiency, and interpretability with five commonly used machine learning methods. Our analyses reveal that the Extreme Gradient Boosting (XGB) model and Random Forests (RFs) model outperform other models in prediction accuracy, achieving f1 scores of 69.3% and 65% respectively. Notably, our per-mutation importance analysis indicate that the shipper’s age, income, and the compensation awarded per order are the most influential determinants in their decision to accept or decline orders.
Order acceptance choice modeling of crowd-sourced delivery services: a systematic comparative study

Shixuan Hou, Jie Gao, and Chun Wang Member, IEEE

Abstract—The efficiency of crowd-sourced delivery services (CDS) like UberEats and AmazonFlex highly depends on the decisions of individual shippers. Operating as freelancers, these shippers have the freedom to accept or decline orders from the CDS platform. Their decisions not only affect their earnings and the waiting times for orders but also influence the platforms’ overall revenue and reputation. Understanding the factors that shape these decisions is thus crucial. In our study, we gather data from CDS shippers in Shanghai, China, using stated preference surveys. We then design a discrete choice model to predict shippers’ behaviors and compare its accuracy, computational efficiency, and interpretability with five commonly used machine learning methods. Our analyses reveal that the Extreme Gradient Boosting (XGB) model and Random Forests (RFs) model outperform other models in prediction accuracy, achieving $f_1$ scores of 69.3% and 65% respectively. Notably, our performance improvements are significant, and the compensation awarded per order are the most influential determinants in their decision to accept or decline orders.

Index Terms—Crowd-sourced delivery, choice modeling, order acceptance behavior, machine learning, interpretability

I. INTRODUCTION

As e-commerce and online shopping continue to expand, there is a growing demand for cost-effective, environmentally sustainable, and efficient last-mile delivery solutions. An emerging trend in this domain is the adoption of crowd-sourced delivery services (CDS) [22]. In this model, individuals, whether they are regular shoppers or daily commuters, use their own vehicles to undertake delivery tasks. These ‘crowdshippers’ modify their usual routes, delivering products to e-shoppers and earning compensation for their service. This type of service provides a solution that not only lowers operational costs for retailers by reducing their dependence on full-time drivers, but also maximizes the efficiency of current transportation resources [23], [24]. This efficient utilization can help alleviate urban traffic and decrease greenhouse gas emissions. Recognizing its potential, many companies, ranging from retail giants such as Walmart1, Amazon2, and JD.com3, to startups like Piggybee4, as well as food delivery companies like UberEats5 and DoorDash6, are integrating CDS strategies into their logistical operations.

The operation of crowd-sourced delivery services requires three level of decisions: strategic, tactical, and operational. Strategic decisions outline the business model, choosing between intra-city and intercity delivery formats. These decisions also involve identifying target customers and leveraging potential transport resources, such as in-store patrons, daily commuters, and occasional travelers. Notably, studies by [14] and [4] have explored the factors influencing crowd-shipping service adoption. Additionally, [6] delves into Generation Z’s willingness to engage with these services. Tactical decisions, on the other hand, deal with medium-term management policies. These include decisions about which areas to service [26], [27], setting limits on detour times [28], [29], and shaping delivery pricing strategies [19] [31]. Operational decisions are immediate. They concern matching packages to crowd-sourced drivers and deciding their compensation [30], [33]. However, it’s worth noting that crowd-sourced delivery is inherently more unpredictable than traditional urban logistics, as highlighted by [25]. The motivations and criteria behind a crowd-sourced driver’s order acceptance are complex and not entirely understood. Gdowska et al. [32] suggest that drivers follow certain patterns in accepting orders, while Hou et al. [33] believe it based on the perceived benefits for the drivers. Despite these insights, a comprehensive understanding of these behaviors is still limited in this domain. To address this gap, our contributions are as follows:

- We design a set of stated preference survey questionnaires to obtain data on crowd-shipper acceptance choice;
- We design a binomial logit discrete choice model to reveal the main factors that influence crowd-

1https://drive4spark.walmart.com/  
2https://flex.amazon.ca/  
3https://www.jdl.com/  
4https://www.piggybee.com/  
5https://www.uber.com/deliver/  
6https://dasher.doordash.com
shipper acceptance choice and construct an intuitive choice model;

- We compare the performance of DCM and ML classifiers in terms of prediction performance and computational efficiency;
- We clarify the impact of different features on crowd-shipper acceptance choice and provide decision support for the operation of the crowd-sourced delivery platform through feature importance analysis.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III introduces the commonly used classifiers that we will compare in this paper. Section IV presents the questionnaire design and parameter estimation. The comparative study of the machine learning and discrete choice modeling approach is given in Section V. Finally, future research directions are discussed in Section VI.

II. RELATED WORK

In this section, we first review several survey papers and choices modeling papers on crowd-shipping problems, summarizing the problems they addressed and the primary behavioral modeling methods they employed. Subsequently, we review some related works in other fields that are similar to our work.

The majority of relevant studies in the field of crowd-shipping are focused on the demand side. Some papers analyze existing transaction data, such as Le et al. [13] examine shipping behaviors, potential crowd-shipping driver-partners, and stakeholder characteristics in the US, using revealed and stated preference surveys to inform logistics improvements, driver recruitment, and business strategies that align with the requester and driver expectations.

Furthermore, some researchers design their own stated preference survey questionnaires to obtain the information, as demonstrated by Le et al. [14], who investigate factors influencing senders’ choice of shipping services. The survey results show that shipping costs and real-time services such as courier reputation, tracking, and customization significantly impact decisions, with senders willing to pay more for crowd-shipping groceries. And this paper uses Random Utility Maximization (RUM) and Random Regret Minimization (RRM) to be the behavior modeling methods. Moreover, Galkin et al. [15] study Bratislava citizens’ attitudes toward working as occasional crowd-shipping couriers, finding that socio-demographic factors and fee value significantly influence participation. A regression relationship model between these factors and maximum parcel weight is built to model citizens’ behaviors. Al-Saudi at al. [11] investigates consumers’ attitudes toward crowd logistics in Qatar, finding that package insurance is the most valued attribute, followed by flexible delivery and transparent profiles. In [18], to understand factors influencing crowd-shipping adoption, a structural equation modeling method is used on a US survey. Survey results show that men, younger, full-time employed individuals in areas with high population and low employment density have more willingness to be crowd-shippers.

Regarding the supply side, some studies have also made their contributions. Specifically, Ermagun et al. [19] develop a binomial regression model to understand the bidding behavior of the supply side in crowd-shipping services. Thereafter, Ermagun et al. [12] use the random forest algorithm to predict crowd-shipping delivery performance across bidding, acceptance, and delivery stages in the US, finding that context, reward, and timing significantly impact the process, and demonstrating the potential to improve delivery probability through pricing and request timing adjustments. Regrettably, this paper lacks a comparative learning process and intuitively employs the Random Forest method as the approach to predict shipment status. Zehtabian et al. [17] model crowd-shipping pickup and delivery time estimation as a Markov decision process, proposing two look-ahead policies to improve accuracy.

To our knowledge, there is no literature specifically examining driver acceptance choice in crowd-shipping, but we found some similar studies in ride-sharing. Ashkrof et al. [20] use the focus group method to collect and analyze the decision behaviors of drivers in the ride-sharing market, which is similar to the crowd-shipping market. They also develop a conceptual model of tactical and operational decisions of ride-sourcing drivers. The model reveals that factors such as the rider’s pickup location, drop-off location, and the size of the luggage have a significant impact on whether a driver accepts a ride request. Two years later, Ashkrof et al. [21] use a set of stated preference survey questionnaires to explore the key factors that determine a driver’s acceptance behavior. This study employs a discrete choice modeling approach.

Our study also fills the gap in the literature on the choices modeling of crowd-shippers, which has not been explored systematically. While discrete choice models and machine learning methods have been widely applied to model and predict the behavior of customers or drivers in the transportation field, there is a lack of systematic comparison and analysis of these methods in the context of shared mobility, especially in the domain of crowd-shipping. Thus, unlike other data analysis and behavior modeling studies, our main contribution lies in collecting data on the crowd-shopper’s acceptance choices through a stated preference survey and comparing the accuracy, computational efficiency, and interpretability of discrete choice models and commonly used machine learning methods.

III. METHODS AND COMPARATIVE STUDY

In this section, we briefly introduce the Discrete choice model (DCM), and five commonly used ML
classifiers, such as random forests, K-nearest neighbor, extreme gradient boosting, support vector machine and artificial neural network.

A. Binomial logit model

The binomial logit discrete choice model, developed by [7], is one of the most commonly used discrete choice models to predict binary classification problems. The basic elements of the model are given as follows:

- Decision makers: the subject who chooses to act, in the crowd-sourced delivery is the self-employed courier.
- Alternative: there are usually multiple options for decision makers to choose from (e.g. couriers can choose orders among the given order menu, or choose to accept or reject the assigned orders)
- Attributes: the factors influencing the decision makers’ choice behaviors (e.g. the distance between couriers’ current locations and parcel pick-up locations)
- Decision rules: defaulted decision makers’ behavioral guidelines when making a choice. In this paper, we assume each courier is rational and obeys the utility-maximization principle.

The utility function, given by Eq. (1), determines whether a courier \( n \) accepts or rejects an assigned order \( j \).

\[
U_{nj} = \alpha + \beta X_{nj} + \epsilon_{nj} \tag{1}
\]

Where \( \alpha \) is an alternative specific constant (ASC), also can be regarded as an intercept for the \( j^{th} \) alternative order; meanwhile \( \beta \) is a vector capturing model parameters (coefficients), \( X_{nj} \) is a vector that captures all observable characteristics that influence the acceptance choices of decision makers. And let \( \epsilon_{nj} \) be the unobservable component for the specific decision maker and the respective alternative. As usual, the classic logit model assumes the stochastic term \( \epsilon_{nj} \) are independent and identically variable following type-I generalized extreme value distribution (Gumbel distribution). Therefore, the probability that a decision maker \( n \) choose to accept an assigned order \( j \) is defined as Eq. (2):

\[
P_n(j) = \frac{1}{1 + e^{-(\alpha + \beta X_{nj})}} \tag{2}
\]

Since we assume the error term \( \epsilon_{nj} \) follows Gumbel distribution, and the coefficient vector \( \beta \) is given, the likelihood function can be defined as Eq. (3)

\[
LL = \prod_{n=1}^{N} \left[ P_n(j)^{\lambda_{nj}} \cdot (1 - P_n(j))^{1-\lambda_{nj}} \right] \tag{3}
\]

where \( \lambda_{nj} \) is equal to 1 if the individual \( n \) choose to accept the assigned order \( j \). Commonly, the natural logarithm of Eq.(3) is used, Eq.(3) uses the logarithm function to simplify the math and computations.

\[
lnLL = \sum_{n=1}^{N} \left[ \lambda_{nj} ln(P_n(j)) + (1 - \lambda_{nj}) ln(1 - P_n(j)) \right] \tag{4}
\]

Then, we use the maximum likelihood method to compute the vector \( \beta = \arg\max_{\beta} lnLL \) that maximizes the joint-density of the samples. Finally, upon acquiring the estimated values of \( \beta \), by integrating these coefficients into the preceding formula Eq. (2) and given the observable feature values, we can prognosticate the probability of each individual accepting a specific order.

B. Decision Trees and Random forests

The decision tree (DT) is a classical classifier and regressor frequently employed in the realm of data science, representing a relatively commonly used supervised machine learning methodology. By assimilating simple rules from feature data within the decision-making process, it build a tree structure to forecast the values of target variables. Its main advantages are:

- Interpretability: Since the tree structure can be visualized and closely resembles the human decision-making process, it can be easily understood by non-experts after a simple description.
- Requires little data preparation: Owing to the capability of the Decision Tree (DT) methodology to handle qualitative data without necessitating the implementation of dummy encoding, it stands as a more efficient predictive approach in comparison to other classification algorithms.
- In built feature selection. The hierarchical structure intrinsic to the Decision Tree (DT) allows for an intuitive reflection of feature significance. Features located in the upper echelons of the tree generally exert a more pronounced impact on predictive outcomes due to their heightened information entropy compared to those situated at lower levels. Consequently, features at these lower strata can be judiciously retained or omitted based on specific requirements.

However, there are many limitations of DTs.

- Unstable: changes to the training set may have a significant impact on the structure of the tree
- NP-complete: each node is generated based on some heuristic algorithms such as greedy algorithms, which can cause the prediction results to not achieve the global optimum
- Overfitting: deeper and more complex tree structures may cause overfitting

The common algorithms for solving decision trees are ID3, C4.5, and CART. The main solution idea is to calculate the information entropy (information gain rate) and Gini impurity.
Random forests (RFs) [8] are an ensemble learning method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. RF is an extended variant of Bagging [16], which further introduces random feature selection in the training process of decision trees based on the decision tree as the base learner to build Bagging integration, so it can be summarized that RF consists of four parts: random sample selection (put-back sampling); random feature selection; decision tree construction; and random forest voting (averaging).

Random Forest (RF) [8] is an ensemble learning technique, applicable to tasks such as classification, regression, among others. Its predictive output is predicted upon the amalgamation of multitude of decision trees. Specifically, for classification tasks like forecasting the probability of a driver accepting an order, the output of the RF is determined by the class favored by the majority of trees. RF can be perceived as an augmented variant of the Bagging algorithm [16]. It integrates an element of randomness into feature selection during the decision tree’s training phase. Built upon decision trees as the foundational learners, RF fabricates a Bagging ensemble. Summarily, RF can be summarized into four principal parts: random sample selection (with replacement); stochastic feature selection; construction of decision trees; and the Random Forest’s voting (averaging).

The advantages of RFs are: It can come out with very high dimensional (many features) data and without dimensionality reduction, no need to do feature selection it can determine the importance of features can determine the interaction between different features is not easy to overfit training is faster and easy to make parallel method is relatively simple to implement for unbalanced data sets, it can balance the error. If a large portion of the features is missing, the accuracy can still be maintained. Signal data without having to do the feature selection.

The merits of RFs include the following: RFs can handle data with a very high dimensionality (numerous features) without the necessity for dimensionality reduction. There’s no mandatory feature selection; RFs can inherently discern feature importance. They possess the ability to determine interactions between different features. RFs are less susceptible to overfitting. They exhibit faster training times. They can be readily parallelized, facilitating swift computations. Implementation-wise, they are comparatively straightforward. For imbalanced datasets, RFs can equilibrate the error. Even when a substantial portion of features is missing, they can still maintain a commendable level of accuracy. They are adept at handling raw signal data without necessitating feature selection.

While RFs typically achieve higher accuracy than individual decision trees, they sacrifice the inherent interpretability found within a single decision tree. It has been demonstrated that RFs can overfit in certain noisy classification or regression scenarios. For datasets with varying attribute values, attributes with more split points exert a greater influence on the Random Forest. Consequently, in terms of attribute weightings, the outputs produced by RFs for such datasets are deemed unreliable.

C. K-Nearest Neighbor

The k-Nearest Neighbors (KNN) algorithm is a non-parametric unsupervised learning technique developed by [2], suitable for both classification and regression tasks. In both instances, the input consists of the k closest training samples from the dataset. The output for KNN classification is a class membership. Objects are classified based on a majority vote of their neighbors, with an object being assigned to the most common class amongst its k nearest neighbors (where k is a positive integer, typically small). Neighborhoods are commonly defined using the Euclidean distance metric. A conspicuous advantage of the KNN algorithm is its simplicity. However, computational costs can escalate with high-dimensional data. Moreover, proximate samples might not always pertain to the same category.

D. Extreme Gradient boosting

Extreme Gradient Boosting (XGB) is a powerful and widely-used machine learning algorithm for regression and classification problems. It was developed by [9] and is now maintained by a team of developers at DMLC. XGB is a gradient-boosting algorithm that combines the strengths of tree-based algorithms and gradient boosting to produce high-quality, accurate models.

Gradient boosting is a technique that involves iteratively improving a weak model by fitting it to the residual errors of the previous iteration. This process continues until the model achieves its optimal performance. Extends this approach by adding additional regularization and parallel processing capabilities, which allows it to achieve high accuracy and performance even with large datasets.

One of the key features of is its ability to handle missing data. It uses a technique called gradient-based imputation to fill in missing values, which involves fitting a model to the observed data and then using that model to predict the missing values. This approach can produce more accurate imputations than traditional imputation methods.

Another important feature of is its ability to handle both numerical and categorical data. It can convert categorical variables into numerical variables using techniques such as one-hot encoding, which creates a separate binary variable for each possible value of the categorical variable. This allows the algorithm
to accurately capture the relationships between the categorical variables and the target variable.

also includes a number of regularization techniques to prevent overfitting and improve the generalization performance of the model. These include L1 and L2 regularization, which add penalty terms to the objective function to discourage large coefficients, and tree pruning, which removes branches of the tree that do not contribute to the model’s overall accuracy.

One of the key advantages of is its speed and scalability. It is designed to be highly optimized and can be run in parallel on multi-core CPUs or GPUs. It also includes a number of optimizations, such as approximate greedy algorithm and caching, that can speed up the training process.

XGB has become a popular algorithm for a wide range of applications, including image and speech recognition, natural language processing, and financial modeling. Its versatility and speed make it a valuable tool for data scientists and machine learning practitioners.

E. Support vector machine

Support vector machines (SVMs, also known as support vector networks) are one of the most robust supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at ATT Bell Laboratories by [10]. Given training samples, categorized to be two classes, an SVM training algorithm builds a model that assigns new instances to one class or the other, making it a non-probabilistic binary linear classifier. SVM maps training examples to points in space so as to maximize the width of the gap between the two classes. New examples are then mapped into that same space and predicted to belong to a class based on which side of the gap they fall. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Support Vector Machines (SVM), also referred to as Support Vector Networks, rank among the most potent supervised learning models equipped with associated learning algorithms for data analysis, classification, and regression. Developed at ATT Bell Laboratories by [10]. SVM operates on a foundational premise: for a binary classification problem with two discernible classes, the SVM training algorithm builds a model that assigns new instances to one class or the other, rendering it a non-probabilistic binary linear classifier. Conceptually, SVM maps training instances to points in a space, maximizing the margin or gap between the two classes. Subsequent instances are then projected into this same space and classified based on which side of the gap they fall on. Beyond the realm of linear classification, SVMs can transcend to non-linear classification efficaciously using the so-called kernel trick. This technique allows the SVM to operate in a high-dimensional feature space by implicitly mapping their inputs into this space, without the explicit transformation.

The advantage of support vector machines is that:

- Since SVM is a convex optimization problem, the solution obtained must be globally optimal rather than locally optimal.
- It is not only applicable to linear problems but also to nonlinear problems (using kernel tricks).
- Data characterized by a high-dimensional sample space can be effectively utilized in conjunction with SVMs. This is primarily because the complexity of the dataset, in the context of SVMs, is contingent upon the support vectors rather than the dimensionality of the data itself. This attribute intrinsically circumvents the "curse of dimensionality" to a certain extent.
- The theoretical basis is better (e.g., neural networks are more like a black box).

Nevertheless, the disadvantage of support vector machines is that the solution of a quadratic programming problem will involve the computation of a matrix of order $m$ ($m$ being the number of samples), so SVMs are not suitable for very large data sets. It is only applicable to binary classification problems.

F. Artificial neural network

Artificial Neural Networks (ANNs), commonly referred to as neural networks, are inspired by biological neural systems. Emulating the functioning processes of the brain, ANNs are adept at executing a diverse array of tasks such as classification and regression. Through continuous transformations of the feature space, ANNs endeavor to smooth non-linearities, aiming to identify an informative foundation upon which to approximate a fundamental model [3]. A salient characteristic of ANNs is their ability to autonomously learn feature representations, adjusting to the classification model.

A common multilayer structured feed-forward network (Multilayer Feedforward Network [11]) consists of three components:

- Input layer: where neurons receive a large number of non-linear input data
- Output layer: where data are transmitted, analyzed and weighed in the neuron links to form the output result.
- Hidden layer: is the layer of neurons and links between the input and output layers. The hidden layer can have one or more layers. The number of nodes (neurons) in the hidden layer is variable, but the greater the number, the more significant the nonlinearity of the neural network, and thus the more significant the robustness of the neural network.
IV. QUESTIONNAIRE DESIGN AND DATA ANALYSIS

In this section, we present the process of designing our questionnaires and provide descriptive statistics of the collected data.

A. Questionnaire design

Our questionnaire is divided into two parts. The first part involves providing basic information, including age, gender, and income, which are the three fundamental personal details. Additionally, respondents are asked to indicate their primary mode of transportation, with four basic options: car, public transit, bicycle, and walking. The second part contains order information. Before designing this section of the questionnaire, we considered the following aspects:

- **Detour time**: Longer detour time greatly reduces crowd-shippers’ willingness to deliver orders.
- **Payment**: Intuitively assessed, the payment offered emerges as one of the most influential factors impacting the acceptance behavior of crowd-shippers.
- **Weather and season condition**: Variations in weather conditions undeniably influence human mobility choices. Grounded in this consideration, we posit that the delivery willingness of crowd-shippers is similarly susceptible to the changes of weather and season conditions.
- **Parcel size**: Taking into account that larger packages may not be deliverable by crowd-shippers using modes of transportation other than cars and that females might be reluctant to carry overly heavy packages.

Some parts of attributes and corresponding descriptions are summarized in Table I. The generation of weather, season, and package size data follows a discrete uniform distribution.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weather</td>
<td>Sunny, rainy (2 levels)</td>
</tr>
<tr>
<td>Season</td>
<td>Summer, winter (2 levels)</td>
</tr>
<tr>
<td>Parcel size</td>
<td>Small, medium, large (3 levels)</td>
</tr>
</tbody>
</table>

We utilize the “geopy” Python library to randomly select four locations on the map: the origin (O), destination (A), order pickup location (B), and order drop-off location (D), all within Shanghai and its surrounding areas. Subsequently, we obtained the routes and durations between the four points by calling the Google Maps API. The travel time of the original route is denoted by OD, and the travel time of the new route is defined as the route taken by the crowd-shipper from point O to completing the order (AB) and reaching the final destination (D), the detour distance is defined as

\[ D' = D_{OA} + D_{AB} + D_{BD} - D_{OD} \]

Additionally, for the payment amount, we employ the prevalent detour time-tiered compensation mechanisms currently in use, refer to Eq. 5.

\[
\text{payment} = \begin{cases} 
5 + D' & \text{if } D' \leq 5 \\
10 + 1.5 \times D' & \text{otherwise}
\end{cases}
\]  

(5)

Our survey was disseminated through various channels, encompassing social media platforms such as TikTok, Instagram, and Weibo, as well as communication applications including WeChat, WhatsApp, and QQ. Each respondent was provided access to a unique, individualized survey link. A sample of the questions posed to each participant is illustrated in Fig. 1, and provide us with their personal information mentioned above.

![Sample questionnaire](image)

**Fig. 1: Sample questionnaire**

B. Survey data description and parameter estimation

After discarding invalid or incomplete surveys, we obtained 308 valid questionnaires, including 3080 order samples from Shanghai, China. Table II summarizes the descriptive statistics of collected data in China between December 2022 and April 2023. In the survey, 1723 orders (55.94%) are rejected by respondents, and 1357 orders (44.06%) are accepted.

Then we use the maximum log-likelihood estimation method to estimate the parameters of the DCM. The parameter estimation is executed on a computer with an Intel Core i7 6-core CPU with 16 GB of RAM, running at 2.6 GHz, using Mac OS X version 19.16.1. The model is implemented in Python version 3.8.5, using “statsmodels” module.

Table III summarize the estimated parameter values, the corresponding standard errors, Z-values, and p-values. Also it shows that, \( t_{BD}, t_{OD}, \) income, and payment have a relatively profound impact on crowd-shippers’ acceptance choices. The probability that a crowd-shippers accepts a delivery request decreases as the growth of value of BD (\( \beta_{BD} = -0.01, p < 0.01 \)),
income \( (\beta_{\text{income}} = -0.04, p < 0.01) \), mode_{bus} \( (\beta_{\text{mode-bus}} = -1.14, p < 0.015) \) and payment \( (\beta_{\text{payment}} = -0.11, p < 0.01) \), while the probability decreases as the \( t_{OD} \) \( (\beta_{t_{OD}} = 0.01, p < 0.01) \), and age \( (\beta_{age} = 0.01, p < 0.015) \). The value of the ASC (intercept) is 1.79, and \( p < 0.01 \). Factors with a p-value greater than 0.01 are considered to have an insufficiently significant influence on the decision-making value greater than 0.01 are considered to have an insufficiently significant influence on the decision-making.

### TABLE III: Parameter estimation results

|         | \( \text{coef} \) | \( \text{std err} \) | \( z \) | \( P > |z| \) |
|---------|-------------------|---------------------|-------|------------------|
| \( \text{intercept} \) | 1.79              | 0.338               | 5.319 | 0.000            |
| \( AB \) | 0.00              | 0.00                | 0.83  | 0.41             |
| \( BD \) | -0.01             | 0.00                | -3.67 | 0.00             |
| \( OA \) | 0.01              | 0.00                | 1.25  | 0.21             |
| \( OD \) | 0.01              | 0.00                | 3.31  | 0.00             |
| age     | 0.01              | 0.01                | 2.58  | 0.01             |
| gender_m | 0.06              | 0.10                | 0.53  | 0.60             |
| income  | -0.04             | 0.01                | -5.60 | 0.00             |
| mode_{bus} | -0.40            | 0.15                | -2.67 | 0.01             |
| mode_{car} | -0.05            | 0.11                | -0.43 | 0.67             |
| mode_{walk} | -1.14            | 0.53                | -2.14 | 0.03             |
| parcel_{size-medium} | -0.01           | 0.12                | -0.10 | 0.92             |
| parcel_{size-small} | 0.14             | 0.12                | 1.13  | 0.26             |
| payment  | -0.11             | 0.01                | -1.22 | 0.22             |
| season_{winter} | -0.12            | 0.10                | -1.28 | 0.20             |
| weather_{sunny} | -0.06            | 0.09                | -0.65 | 0.51             |

### TABLE II: Descriptive statistics of personal information

<table>
<thead>
<tr>
<th></th>
<th>( \text{mean} )</th>
<th>( \text{std} )</th>
<th>( \text{min} )</th>
<th>( \text{25%} )</th>
<th>( \text{50%} )</th>
<th>( \text{75%} )</th>
<th>( \text{max} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>39.20</td>
<td>11.14</td>
<td>24.00</td>
<td>29.00</td>
<td>39.00</td>
<td>47.00</td>
<td>58.00</td>
</tr>
<tr>
<td>income</td>
<td>11.44</td>
<td>11.77</td>
<td>11.14</td>
<td>11.77</td>
<td>11.44</td>
<td>11.77</td>
<td>11.77</td>
</tr>
<tr>
<td>( mode_{bus} )</td>
<td>14.65</td>
<td>12.85</td>
<td>12.85</td>
<td>14.65</td>
<td>12.85</td>
<td>14.65</td>
<td>14.65</td>
</tr>
<tr>
<td>( BD )</td>
<td>15.02</td>
<td>13.60</td>
<td>13.60</td>
<td>15.02</td>
<td>13.60</td>
<td>15.02</td>
<td>15.02</td>
</tr>
<tr>
<td>( OD )</td>
<td>10.96</td>
<td>6.48</td>
<td>6.48</td>
<td>10.96</td>
<td>6.48</td>
<td>10.96</td>
<td>10.96</td>
</tr>
<tr>
<td>payment</td>
<td>17.95</td>
<td>308.00</td>
<td>308.00</td>
<td>308.00</td>
<td>308.00</td>
<td>308.00</td>
<td>308.00</td>
</tr>
</tbody>
</table>

### V. COMPARATIVE STUDY

In this section, we primarily compare the differences in predictive performance, efficiency, and interpretability between various machine learning (ML) methods and discrete choice model (DCM) approaches. Additionally, the tuning of different ML hyper-parameters and the interpretation of the results are also presented.

#### A. Hyper-parameter tuning

Before entering the training stage, it is essential to identify a collection of hyper-parameter values that yield optimal performance for each model on the given data within a reasonable time frame. This procedure is referred to as hyper-parameter optimization or tuning and significantly impacts the predictive accuracy of machine learning algorithms [5]. For the sampling strategy, the Repeated Stratified K-fold cross-validation method is used to effectively manage the imbalance in the number of individuals selecting each mode. Additionally, statistical inference tools are utilized to assess the differences in prediction accuracy between the methods. Validation and estimation of machine learning models are carried out using Python’s Scikit-Learn library, while hyper-parameter selection for each machine learning classifier is achieved via Python’s Scikit-Learn library RandomizedSearchCV package. To guarantee the reproducibility of the results, a random seed is established. The hyper-parameter selection results are presented as follows, refer to Table. IV.

#### B. Classification performance

We employed a widely acclaimed evaluation tool designed for assessing binary classification prediction methods, known as the Receiver Operating Characteristic (ROC) curve, to gauge the predictive performance and capability of the various behavioral models discussed in this study in forecasting the order acceptance choices of crowd-shippers. The ROC curve’s x-axis represents the False Positive Rate (FPR), which, in the context of this research, denotes the proportion of delivery requests that were wrongly predicted as “accepted” among all “rejected” requests. Conversely, the y-axis signifies the True Positive Rate (TPR), which is the fraction of delivery requests accurately predicted as “accepted” amidst all “accepted” requests. Thus, a lower FPR coupled with a higher TPR suggests commendable predictive prowess. The ROC curve of our model, as illustrated in Figure 3, lies proximate to the upper-left corner, underscoring the model’s robust predictive capacity.

Furthermore, to prevent overfitting, we randomly split 80% of the survey results as training data and the rest of the observations are regarded as testing data. The training data is utilized to estimate parameters, while the testing data serves to evaluate the model’s predictive capacity. To enhance the reliability of the model assessment, we repeated this procedure 200 times and discovered that the results are robust. The F1-score comparative results are given in Table. V

Based on our observations, we find that the (XGB) algorithm outperforms other algorithms in terms of both the ROC curve and F1-score performance. Moreover, in terms of runtime, as shown in Table.VI, XGB also exhibits a prominent performance, only surpassed by the simpler K-Nearest Neighbors (KNN) and Binomial Logit Regression models.
C. Feature importance

Taking into account the interpretability of different classification methods and the desire to ascertain the impact of specific features on crowd-shipper order acceptance choice, we employ the Permutation Importance method from Python’s Scikit-Learn library which is a technique used to determine the feature importance in a machine learning model by evaluating the change in model performance when the values of a particular feature are randomly shuffled. Through this approach,
TABLE V: F1 score comparison of different classifiers

<table>
<thead>
<tr>
<th>Classifier</th>
<th>mean</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>0.65</td>
<td>0.018</td>
</tr>
<tr>
<td>ANN</td>
<td>0.621</td>
<td>0.026</td>
</tr>
<tr>
<td>KNN</td>
<td>0.626</td>
<td>0.016</td>
</tr>
<tr>
<td>XGB</td>
<td>0.693</td>
<td>0.015</td>
</tr>
<tr>
<td>SVM</td>
<td>0.628</td>
<td>0.015</td>
</tr>
<tr>
<td>DCM</td>
<td>0.608</td>
<td>0.019</td>
</tr>
</tbody>
</table>

TABLE VI: Running time comparison of different classifiers

<table>
<thead>
<tr>
<th>Classifier</th>
<th>mean</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>0.265</td>
<td>0.008</td>
</tr>
<tr>
<td>ANN</td>
<td>1.069</td>
<td>0.042</td>
</tr>
<tr>
<td>KNN</td>
<td>0.057</td>
<td>0.005</td>
</tr>
<tr>
<td>XGB</td>
<td>0.139</td>
<td>0.086</td>
</tr>
<tr>
<td>SVM</td>
<td>0.157</td>
<td>0.1</td>
</tr>
<tr>
<td>DCM</td>
<td>0.062</td>
<td>0.014</td>
</tr>
</tbody>
</table>

we calculate the influence of various features on the outcomes in different ML methods. Please refer to Fig. 2 for details.

Through the feature importance analysis, we can observe that age, income, and payment are the three most influential factors affecting drivers’ order acceptance choice. All six different classification methods demonstrate relatively high importance scores for these features. Furthermore, we find that the interpretability of the prediction results from K-Nearest Neighbors (KNN) and Random Forest (RF) classifiers are relatively poor. The primary reason for this is that the KNN algorithm underperforms when dealing with features exhibiting high correlations. In our data generation process, the payment value is influenced by the distances between the four locations, resulting in high feature correlations. As for the RF method, besides sharing the same issue as the KNN algorithm, it is better suited for large-scale data. Our survey data has a relatively small scale, which may not be sufficient to identify the differences in feature importance.

VI. CONCLUSION AND FUTURE WORK

This paper designs a stated preference survey to collect driver acceptance choice data in the CDS system. By using the binomial logit model, we reveal the main factors that influence the driver’s acceptance choice and estimate a driver acceptance utility function. Furthermore, we compare the logit model with five other commonly used machine learning methods and found that the XGB method outperforms the others on the collected data set in terms of prediction accuracy, computation efficiency, and interpretability. Our computational results reveal that age, income, and payment price per order are the main factors that influence drivers’ decision-making. In addition, the driver’s original route and the delivery route of the order also have a certain impact on the driver’s acceptance choice. Moreover, the results of our experiments can provide decision support for the design of CDS order assignment systems in the industry. For the academic community, we quantified the uncertainty of drivers’ decision-making choice and found suitable behavioral modeling methods based on real data, laying the foundation for future research in this area.

Future work includes continuing our survey, analyzing the impact of additional compensation on drivers’ secondary decision-making, introducing time variables to explore decision-making patterns under dynamic mitigation, and integrating this behavioral model with optimization problems.

ACKNOWLEDGEMENT

This study, related to human behaviors data, is approved by Concordia University, under Certification Number: 30018372.

REFERENCES