

A COMPUTER-IMPLEMENTED METHOD, AND A SERVER

Technical Field

The invention relates generally to the field of logistics. One aspect of the invention relates to a computer-implemented method. Another aspect of the invention relates to a server.

Background

In terms of logistics, it may be useful to be able to predict a delivery time. This may then be used to inform the customer when to expect the delivery. The more accurate the estimate is, the happier the customer may be. Further it may also be useful to accurately inform the merchant when the driver will arrive so they can optimise the order preparation time. The driver in turn, will be usefully informed when the order is ready for pickup.

However, it is sometimes problematic when some merchants do not begin order preparation until the driver arrives. This can cause a flow on problem for later estimation, when the data from such non standard merchants distorts historical data.

Summary

Embodiments may be implemented as set out in any of the independent claims. Some optional features are defined in the dependent claims.

According to a further embodiment there may be provided a method of supervised training of a ML model to determine food preparation time for standard merchants using historical transaction data from standard merchants.

According to a further embodiment there may be provided a method of estimating food preparation time for standard merchants using a trained ML model, based on at

least one parameter regarding the food order, wherein the trained ML model has been trained according to the method of the preceding paragraph.

Implementation of the techniques disclosed herein may provide significant technical advantages. Advantages of one or more aspects may include one or more of the following:

- The technical solution of faster processing of orders based on the technical problem of inaccurate training data.
- The technical solution of more accurate estimate of order processing time and/or estimated time of arrival based on the technical problem of inaccurate training data.
- The technical solution of reduction in the amount of communication traffic due to less cancelled orders based on the technical problem of inaccurate training data.
- The technical solution of better predictions for order ready time for non-Merchant start Preparing Before Driver Arrives (MPBDA) merchants by using nearby merchants' data instead of relying on merchant's historical data, which may not reflect the real food preparation time based on the technical problem of inaccurate training data.
- The technical solution of improved accuracy of estimated time of arrival (ETA) predictions by predicting the food preparation time component for non-MPBDA merchants before the order is placed based on the technical problem of inaccurate training data.
- The technical solution of improved delivery experience by allowing drivers to arrive at the restaurant at the right time, avoiding waiting time for both the food and the driver based on the technical problem of inaccurate training data.
- The technical solution of improved data quality and accuracy, based on the technical problem of misuse of the Merchant Order Ready (MOR) button by merchants.

- The technical solution of reduced greenhouse emissions based on the technical problem of inaccurate training data. Because the orders are limited to those with accurate order processing time and/or estimated time, there may be less wasted trips by drivers, or less wasted sales by merchants, and therefore greenhouse emissions for any unnecessary trips or unnecessary product manufacturing may be avoided.
- The technical solution of reduced data centre energy requirements based on the technical problem of inaccurate training data. Because the training is limited to accurate training data, therefore less energy is required for powering the servers and/or for cooling the servers to do the training. This further results in the technical solution of reduced greenhouse emissions due to the lower server energy requirements.
- The technical solution of reduced server hardware required for the technical problem of inaccurate training data. Because the training is limited to accurate training data, so less hardware is required. Reduced greenhouse emissions result from less manufacturing of hardware.
- The technical solution of reduced bandwidth requirements based on the technical problem of inaccurate training data unregistered users booking transport. Because the training is limited to accurate training data, so less bandwidth is required for communications. Reduced greenhouse emissions result from less communications bandwidth requirements.
- The technical solution of less latency for the technical problem of estimating order processing time and/or estimated time of arrival. Because the data set for estimation is limited to MPBDA merchants, the estimation is quicker.

In an exemplary implementation, the functionality of the techniques disclosed herein may be implemented in software running on a server communication apparatus (such as a one or more servers, one or more virtual machines, one or more processors or a cloud computing platform), which communicates with the applications running on the user terminals, driver terminals and/or merchant

terminals such as mobile phones. The software which implements the functionality of the techniques disclosed herein may be contained in a computer program, or computer program product. The server communication apparatus establishes secure communication channels with the driver, merchant and/or user terminals, for estimating the order preparation and/or delivery time.

Brief Description of the Drawings

The invention will now be described, by way of example only, and with reference to the accompanying drawings in which:

Figure 1 is a schematic block diagram illustrating an exemplary delivery/transportation service.

Figure 2 is a schematic block diagram illustrating an exemplary communications server for the delivery/transportation service.

Figure 3 is a flow diagram of a typical order delivery journey.

Figure 4 is a flow diagram of a machine learning training process.

Figure 5 is a visualisation of different dish vectors.

Figure 6 is a schematic diagram of a machine learning training process.

Detailed Description

The techniques described herein are described primarily with reference to use in estimating order processing time and/or time of arrival for a platform provider. This may be useful to identify merchants with non standard behaviour, which would otherwise corrupt training data. The data from non standard merchants can be excluded from the training data, and the model used to estimate or predict timings for non standard merchants can be modified to be more accurate for that subset of merchants.

Figure 1 shows an exemplary architecture of a system 100, with a number of users each having a communications device 104, a number of merchants each having a communications device 109, a number of drivers each having a user interface

communications device 106, a server 102 (or geographically distributed servers) of a platform provider and communication network 108 connecting each of the components. Each user contacts the server 102 using a user software application (app) on the communications device 104. Similarly, the drivers and merchants may use an app on their devices 106, 109.

A platform provider may use the system 100 for the purpose of logistics management. This may involve deliveries or e-commerce-based transactions, where a customer orders some food, a product, or other physical item needing delivery. Typically, the system 100 used by the platform operator will facilitate transactions between the users, and goods or services providers such as drivers or merchants. The system 100 may then be used to optimise the allocation of available drivers for delivery of the orders.

For deliveries or e-commerce-based transactions, the user device 104 may allow the users to input queries containing the keywords for the items of interest and delivery addresses. The user may see a list of merchants and/or items provided by the merchants, and order items from the merchants. The merchant may contact server 102 using merchant device 109 for providing information about their items and receiving orders for each confirmed transaction. The driver contacts server 102 using driver device 106. The driver device 106 allows the drivers to indicate their availability to take the delivery jobs, information about their vehicle, their location. Server 102 may then match drivers to the delivery, based on, for example: geographic location of merchants and drivers, maximising revenue, user or driver feedback ratings, weather, driving conditions, traffic level / accidents, relative demand, environmental impact, and/or supply levels. The user may be offered a particular delivery cost and approximate delivery ETA. If the user accepts the offer, the system may go through a payment authorisation process. If the authorisation is approved, the merchant will then be notified and directed to provide goods for the driver to pick up. The selected driver will then be notified and directed to the pickup

location to pick up the goods. The driver may be optimised to pick up goods from several merchants and drop off the series of orders in a single efficient path. During the delivery, the user device 104, the driver's device 106, the merchant's device 109 and the server 102 may be updated with real-time trip information including the real-time location of the driver's vehicle, the destination, the driver fare and/or other trip-related information. After the trip, the driver's device 106 may send a confirmation that the trip has ended to server 102. Once the transaction is approved and/or the delivery is completed, the user device 104, the driver's device 106, the merchant's device 109 and the server 102 may be updated with details of the completed financial transaction. This allows an efficient allocation of resources because the available fleet of drivers is optimised for the users' demand in each geographic zone.

Referring to Figure 2, further details of the components in the system of Figure 2 are now described. The system 100 comprises the communication server 102, and it may include the user communication device 104, the merchant communication device 109 and the driver communication device 106. These devices are connected in the communication network 108 (for example, the Internet) through respective communication links 110, 111, 112, and 114 implementing, for example, internet communication protocols. The communication devices 104, 106 and 109 may be able to communicate through communication networks and/or protocols, including cellular communication networks, LAN, WAN, private data networks, VPN, fibre optic connections, laser communication, microwave communication, satellite communication, Bluetooth, Wifi, NFC, etc., but these are not specified in Figure 2 for the sake of clarity.

In the example shown in Figure 2, the communication server apparatus 102 may comprise several individual components including, but not limited to, one or more microprocessors 116, one or more memories 118 (e.g., a volatile memory such as a RAM), and/or longer-term, non-volatile, or persistent, memory 119 (e.g., SSD (Solid

State Drive) or Hard disk drives (HDD)) for the loading of executable instructions 120, the executable instructions defining the functionality the server apparatus 102 carries out under the control of the microprocessor 116. The communication server apparatus 102 also comprises one or more input/output modules 122 allowing the server to communicate over the communication network 108. One or more user interfaces 124 are provided for administrator control and may comprise, for example, computing peripheral devices such as display monitors, computer keyboards and the like.

The communication server apparatus 102 may be a single server as illustrated schematically in Figure 2. Alternatively, the functionality performed by the server apparatus 102 may be distributed across multiple physically or logically separate server components. In the context of the present specification, "a server", "the server" or "said server" is a computer program that is running on appropriate hardware and is capable of receiving requests (e.g., from electronic devices) over a network, and carrying out those requests, or causing those requests to be carried out. The hardware may be implemented as one or more physical computers, one physical computer system, one or more virtual machines, or a cloud-based server network. In the present context, the use of the expression a "server" is not intended to mean that every task (e.g. received instructions or requests) or any particular task will have been received, carried out, or caused to be carried out, by the same server (i.e. the same software and/or hardware); it is intended to mean that any number of software elements or hardware devices may be involved in receiving/sending, carrying out or causing to be carried out any task or request, or the consequences of any task or request; and all of this software and hardware may be one server or multiple servers. The same approach is to be taken for references to other systems, software or hardware components e.g., processor, memory, storage etc.

Here the transactions referred to may be payments done for anything the platform provider offers, for e.g., Ride-hailing, Ride-sharing, food delivery, e-commerce

deliveries, etc. Transactions can include any interaction where a user, driver or merchant has to pay to the platform provider for its services or products, accept orders, or otherwise communicate with the platform provider.

The server apparatus 102 may also comprise one or more databases 126 stored in volatile memory 118 and/or non-volatile memory 119, for storing data, which may include data on merchant behaviour, geographic information, images, products, points of interest, users, drivers, merchants, transaction data, aggregated data or parameters, payment data, and other relevant data. The data may be stored in a data structure according to the requirements of the application, or as described in more detail below. The database 126 may be replicated, distributed, sharded or otherwise optimised according to the requirements of the application.

The user communication device 104 may comprise several individual components including, but not limited to, one or more microprocessors 128, a memory 130 (e.g., a volatile memory such as RAM), and/or longer-term memory such as flash memory or SSD (Solid State drives) for the loading of executable instructions 132, the executable instructions defining the functionality the user communication device 104 carries out under the control of the microprocessor 128. The user communication device 104 also comprises an input/output module 134 allowing the user communication device 104 to communicate over the communication network 108. A user interface 136 is provided for user control. If the user communication device 104 is, say, a smartphone or tablet device, the user interface 136 will have a touch panel display as is prevalent in many smartphones and other handheld devices. Alternatively, if the user communication device 104 is, say, a desktop or laptop computer, the user interface 136 may have, for example, computing peripheral devices such as display monitors, computer keyboards and the like.

The merchant communication device 109 may be, for example, a smartphone or tablet device with the same or a similar hardware architecture to that of the user communication device 104.

The driver communication device 106 may be, for example, a smartphone or tablet device with the same or a similar hardware architecture to that of the user communication device 104. Alternatively, the functionality may be integrated into a bespoke device such as a taxi fleet management terminal.

The driver mentioned above may also be implemented by a remote driver or an autonomous driver. Examples include autonomously driven vehicle, UAVs, and drones. In that case the merchant may be responsible for loading the order with the delivery vehicle, or that task may be automated as well.

Thus, it will be appreciated that Figures 1 and 2 and the foregoing description illustrate and describe a system 100 comprising:

- a communication server 102;

- at least one merchant communication device 109 having an associated merchant and configured to accept a potential order and/or confirm an order is ready for pickup; and

- communication network equipment 108 configured to establish communication with the communications server 102, and the at least one user communication device 104;

- wherein the communications server 102 comprises at least one processor, and at least one memory 118, the server being configured, under control of one or more of the at least one processor 116, to execute instructions stored in one or more of the at least one memory 118 to:

- store data relating to historical merchant behaviour including aggregated data on when potential orders accepted and/or when orders

are ready for pickup from the merchant communication device and/or the driver communication device;

determine which of the merchants exhibit standard behaviour and which exhibit non standard behaviour based on the stored historical merchant behaviour, wherein non standard behaviour includes a deviation above a threshold time between when the orders are accepted and ready for pickup, which indicates preparation delays; and

store a list of standard behaviour merchants and non standard behaviour merchants based on the determination.

Further, it will be appreciated that Figures 4 and 6 illustrates and describes a method performed in a communication server apparatus 102, the method comprising, under control of a microprocessor 116 of the communication server apparatus 102:

storing data relating to historical merchant behaviour;

determining which of the merchants exhibit standard behaviour and which exhibit non standard behaviour based on the stored historical merchant behaviour, wherein non standard behaviour includes behaviour which delays delivery; and

storing a list of standard behaviour merchants and non standard behaviour merchants based on the determination.

Order Preparation Time

An order delivery journey 300 typically consists of several components demarcated by actions of a driver and a merchant as shown in Figure 3.

First an order is created 302 by a user 104. The order is promulgated to applicable merchants, and once a merchant accepts 304, a driver is allocated 306. The delay between the order creation and driver allocation is called Allocation Buffer Time (ABT) 308.

Once the merchant accepts the order, they should immediately begin preparing the order. Once the merchant order is ready 309, it then waits for the driver. The food preparation time (FPT) 310 is defined as the time required to prepare the food, i.e., time duration between the merchant accepting the order 304 timestamp and the MOR 309 timestamp, where MOR 309 could happen anytime in between “Merchant Accept” timestamp and “Food Collection Time” timestamp.

When the driver arrives 312, there may be a further delay before the order is collected (FCT) 314. The delay is called the wait time (DWT) 316. Lastly the driver drop-off time (DDT) 318 is between the FCT 314 and the time the driver arrive at consumer 320.

An allocation system may start allocating a driver as soon as an order gets created, and allocates a driver who takes the least time to reach the pick-up location. Many times, if the merchant requires a long time to prepare the food or is busy with long queues, this leads to high driver wait time at the Merchant.

In a delayed allocation system, the allocation process may be deliberately delayed such that driver will reach the merchant store just in time when the food is ready for collection.

The amount of delay is heavily dependent on the time required by merchant to prepare the food. Therefore, FPT prediction may be an important variable in delayed allocation implementation.

Accurate prediction of the FPT may be useful for one or more of the following reasons:

- If the driver reaches the merchant location earlier than food/item is ready - he/she may have to wait for long, wasting the driver's productive time and reducing driver efficiency. Similarly, if the driver reaches late in some types of

orders it may affect the order food quality (food getting cold/soggy, urgent medical orders may cause the order to be unusable), increases the total delivery time which in turn impacts the consumer experience.

Thus, we want to allocate the driver at the right time so he/she arrives at the merchant as close to exactly when the item is prepared.

- FPT is a significant component of total delivery time. An accurate FPT prediction leads to better ETA estimates.
- For self-pickup (take away) orders, the total ETA is equal to the FPT.
- Accurate FPT prediction sets a fair target for merchants when they are evaluated on their service quality in terms of on time preparation rate (percentage of order prepared within targeted time)

Therefore, in order to optimise the cost to serve an order and improve consumer experience, it may be useful for the system to more accurately predict FPT.

In the Merchant app 109 there may be at least two buttons to press in the user interface. There may be a first accept button beside each order. Then for an ongoing order, there may be a separate screen with a button marked MOR. The platform provider may predict FPT using historical data collected from merchants pressing the order ready button. However, the distribution of FPT ground truth data varies from merchant to merchant depending on their behaviour, placement of devices in restaurants, type of merchant (cloud-kitchen), demand-supply ratio, location/country etc.

We can classify the merchants based on their behaviour as

- MPBDA (Merchant start Preparing Before Driver Arrives)
These merchants are mostly well behaved and would typically start preparing order as soon as they accept it.

- **Non-MPBDA:**

Most of these merchants in this category wait for the driver to arrive before they start preparing the order. Such merchants tend to press the MOR button before the food is prepared, assuming this would make the driver arrive sooner. The reason for such behaviour can be their scepticism that driver may not arrive and food might be wasted, amongst others. And those merchants with uncertain behaviour are also classified into this category: they might prepare the order in advance sometimes and might only start to prepare the order after the driver's arrival without a fixed pattern.

For non-MPBDA merchants, delayed allocation (DA) leads to following vicious cycle of inaccurate predictions in Table 1:

DA enabled non-MPBDA order	Estimated FPT	Calculated FPT (Driver arrival time + 10 mins prep time)
Order 1	10mins	20mins
Order 2	20mins	30mins
Order 3	30mins	40mins
...		

Table 1

Consider an order that takes 10 mins to get prepared and its predicted FPT is 10 mins as well. System would delay the allocation such that the driver reaches the merchant at 10th minute. However, since the merchant is non-MPBDA, the merchant would only start preparing after the driver arrives. The preparation would be finished on 20th minute.

Further, consider the next order (order 2) with the same order items, the prediction system (Machine Learning (ML) model) would use historical actual FPT (calculated

FPT) to predict 20 mins. Once again, the driver reaches at 20th minute according to prediction but food takes another 10 mins to prepare, thus making calculated FPT = 30 mins. This feedback cycle continues and accuracy of prediction keeps decreasing with time, unless recalibrated.

For example, Machine Learning prediction models rely on historical ground truth (actual) FPT to make future predictions for an order. Prior art platform providers either don't make distinctions based on merchant behaviour, or use the MPBDA ground truth data to generalise on non-MPBDA as well. In other words, businesses don't have a way to accurately estimate FPT for Non-MPBDA merchants, because the historical data (from merchant button press) is inaccurate, leading to inaccurate predictions. Equally, the estimated FPT for MPBDA merchants is skewed by the non-MPBDA merchant historical data.

The percentage of non-MPBDA merchants can be as high as 30% depending on location/country. Thus, predicting accurate FPT estimates for non-MPBDA may be useful in some applications.

One or more embodiments relates to a system and method for identifying non-MPBDA merchants and/or more accurately predicting food preparation time for each. In one embodiment the system separates merchants into two categories: MPBDA and non-MPBDA, and generates FPT predictions from two separate ML models.

Phase 1: To identify Non-MPBDA merchants

The system 100 makes use of signals like Driver feedback, Driver wait time etc. to estimate the probability of a merchant being Non-MPBDA. The ML model may be a binary classifier which predicts the score/probability of a merchant being non-MPBDA. The system 100 then generates the list of all merchants beyond a certain threshold of probability, as non-MPBDA merchants. For example, all merchants

predicted with score > 0.8 will be classified as non-MPBDA. Since the merchant behaviour is likely to remain the same over weeks/months, the list of non-MPBDA merchants is updated periodically, such as weekly/bi-weekly.

To get verified ground truths, an operations teams may be deployed to manually surveyed merchant locations to identify non-MPBDA merchants. Then a ML model may be developed to predict the probability/score of merchants behaving as non-MPBDA. The following features may be used as input to the ML model:

1. Driver feedback:

When a driver reaches the merchant location, the driver communication device 106 user interface includes a feedback form to report whether order is already prepared, in-preparation, not started preparation.

2. Driver waiting time:

The median wait time for drivers is generally high for non-MPBDA merchants since drivers wait for entire duration of the order being prepared.

3. Percentage of orders prepared on time.

Non-MPBDA merchants will have large differences between estimated FPT and actual FPT, leading to less % orders prepared on time. An order is said to be prepared on-time if absolute deviation between actual preparation time and predicted preparation time is less than 5 mins, or some other threshold depending on the requirements of the application.

The machine learning model 400 in Figure 4 is trained to predict the score/probability of a merchant being non-MPBDA.

The model 400 may be supervised classification (because we use annotated ground truths). The loss function may be a cross-entropy loss (typical loss for classification). The model 400 may use Logistic regression. The model 400 output may be a probability of a merchant being non-MPBDA.

Phase 2: To predict FPT for Non-MPBDA merchants

For example, there may be 3 merchants X, Y, Z in the same geohash (locality). From phase 1, (to predict non-MPBDA) we predict that Y is non-MPBDA & (X,Z) are MPBDA. For phase 2, to predict FPT of non-MPBDA merchants, we train a machine learning model (Fig 4) using supervised regression (because we use target actual FPT). The loss function may be a: RMSE loss (typical loss for regressions). The model may be a XGBoost Regressor.

The supervised model may only use X,Z data for training, because their FPT ground truth is accurate. The training data may include:

- order level features (price, quantity etc.)
- Dish embeddings / order vector (dish/item related)
- If an order is placed at merchant X, we use X's real-time feature (table 2).
Likewise, if order was at merchant Z.
- Historical aggregated data i.e., mean & median of FPT for nearby MPBDA merchants including itself (X, Z) at geohash level, hour, day etc.
 - i.e. if order placed at X, we still aggregated X,Z both (geohash level).
So, we don't use aggregated features only for merchant X
 - Idea is to predict FPT without using that particular merchant's historical FPT data. This is intentionally done because at time of prediction for non-MPBDA merchant, historical FPT will be inaccurate.

Example: a training sample of an order for 2 chicken_rice bowl at X may look like:
[price: \$14, quantity: 2, dish_embedding of chicken_rice: [...], realtime info of merchant X, mean FPT of nearby MPBDA merchant (X,Z) at this hour/on this day etc.]

target label: actual FPT for this order: 15 mins

Inference can be made on non-MPBDA merchant Y.

Example: an order for 3 pasta bowls at Y, input to the model may look like:

[price: \$20, quantity: 3, dish_embedding of pasta: [...], realtime info of merchant Y, mean FPT of nearby MPBDA merchant (X,Z) at this hour/on this day etc.]

Prediction: estimated FPT for this order: 17 mins

Note that historical aggregated features (meaning FPT) used during training & inference are from the same geohash. We refrain from using merchant Y's historical FPT due to its inaccuracy. Hence, the model is estimating FPT of 3 pasta bowls as roughly the time it would take nearby MPBDA merchants (X,Z) to prepare it.

The system makes use of historical data of nearby merchants in the same geohash levels and historical FPT of similar merchants (if available). The FPT prediction is done assuming same order items are being prepared by other nearby MPBDA merchants.

The predicted FPT is used to generate accurate estimates of total ETA displayed on the user communication device 104 user interface. Besides, for non-MPBDA merchants, the driver is allocated to reach the restaurant as soon as possible to avoid delay in the start of food preparation.

The proposed features described below can be stored in some key-value database (like Amazon DynamoDB) and during inference, the system can find the corresponding features using the keys. Below are four different kinds of features that are used as input to ML model:

Order level features

Every order has following information which is used as input features to the model:

- Total quantity
- Total price
- City/Country
- Hour of the day

- Day of the week

Dish embedding features:

A Word2Vec algorithm may be used to represent textual information of item/dish names as numeric vectors in an n-dimensional space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Word2Vec embedding helps to enable the model to pick up on the behaviour of comparable foods in terms of cuisine and cooking methods. The system uses word2vec algorithm to generate vectors of n=20 dimensions for each order. Since an order may contain multiple items/dishes, we take weighted average (by quantity) of all items to determine order vector in Equation 1:

$$order\ vector = \frac{\sum item\ vector(i) * quantity(i)}{\sum quantity(i)} \quad (1)$$

Figure 5 shows a visualisation of different dish vectors, where each dish has 20 dimensions (properties). For example, the 11th dimension can represent the dish being non-vegetarian as we see this dimension being high value for dishes containing fish/chicken. The system precomputes the embedding vectors for each item and uses it at time of inference.

Real-time signals

The following real time signals in Table 2 are computed over sliding windows of given window-size. The window-size is decided based on frequency of signal change.

S.No.	Signal	Description	Window size (mins)
1	numInPrep	The number of pending orders (merchant accepted by not prepared) at merchant.	15 and 30
2	avgPrepTimeSecondsMOR	The average preparation time of all orders based on MOR.	15 and 30
3	lastOrderPrepTime	The last calculated order preparation time.	60
4	avgTotalPriceOrders	The average of the price from all orders.	15 and 30
5	lastOrderArrivedAtMex	The timestamp of the last order received by merchant.	60
6	avgTimeBetweenOrders	The average time between orders.	30 and 60

Table 2

Figure 6 shows a practical implementation of the inference system. The features 602 are computed 604 in real time from kafka streams 606 and ingressed to the Feature Store (DDB) 608 for usage during inference/prediction).

In Figure 6, Step 1 involves the Client 616 referring to the service 630 that calls the model endpoint. Clients can call the model using either HTTP or gRPC protocols. The input request in json format may contain:

[request_timestamp, merchant_id, items/dish, quantity, price, geohash etc.]

Step 2 & 3: The Request Handler 618 retrieves real time features 610 and aggregated features 612 from the feature store 608 based on geohash and merchant_id from DB (Amazon DynamoDB in this case).

Step 4: Preprocessing 620 is done before passing the request to model to comply with model API. It has several parts:

- ensure the features retrieved from DB are not missing. In case of missing features, there is a fallback mechanism to replace the missing value. If all features are missing (couldn't retrieve from DB), we return an error.
- ensure features are within proper bounds (range). This is done to avoid outliers. If not, its clipped between p1 (1st percentile) & p99 (99th percentile).
- calculate time_of_day, day_of_week etc. from request_timestamp
- convert prices to common currency (SGD)

Step 5: Pass the request to Model serving 622 as per API format (i.e., all features have to be of correct data types & in same order as at the time of training). The Model in this case is ONNX serving which is platform agnostic and low-latency prediction framework. The model responds with predicted FPT in real number format (float32).

Step 6 & 7: Postprocessing 624 is done before sending the predicted value back to client. This step is to check if predicted FPT is as per API specs, primarily data type and value range. In case of error in model response, we pass the same error to client.

Step 8: The final response in json format is passed to client using same HTTP/gRPC protocol.

Historical aggregated features

For non-MPBDA merchants, the system uses historical aggregated features of nearby MPBDA merchants as a proxy for the given merchant.

To identify nearby locations, the system uses Geohash technique which is a method used to encode geographic coordinates (latitude and longitude) into a short alphanumeric string.

The system pre-computes the aggregation features as Mean, Median and standard deviation of Food_prep_time and driver_wait_time over following level of aggregation:

- Geohash 6
- Geohash 5
- Geohash 5 x hour_of_day
- Geohash 6 x hour_of_day
- Geohash 5 x day_of_week
- Geohash 6 x day_of_week

The aggregation script runs daily at scheduled time, that aggregates the historical features and ingress to Feature Store for consumption at inference time (Figure 6)

The machine learning model is trained using above four types of features, to generate prediction for Food preparation time for each order for non-MPBDA merchant.

There are respective ML models that predict total ETA and each of the ETA components by considering various relevant factors such as driver supply demand situation, time of the day, day of the week, driver/merchant/eater location, traffic situation, vehicle type, etc.

For driver allocation, there is an optimisation model based on Kuhn-Munkres algorithm to match orders and drivers; additional consideration on allocation is that we also introduce certain amount of delay (i.e., delayed allocation) such that driver will reach merchant stores just in time when food is ready for collection.

Claims

1. A computer implemented method performed in a communication server apparatus for a delivery platform provider, the method comprising, under control of a processor of the communication server apparatus:
 - storing data relating to historical merchant behaviour;
 - determining which of the merchants exhibit standard behaviour and which exhibit non standard behaviour based on the stored historical merchant behaviour, wherein non standard behaviour includes behaviour which delays delivery; and
 - storing a list of standard behaviour merchants and non standard behaviour merchants based on the determination.
2. The method of claim 1 further comprising determining an order preparation time for a delivery, based on at least whether the merchant is a standard behaviour merchant or a non-standard behaviour merchant.
3. The method of claim 2 wherein the determination of order preparation time (FPT) for non standard behaviour merchants is based on historical merchant delivery preparation times (FPT) for standard behaviour merchants that are adjacent and/or of similar product type.
4. The method of claim 2 or 3 wherein the determination of order preparation time (FPT) for standard behaviour merchants is based on historical merchant delivery preparation times (FPT) for that merchant.
5. The method of any preceding claims wherein the behaviour which delays delivery includes delaying preparation of an order until a delivery vehicle arrives (non-MPBDA).

6. The method of claim 5 wherein the merchant standard behaviour and the merchant non standard behaviour is determined based on historical delivery driver feedback about the behaviour of each merchant, historical delivery driver wait time (DWT) at each merchant and/or historical percentage of orders prepared on time at each merchant.
7. The method of claim 6 wherein a machine learning model is trained to predict which merchants are non standard merchants using annotated merchant behaviour, historical delivery driver feedback about the behaviour of each merchant, historical delivery driver wait time (DWT) at each merchant and/or historical percentage of orders prepared on time at each merchant.
8. The method of claim 7 wherein the annotated merchant behaviour includes observed data about each merchant, the data including whether the merchant is observed to delay preparation of an order until a delivery vehicle arrives.
9. The method of any preceding claim wherein a delivery vehicle is allocated for an order for a non standard merchant as soon as possible.
10. The method of any preceding claim wherein a driver is allocated for a delivery from a standard merchant based on at least the estimated merchant delivery preparation time (FPT), and the estimated delivery vehicle to merchant travel time.
11. The method of claim 10 wherein a driver is allocated for a delivery from a standard merchant further based on minimising driver wait time (DWT) and/or completed food wait time.
12. A method of supervised training of a machine learning (ML) model to determine non standard merchants for a delivery platform provider including

storing manually annotated data regarding which merchants are non standard;

storing historical order information for each merchant including: delivery driver feedback about the behaviour of each merchant, historical delivery driver wait time (DWT) at each merchant and historical percentage of orders prepared on time at each merchant; and

training the ML model based on the stored manually annotated data and the stored historical order information;

wherein non standard merchants are determined based on behaviour which delays delivery.

13. A method of determining non standard merchants using a trained ML model, wherein the trained ML model has been trained according to the method of any of claims 1 to 12.

14. A method of supervised training of a ML model to determine order preparation time for non standard merchants using historical transaction data from standard merchants annotated with at least location and/or food type, wherein non standard merchants are determined according to the method of claim 13.

15. A method of estimating food preparation time for non standard merchants using a trained ML model, based on at least one parameter regarding the food order, wherein the trained ML model has been trained according to the method of claim 14.

16. A computer program or computer program product comprising instructions for implementing the method of any one of claims 1 to 15.

17. A computer program carrier carrying a computer program according to claim 16, wherein the computer program carrier is one of an electronic signal, optical signal, radio signal or non-transitory tangible computer-readable storage medium.
18. A non-transitory tangible computer-readable storage medium storing a computer program according to claim 16.
19. A system comprising
- a communication server;
 - at least one merchant communication device having an associated merchant and configured to accept a potential order and/or confirm an order is ready for pickup; and
 - at least one driver communication device having an associated driver vehicle; and
- communication network equipment configured to establish communication with the communications server, the at least one user communication device, the at least one merchant communication device and the at least one driver communication device;
- wherein the communications server comprises at least one processor(s), at least one memory, the server being configured, under control of one or more of the at least one processor(s), to execute instructions stored in one or more of the at least one memory to:
- store data relating to historical merchant behaviour including aggregated data on when potential orders accepted and/or when orders are ready for pickup from the merchant communication device and/or the driver communication device;
 - determine which of the merchants exhibit standard behaviour and which exhibit non standard behaviour based on the stored historical merchant behaviour, wherein non standard behaviour includes a

deviation above a threshold time between when the orders are accepted and ready for pickup, which indicates preparation delays; and
store a list of standard behaviour merchants and non standard behaviour merchants based on the determination.

20. The system in claim 19 wherein the at least one merchant communication device is configured to:

receive potential orders;
accept potential orders; and/or
confirm the order is ready.

21. The system in claim 19 or 20 further comprising at least one user communication device configured to receive an estimate of when the order is expected to arrive based on whether the merchant allocated for the order is a standard behaviour merchant or a non standard behaviour merchant.

22. The system in any one of claims 19 to 21 wherein the at least one driver communication device is configured to receive an instruction to drive to the merchant, where a timing of the instruction is based on whether the merchant allocated for the order is a standard behaviour merchant or a non standard behaviour merchant, a location of the driver vehicle, and/or a location of the merchant.

23. A communication server apparatus for a delivery platform provider, the communication server comprising at least one processor, the communication server apparatus being configured, under control of one or more of the at least one processors, to execute instructions, to:

store data relating to historical merchant behaviour;
determine which of the merchants exhibit standard behaviour and which exhibit non standard behaviour based on the stored historical merchant

behaviour, wherein non standard behaviour includes behaviour which delays delivery; and

store a list of standard behaviour merchants and non standard behaviour merchants based on the determination.

24. A user communication device, a driver communication device, or a merchant communication device for a transaction platform provider, the user communication device, the driver communication device, or the merchant communication device comprising a processor and a memory, the user communication device being configured, under control of the processor, to execute instructions stored in the memory, to implement the method according to claim 15.