A data-driven method to predict service level for call centers

Chenyu Hou  | Bin Cao  | Jing Fan

Department of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China

Correspondence
Bin Cao, Department of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China.
Email: bincao@zjut.edu.cn

Funding information
National Key R&D Program of China, Grant/Award Number: 2018YFB1402800; Fundamental Research Funds for the Provincial Universities of Zhejiang, Grant/Award Number: RF-A2020007

Abstract
In call centers, the service level is an important metric to measure the reasonability of the staffing schedule. Traditional service level calculation methods are based on the queue theory, which has very strict restrictions and is not suitable for real scenarios. Therefore, in this paper, a data-driven method to solve the service level prediction problem is proposed to be used. To this end, the relationship between service level and other factors, such as number of calls, number of agents, time, is explored. Then some features are extracted based on empirical analyses and propose to use decision tree based ensemble methods, like random forest and GBDT, to model the relationship between service level and input features. Finally, extensive experimental results show that the proposed method outperforms other baselines significantly. Especially compared with the traditional queue theory methods, our method improves the performance by 6% and 9% in terms of MAE and MAPE.

1 | INTRODUCTION

In recent decades, telephone call centers are becoming an important service platform to solve business problems for customers. For companies, they want to provide high-quality services while saving labor costs as much as possible. To this end, reasonable staff scheduling is necessary. Normally, when arranging the personnel, decision-makers will calculate an important indicator service level to evaluate whether the schedule is reasonable. Service level refers to the percentage of incoming calls that are answered within $\sigma$ seconds, where $\sigma$ is the user-defined parameter (e.g. 20 s). Therefore, a key challenge for staff scheduling is how to predict service levels accurately.

As far as we know, the current widely used method for service level prediction is the Erlang-C model [1–4] and Erlang-A model [5]. However, there are many strict constraints for adopting them. First, the Erlang-C model assumes that call arrivals follow a Poisson distribution, and they are serviced by agents with service times that follow an exponential distribution. Second, the Erlang-C model supposes all customers will wait as long as necessary for service without hanging up. Although Erlang-A improves the Erlang-C model by introducing a patience factor to model customers’ patience, it still not feasible for true scenarios. For example, Brown et al. [6] find that a time-inhomogeneous Poisson process fits their data. Some authors argued that call arrivals follow a stochastic process rather than Poisson process [7, 8]. All of these constraints cannot be met completely in the real world, which results in poor performance when they are used for true datasets.

Since complex realistic scenarios are difficult to be mathematically modelled accurately, in this paper, we attempt to use a data-driven method to solve the service level prediction problem. Due to the collection of historical data, we can train machine learning models that are able to learn the intrinsic relationships of the data through a supervised way. In order to obtain accurate prediction results, the key challenges are how to extract useful inputs from the raw data and how to select reasonable models.

In this paper, to overcome the above two challenges, we mainly complete the following works. First of all, we conduct empirical analyses to study the influences of various factors. Based on the empirical results, we extract three groups of features, namely primary features, intensity feature and date features. Next, we propose to use ensemble models, like Random Forest [9], GBDT and Xgboost to model the relationship between inputs and service levels. Finally, we conduct extensive experiments based on a true dataset to study the performance of proposed methods and analyze the feature importance.
Specifically, since historical data is time dependent, we first study the distribution of service levels at different times. We divide days into four types, namely weekday, weekend, holiday and overtime, respectively. By plotting the kernel density estimation curves, we find the distributions of service levels are different on four types of days. Besides, we analyze the distribution of the service levels against the time of the day. And the result shows that service levels are varying at different times. On the other hand, we also study the relationship between call arrivals, agents and service levels. We observe that the cases of low service levels often occur when there are few agents or few incoming calls. Besides, the number of agents is positively correlated with the number of call arrivals. Based on the empirical results, we extract three groups of features, namely primary features, intensity feature and date features. By feeding them into ensemble models, we can obtain accurate prediction results.

Finally, we conduct extensive experiments based on a real-world dataset coming from the TeleCom company. The experimental results show that our methods outperform the Erlang-A model and Erlang-C model significantly. Besides, we conduct detailed experiments to analyze the effects of all features. We find not all of the extracted features are useful for improving prediction accuracy and we analyze the reasons, which provides meaningful insights for other follow-up works.

The contribution of this paper can be summarized as follows:

- Instead of using mathematical modelling, we propose to use a data-driven method to solve the service level prediction problem. As far as we know, we are the first one to solve the service level prediction problem by the data-driven method.
- We conduct empirical studies based on the true dataset and find some insights.
- We extract some features and conduct extensive experiments to study their importance.
- We conduct extensive experiments based on a real-world dataset and compare the performance of our proposed methods with baselines. The experimental results demonstrate that our methods outperform the Erlang-A model by 6% and 9% in terms of MAE and MAPE.

The rest paper are organized as follows. Section 2 outlines the preliminaries, including our problem definition and some important concepts. Section 3 introduces our dataset and empirical studies. Section 4 presents the features and models we used. Section 5 presents detailed experimental results. Related works are presented in Section 6. Finally, we conclude our paper in Section 7.

2 PRELIMINARIES

This section presents a set of preliminaries that are important for understanding our problem. In particular, we first introduce the meaning of agents, the call processing flow and service level. Then we propose the formal problem definition.

2.1 Agents

In call centers, agents refer to the staff who answer calls and offer solutions for customers. In this paper, the work efficiency of each agent is different. For example, some agents are skilled thus they solve problems more quickly than those unskilled agents. Therefore, there is a criterion skill level to indicate the experience and efficiency of agents. The skill level of each agent is evaluated by decision-makers, which is not our concern. Finally, decision-makers arrange different numbers of agents at different times to answer incoming calls.

2.2 Call processing flow

The processing flow of call centers is depicted in Figure 1. When a call reaches the call center, the customer is first greeted by a voice response unit (VRU) which is an automated telephone answering system offering common functions such as bill inquiry. Approximately 80% of problems can be solved through the VRU while the remaining 20% of customers advance to the manual service. There are three cases:

1) Immediate Service: If there is an agent available and the waiting queue is empty, the customer will be served immediately. The waiting time of the customer can be ignored.
2) Holding Service: If there is no agent available and the waiting queue is not empty, the customer will advance to the service queue to wait. The service queue follows the first in first out (FIFO) principle. Therefore, the customer has to hold the phone until previous other customers leave the queue. The waiting time of the customer depends on the length of the queue and the service efficiency of agents.
3) Failed Service: When the customer is in line, he/she may lose patience and hang up. Since the customer is not served by an agent, this will be regarded as a service failure case.

2.3 Service level

Service Level: Service level is an important criterion indicating the service quality of a call center. On the one hand, call
centers want to improve the service level as much as possible. On the other hand, call centers want to save the staff budget. Over a given period of time, \( \sigma \)-s service level refers to the percentage of incoming calls that are answered within \( \sigma \) seconds, where \( \sigma \) is a threshold decided by call centers. The formula of service level is defined as below:

\[
l = \frac{|IS| + |\hat{IS}|}{|IS| + |\hat{IS}| + |FS|},
\]

where \(|IS|\), \(|\hat{IS}|\) and \(|FS|\) represent the number of immediate service, holding service and failed service, respectively. \(\hat{IS}\) is the call set where the waiting time of each call is less than \(\sigma\). We call \(V\) \(= IS \cap \hat{IS}\) as valid services.

2.4 Problem definition

Given the historical call processing dataset \(D = \{d_1, ..., d_n\}\), where \(d_i = (c_i, a_i, t_i, s_i)\) is a quintuple representing that at time \(t_i\), there are \(c_i\) calls and \(a_i\) agents with total \(a_i\) ability, the service level is \(s_i\). The goal of this paper is to train a model \(F^*\) using the data-driven method to infer service levels accurately.

It worth noting that to calculate the service level, we have to acquire the input information like call volumes and agents in advance. This involves a call volume forecasting method and staff scheduling method. However, these two methods are not our concern in this paper. In order to better understand this paper, suppose we have implemented these two methods. In the experiments, we use ground truth as predicted values of call volumes and agent information. Therefore, the optimal model \(F^*\) can be defined as below:

\[
F^* = \arg \min_{F} |F(nc, na, aa, T) - l|,
\]

where \(nc, na, aa\) are the predicted call volume, agent number and agent ability. \(l\) is the ground truth of service level while \(T\) is the corresponding timestamp.

3 DATA DESCRIPTION AND ANALYSIS

In this section, we first describe the real-world call traffic dataset that we used in our experiments. Then based on this dataset, we conduct empirical analyses to explore the characteristics of service levels.

3.1 Data description

The real-world dataset comes from a branch company of China Telecom, and the time span is from 2019-05-01 to 2019-12-31. The dataset contains detailed call records indicating the status of calls, including call-in times, transit times, service times and finish times. For the purpose of protecting user privacy, users’ phone numbers have been desensitized.

Note that although the call center is 24-h opening, we concentrate our effort on calls received between 9 AM and 10 PM; because this is when the call center is most active. There are two examples representing two service cases in Table 1. The first row indicates a call is served successfully. This call waits 58 s and receives a 245-s service. As for the second record, we can observe that the service time is NULL, which indicates this call is not processed by agents. This is because this customer lost patience when he/she waits for 67 s and hung up at 2019-05-04 14:13:00.

Based on the raw dataset, we then use 15 min as the granularity to divide the time interval of the day, and count the aggregation information of each time interval. The reason why we choose 15 min as the granularity is because 15 min is the atom unit of the staff scheduling in the China Telecom company. The examples of aggregation results in 2019-05-01 are illustrated in Table 2. We count call numbers, total service numbers, valid service numbers and numbers of abandon. From the examples of Table 2, we can conclude two observations: (1) Not all calls will be served successfully. There are some impatient customers who hang up while waiting. Therefore, the hypothesis of Erlang-C formula that customers will wait until they receive service is not reasonable in the real world. (2) Not all services are valid, some calls are not served until they have exceeded the valid time threshold. Our goal is to infer service levels, thus we should focus more on valid service numbers instead of total service numbers.

3.2 Empirical analysis

We conduct empirical analyses on the aggregation dataset and try to answer the following questions: (1) How does service levels vary in different times? (2) What is the relationship between service levels, call arrivals and agents?

3.2.1 Temporal properties of service levels

People have different patterns of behaviour on weekdays compared to weekend days [10], which may account for the incoming call numbers and work efficiency of workers. Therefore, we first study the distribution of service levels over different types of days. In this paper, we divide our daily days into four types: weekday, weekend, holiday and overtime. Note that holiday and overtime are prior to weekday and weekend, that is, if someday is weekday/weekend and holiday/overtime simultaneously, we will treat it as holiday/overtime. For example, although 2019-10-12 is Saturday, it is an overtime day to compensate for reduced working hours due to National Day. Therefore, the type of 2019-10-12 is overtime instead of a weekend.

Figure 2 presents the distribution of service levels among four types of days. We can find that service levels are beyond 80% in most cases, which indicates the current staff arrangement of the call center is reasonable in most cases. However, there are cases where the service levels are lower than 40%
TABLE 1  Examples of call records

| Phone number   | Call-in time          | Transit time   | Service time | Finish Time      |
|---------------|-----------------------|----------------|--------------|------------------|
| 136XXXXX633   | 2019-05-01 12:10:16   | 2019-05-01 12:12:11 | 2019-05-01 12:13:09 | 2019-05-01 12:17:14 |
| 199XXXXX078   | 2019-05-04 14:10:38   | 2019-05-04 14:11:53 | NULL         | 2019-05-04 14:13:00 |

TABLE 2  Examples of the aggregation results

| Time                | Call numbers | Total service numbers | Valid service numbers | Abandon numbers | Service level (%) |
|---------------------|--------------|-----------------------|-----------------------|-----------------|-------------------|
| [09:00:00, 09:15:00) | 591          | 560                   | 524                   | 31              | 88.66             |
| [09:15:00, 09:30:00) | 562          | 539                   | 498                   | 23              | 88.61             |
| ...                 | ...          | ...                   | ...                   | ...             | ...               |
| [20:45:00, 21:00:00) | 178          | 172                   | 157                   | 6               | 88.20             |

in Figure 2a and Figure 2b. This is because decision-makers misestimate the volumes of call traffic and efficiency of staff, resulting in work overload and low efficiency. What is more, we can observe that low service levels only occur on weekdays and weekends, which implies the call center provides better service on holidays and overtime days. Therefore, in order to infer service levels more accurately, we should take the type of days into consideration to prevent misestimation on weekdays and weekends.

The other temporal factor is intra-day time intervals. Weinberg et al. [11] had discovered similar within-day patterns of call arrivals on weekdays. In our dataset, we also find this characteristic. Figure 3 displays the average normalized within-day call arrivals for each day of the week. From the plot, we can find a common trend of call arrivals. First, the call arrivals increase between 9 AM and 10 AM. Then the call arrivals decrease sharply from 10 AM to 1 PM. At about 4 PM, the call center becomes busy again. Finally, as the day ends, the call arrivals gradually decrease to the lowest value. This common trend is very important since it tells the moments when the call center is busy. The decision-makers can arrange more agents to avoid these rush hours.

FIGURE 2  Service level distribution over different types of days
In this part, we study the relationship between service levels and call arrivals. According to our intuition, service levels should be low at the rush hour since there are too many call arrivals that need to be processed. However, according to our exploration, we find that the service levels are high at the rush hours. This is because decision-makers have arranged many workers to deal with high call arrivals. More workers mean more unexpected calls can be handled. On the contrary, in some spare time, there are only a few workers to answer calls. Some unexpected calls will cause a decline in service quality due to insufficient staff to handle them. Figure 5 presents pairwise relationships between service levels, call arrivals and agents. We can observe the following two insights.

First of all, the correlation between agents and call arrivals is roughly positive. The more call arrivals, the more agents are arranged. However, there are some exceptions where a few agents are assigned when call volumes are high or too many agents are assigned to only a few call arrivals. This is because staff scheduling is made based on the forecasted call arrivals. However, in some cases, call arrivals are not predicted accurately, which leads to insufficient or excess staffing.

Second, we can find an interesting phenomenon that is contrary to our intuition. According to our intuition, high call volumes are more likely to result in low service levels than low call volumes. However, from Figure 5, we can find that the low levels of service mostly occur when the number of agents or the amount of call arrivals is small. When the number of agents is small, it is more likely that there are no extra agents to handle sudden calls, which leads to a decline in service level. This is very easy to understand since few agents are unable to handle unexpected call arrivals resulting in low service levels. However, as for the latter cases, the reason is more complex. Since the staff scheduling is determined based on the forecasted call arrivals, the consequence of low forecasted call arrivals is only a few agents are arranged to handle calls. Besides, we have found the correlation between agents and call arrivals is roughly positive, which means the forecasted call volumes are consistent with true call volumes to some extent. Therefore, although we observe that low service levels occur when the amount of call arrivals is small, the intrinsic reason behind this is because there are a few agents resulting from low forecasted call volumes.

## 4 | METHODOLOGY

### 4.1 | Feature extraction

In this section, we will introduce what features we extracted to train models. There are three groups of features, namely primary features, the intensity feature, and date features.

**Primary features.** Primary features refer to the basic information coming from the call centers for calculating service levels for call centers, including the number of call arrivals, the number of agents, and total skill levels of agents. The skill level is a metric representing the experience of each agent. Normally, an agent with a higher skill level is more efficient than an agent with a lower skill level. Therefore, it is not enough to measure the processing efficiency of the call center only based on the number of agents. It is also necessary to take the skill levels of agents into consideration.

**Intensity feature.** In the historical records, we have observed that the correlation between the number of agents and call arrivals is roughly positive. This may be because that decision-makers want to keep the workload of agents within a certain range. Therefore, we design a new feature intensity to measure the workload of agents to study the relationship between service levels and workloads. *Intensity* represents the
average workload of agents, its formal definition is below:

$$\text{intensity} = \frac{\text{The number of call arrivals}}{\text{The number of agents}}.$$  \hspace{1cm} (3)

Figure 6a shows the distribution of the intensity. We can find that the intensity approximately ranged from 0 to 15. This validates our hypothesis that decision-makers keep the workload of agents within a certain range. Figure 6b shows that the intensity
is negatively correlated with the service level, with the Pearson coefficient of $-0.45$.

**Date features.** As discussed in Section 3.2, we have observed service levels show different patterns on different types of days and at different timestamps within the day. Therefore, in this part, we will extract intra-day features and inter-day features.

**Inter-day features:** The first inter-day feature is the type of the day. There are four types of days, namely weekday, weekend, public holiday and overtime. Public holidays refer to the rest time for celebrations and vacations uniformly stipulated by national laws. Note that our dataset contains three public holidays for a total of 13 days. As for the overtime days, they are used to compensate for the reduced working hours due to holidays. Besides, overtime days are set on weekends. In this paper, every day is labelled by one of the types. Note that holiday and overtime are prior to weekday and weekend, that is, if someday is weekday/weekend and holiday/overtime simultaneously, we will treat it as holiday/overtime. For example, although 2019-10-12 is Saturday, it is an overtime day to compensate for reduced working hours due to National Day. Therefore, the type of 2019-10-12 is over-time since we can observe that the call arrivals are slightly different on different days of the week. To study the patterns of the service level on different days more deeply, we add the day of the week into consideration.

**Intra-day feature:** As discussed in Section 3.2, we have observed that service levels are varying at different timestamps within the day. Therefore, it is important to take the timestamps into consideration. However, instead of considering hours and minutes separately, we design a new time index to combine the hours and minutes together. The time index $T$ is calculated as below:

$$ T = \text{hour} + \frac{\text{min}}{60}. $$

The advantage of the time index is that it allows models to judge the sequence of timestamps directly based on the values.

**Summary.** We totally extract seven features, including the amount of call arrivals, the number of agents, and total skill levels of agents, intensity, the type of the day, the day of the week, and time index.

### 4.2 Models

In this section, we propose to use decision tree-based ensemble learning models, like the random forest, gradient boosting decision trees (GBDT) and Xgboost. Compared to other models, decision trees are more intuitive and easy to explain. Besides, we analyze the importance of inputs according to decision trees. As for ensemble learning models, they train a set of base learners (decision trees) and then have those learners ‘vote’ in some fashion to predict the results of inputs. Many existing works [12–14] have proved that ensemble models are more accurate than single models. There are two main ensemble algorithms, namely bagging and boosting. In the rest part of this section, we will introduce them simply.

**Algorithm 1 Bagging**

**Input:**
- Training data $D = (x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$;
- Base learning algorithm $L$;
- Number of base learners $N$

**Output:**
- Fraction $f$ of the entire dataset to generate subset.

```
1. Initialize $F^*$ to empty;
2. for $t = 1, 2, \ldots, T$ do
   3. Training subset $D_s$ ← select $f$ percent of $D$;
   4. Base learn $L_t$ ← $L(D_s)$;
   5. Add $L_t$ to $F^*$
3. Return $F^*$
```

**Algorithm 2 Boosting**

**Input:**
- Training data $D = (x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$;
- Base learning algorithm $L$;
- Number of base learners $T$;
- Loss function $O$;

**Output:**
- The ensemble model $F_T$

```
1. Initialize $F_0 ← L(D)$;
2. for $t = 1, 2, \ldots, T$ do
   3. // Compute pseudo residuals:
   4. $r_i = -\frac{\partial L(y_i, F_{t-1}(x_i))}{\partial F_{t-1}(x_i)}$, for $i = 1, \ldots, m$;
   5. $h_t ← L\left(\{x_i, r_i\}_{i=1}^m\right)$;
   6. $\rho_t = \arg\min_{\rho} \sum_{i=1}^{N} O(y_i, F_{t-1}(x_i) + \rho h_t(x_i))$;
   7. $F_t(x) = F_{t-1}(x) + \rho h_t$;
3. Return $F_T$
```

**Bagging,** which stands for bootstrap aggregating [15], is a simple yet effective ensemble approach, where the most representative algorithm is random forest [9]. It selects different training data subsets from the entire training dataset randomly and then trains multiple base learners independently. Each training data subset is used to train a single base learner. Finally, all of these learners are combined by taking a majority vote to get the final decision. The pseudo-code of bagging is shown in Algorithm 1.

**Boosting,** Different from the bagging where each base learner is trained independently, in the boosting, each base learner is learned iteratively. Besides, each base learner is trained based on the negative gradient of the loss, so as to reduce the error of previous learners. The pseudo-code of boosting is presented in Algorithm 2. First, we can obtain an initial base learner $F_0$ based on the given training dataset $D$. Then we conduct iteration to obtain a better ensemble learner. During each iteration, we first calculate the pseudo-residuals of previous learners. Then we train a new base learner to fit the residuals. Next, we will conduct a linear search to find the optimal step $\rho$. Finally,
we can obtain the final $F_T$, which consists of $T + 1$ base learners with different weights $\rho$.

## 5 EXPERIMENTS

### 5.1 Experimental setting

**Our models:**

- **Random Forest Regressor:** Random forest regressor is a bagging ensemble method based on the random forest for regression problems.
- **GBDT [16]:** GBDT is an ensemble model of decision trees, which are trained in sequence. In each iteration, GBDT learns the decision trees by fitting the negative gradients.
- **Xgboost [17]:** Xgboost is an optimized distributed gradient boosting method designed to be highly efficient, flexible and portable.

**Baseline:** We compare the following baselines:

- **Historical Average (HA):** HA averages the service levels of the same timestamp in the last 7 days as the prediction result.
- **K-Nearest Neighbour (KNN):** Given the agents, call arrivals and skill levels, the KNN finds the five nearest records from the training data and averages their service levels as the prediction result.
- **Erlang-C [2]:** The Erlang-C model supposes call arrivals follow a Poisson distribution, and service time of agents follows an exponential distribution. Besides, Erlang-C assumes all customers will wait until they are served. **Erlang-A [5]:** The Erlang-A model is an extension of the Erlang-C model where customers may lose patience and hang up when they are waiting for service. The patience time of customers follows an exponential distribution.
- **Support Vector Regression:** Support vector regression (SVR) is a powerful algorithm that gives us the flexibility to define how much error is acceptable in our model and will find an appropriate line (or hyperplane in higher dimensions) to fit the data.
- **Linear Regression (LR):** In linear regression, the relationships between the dependent variable and one or more explanatory variables are modelled using linear predictor functions.
- **Multi-Layer Perceptron (MLP):** Multi-layer perceptron (MLP) is a three-layer fully connected network.

**Preprocessing:** We use min–max normalization to normalize the continuous features (e.g. number of agents) into $[0, 1]$, and use one-hot encoding to transform discrete features (e.g. day type). In the evaluation, we rescale the predicted values back to normal values.

**Hyper-parameters:** In our experiments, we conduct the grid-search for every model to find the optimal parameters based using training data. The detailed selection is shown in Table 3. We adopt the off-the-shelf models from the scikit-learn package and the Xgboost package. Except for these parameters listed in Table 3, other parameters are default as packages set. We conduct five-fold cross-validation using the first 80% of the dataset to select the best values of parameters.

**Metrics:** We adopt two metrics to evaluate performance of methods, they are mean absolute error (MAE) and mean absolute percent error (MAPE). The definition of two metrics are as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$  \hspace{1cm} (5)

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left|\frac{y_i - \hat{y}_i}{y_i}\right|.$$  \hspace{1cm} (6)

where $y_i$ and $\hat{y}_i$ are ground truth and the corresponding predicted value, and $n$ is the number of samples in the test set. Note that MAE is more affected by larger values, while MAPE receives more punishments from smaller values. We split the entire dataset into five subsets, and conduct five-fold cross-validation to compare the final performance of all methods.

### 5.2 Performance comparison

In this section, we compare the performances of our proposed models against baselines. We calculate the mean values and standard deviation of metrics to study the accuracy and robustness of different models. Besides, to study the effect of date features, we compare the performance of each model with and without date features. The detailed results are shown in Table 4.

First, we can see that GBDT and Xgboost achieve the best performance regardless of whether the date features are considered, followed by the Random Forest regressor. Second, the worst method is the Erlang-C formula, which indicates the Erlang-C formula cannot fit in the complex practical scenarios

---

1. https://scikit-learn.org/
2. https://github.com/dmlc/xgboost
In this section, we will quantitatively study the importance of date features. In this section, we will quantitatively study the importance of each feature in decision tree-based models (GBDT, Random Forest and Xgboost). Since we use one-hot encoding to transform discrete date features, we calculate their importance by averaging the importance of their one-hot features. Besides, since we conduct five-fold cross-validation, the importance of each feature is different in five experiments. Therefore, the final importance of each feature is the mean of the five experimental results.

Figure 7 shows the importance results. Each bar represents one feature while the height of the bar indicates the importance of the corresponding feature. For the sake of convenient observation, we sort features descendingly according to their importance. Besides, each feature is represented with a fixed color.

From Figure 7, we can observe that the most important feature is intensity, followed by time index. This means that intensity and time index contribute to improving the accuracy of prediction. On the contrary, the other two date features type of the day and day of the week are the least important. As for the primary features, call arrivals and skill levels are competitive, the number of agents is the least important among them. This indicates that the key factor influencing processing efficiency is the skill levels rather than the number of agents.

**5.4 Inside the date features**

In this section, we study the feature importance from the other aspect. By excluding features from the model, we compare the performance shift to study their contribution to the task. To this end, we design the following five variants: 1) Variant-1 (V1), where we take all features into consideration. 2) Variant-2 (V2), where the type of day is excluded. 3) Variant-3 (V3), where we exclude the day of the week. 4) Variant-4 (V4), where the time index is not considered. 5) Variant-5 (V5), where we remove the intensity feature. Note that primary features are included in all variants.

We compare the performance of these variants using different baselines. The comparison results are plotted in Figure 8. There are five subfigures representing the performance of five

**TABLE 4 Performance comparison**

| Method     | MAE (No date feature) | MAE (Use date feature) | MAPE (No date feature) | MAPE (Use date feature) |
|------------|------------------------|-------------------------|------------------------|-------------------------|
| HA         | 10.26% ± 1.39%         | 13.56% ± 2.19%          | /                      | /                       |
| KNN        | 7.76% ± 0.69%          | 10.14% ± 1.38%          | /                      | /                       |
| Erlang-C   | 40.11% ± 9.10%         | 49.38% ± 9.97%          | /                      | /                       |
| Erlang-A   | 12.63% ± 1.33%         | 17.67% ± 2.67%          | /                      | /                       |
| SVR        | 7.91% ± 0.75%          | 10.13% ± 1.48%          | 7.81% ± 0.53%          | 10.09% ± 1.17%          |
| LR         | 8.04% ± 0.86%          | 10.66% ± 1.64%          | 7.92% ± 0.71%          | 10.47% ± 1.48%          |
| MLP        | 7.75% ± 1.27%          | 10.06% ± 1.73%          | 7.07% ± 0.73%          | 9.32% ± 1.44%           |
| GBDT       | 7.16% ± 0.79%          | 9.44% ± 1.50%           | 6.66% ± 0.38%          | 8.73% ± 1.01%           |
| RF         | 7.16% ± 0.84%          | 9.44% ± 1.52%           | 6.77% ± 0.46%          | 8.89% ± 1.09%           |
| Xgboost    | 7.17% ± 0.81%          | 9.45% ± 1.52%           | 6.66% ± 0.36%          | 8.72% ± 1.01%           |
models. The line in each subfigure indicates the MAE while the upper bound and the lower bound denote the results of MAE ± MAPE, respectively.

First of all, we can observe that the V4 that excludes the time index performs worst in all cases. This means that the time index is very important to infer the service level accurately. The reason behind this may be that in the historical data, the service levels are varying at different times. The time index provides time information that drives models to learn independent patterns of service levels. Therefore, models can achieve better performance by considering the time index.

However, the other date feature type of the day is not helpful for this task. As we can see from Figure 8, V1 even achieves slightly better performance than V2. The possible reason is that historical data contains little holiday and overtime records. Due to the lack of historical data, models cannot learn comprehensive patterns of service levels under different types of days. Therefore, considering the type of the day will result in biased forecasting which may harm the accuracy.

As for the day of the week, we observe that the performances of V3 and V1 are neck and neck. This means the feature day of the week is helpless for improving the accuracy. To analyze the reason, we study the distributions of various features. Figure 9 shows the distributions of four types of data against days of the week, namely service levels, call arrivals, agents and intensity. As we can see from Figure 9, the distributions are almost the same on different days of the week. Therefore, considering the day of the week is meaningless since there is no obvious difference between different days.

Last but not least, there is an interesting phenomenon that the performance of V5 is competitive with that of the V1. This implies that the intensity feature is not useful for improving forecasting accuracy. However, in Section 5.3, we have verified that intensity is the most important feature. The reason behind this is that because we have fed the call arrivals and agents into models, such non-linear models like GBDT, RandomForest and Xgboost are able to construct a virtual feature that is equivalent to intensity using these two features. Therefore, in non-linear models, the feature intensity is not meaningful. However, it still improves the accuracy of the linear regression model (as shown in Figure 8e).

6 RELATED WORK

The purpose of this paper is to use a data-driven method to solve the service level prediction problem. And call centers are often modelled as queuing systems [18]. Therefore, we are going to introduce related works from the following three aspects: queue theory, data mining applications in call centers and ensemble methods.

Queue Theory: Queuing models are used to estimate system performance so that the appropriate staffing level can be determined in order to achieve a desired performance metric such as the Average Speed to Answer, or the Abandonment Percentage. The most classic queuing models used for call centers are the Erlang-B model [19], Erlang-C model [1, 2] and Erlang-A model [5].

Erlang-B model assumes Poisson arrivals and exponentially distributed service times. However, there is no waiting queue in the Erlang-B model, that is, the Erlang-B model assumes that any blocked calls are canceled immediately. Due to this reason, the Erlang-B model is not going to be explored further [20]. Different from the Erlang-B, the Erlang-C assumes that blocked calls will be added to a waiting queue and stay there until they are served. By doing this, the Erlang-C model is more suitable for real scenarios. However, the Erlang-C model neglects the possibility that calls abandon the queue before they get service. To solve this issue, the Erlang-A model is proposed. Erlang-A is an extension of the Erlang-C model. It assumes each caller posses
an exponentially distributed patience time with mean \( \theta^{-1} \). If the waiting time exceeds the customer's patience, the caller will abandon the queue and hang up [21]. Gurvich et al. extended Erlang-A model [22] by considering differentiated service level.

According to Kendall's notation [23], there are some more complex queuing systems, like M/G/1 [24], GI/M/k [23], GI/G/1 [25], etc. However, all these systems require strong hypotheses about call arrivals or service time, which is not suitable for complex real scenarios.

**Data mining applications in call centers:** In the field of call centers, most works about data mining aim to forecast call arrivals. Bianchi et al. [26] used ARIMA models with intervention analysis to forecast telemarketing call arrivals. Shen and Huang [27] used the singular value decomposition method to forecast inter-day call arrivals. Barrow [28] proposed to use the seasonal moving average method to forecast intra-day call arrivals. Cao et al. [29] solved the long-term call traffic forecasting problem by taking seasonal dependencies into consideration. As for the performance prediction of call centers, Paprzycki et al. [30] proposed to use a data mining method to predict customer service satisfaction. As far as we know, there is no existing work that proposed to use the data-driven method to predict service levels.

**Ensemble methods:** Various ensemble methods has been used for many applications, like text genre classification [31], web page classification [32], keyword extraction [33] sentiment classification [34]. Onan [35] proposed an ensemble scheme that first partitions the training dataset into multiple subsets using clustering methods, and then trains multiple classifiers. Each classifier is trained on the diversified training subsets and the predictions of individual classifiers are combined by the majority voting rule. Brunese et al. [36] proposed to use ensemble learning to detect brain cancer based on radiomic features.

### 7 CONCLUSION

In this paper, we propose a data-driven method to solve the service level prediction problem. Different from the traditional theory based method, we do not set any hypothesis about input factors while we use machine learning models to learn the relationship between service level and input factors from the historical data. To this end, we first explore the relationship between service level and input factors, such as the number of calls, the number of agents, time. According to detailed exploration, we find temporal properties of service level and influence of call arrivals and agents on service level. Based on the empirical analyses, we extract three groups of features, namely primary features, intensity feature and date features. As for models, we propose to use decision tree based ensemble learning models due to their interpretability and high accuracy. Finally, according to extensive experiments based on true data, we prove the effectiveness of our proposed methods, we conduct detailed experiments to analyze the effects of all features. We find not all extracted features are useful for improving prediction accuracy and we analyze the reasons, which provides meaningful insights for other follow-up works. In the future, due to the success of word embedding [37–39] in natural language processing, we will explore embedding features of date features rather than one-hot encoding. Besides, we will explore more solid feature selection methods [40, 41] to select useful features in advance.

**ORCID**

Chenyu Hou  https://orcid.org/0000-0002-2176-4435  
Bin Cao  https://orcid.org/0000-0003-1062-6309

**REFERENCES**

1. Erlang, A.K.: Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. Post Off. Electr. Eng. J. 10, 189–197 (1917)
2. N.Gans, G. Koole, A. Mandelbaum,: Telephone call centers: Tutorial, review, and research prospects. Manuf. Serv. Oper. Manag. 5(2), 79–141 (2003)
3. Liao, S., et al.: Staffing a call center with uncertain non-stationary arrival rate and flexibility. OR Spectrum 34(3), 691–721 (2012)
4. Robbins, T.R., Harrison, T.P.: A stochastic programming model for scheduling call centers with global service level agreements. Eur. J. Oper. Res. 207(3), 1608–1619 (2010)
5. Robbins, T.R.: Evaluating the fit of the Erlang a model in high traffic call centers. In: 2016 Winter Simulation Conference (WSC). IEEE, pp. 1790–1801 (2016)
6. Brown, L., et al.: Statistical analysis of a telephone call center: A queuing-science perspective. J. Am. Stat. Assoc. 100(469), 36–50 (2005)
7. Whitt, W.: Staffing a call center with uncertain arrival rate and absenteeism. Prod. Oper. Manag. 15, 88–102 (2006)
8. Aksin, Z., Armony, M., Mehrotra, V.: The modern call center: A multidisciplinary perspective on operations management research. Prod. Oper. Manag. 16(6), 665–688 (2007)
9. Breiman, L.: Random forests. Mach. Learn. 45(1), 5–32 (2001)
10. Zawadzki, M.J., et al.: Understanding stress reports in daily life: A coordinated analysis of factors associated with the frequency of reporting stress. J. Behav. Med. 42(3), 545–560 (2019)
11. Weinberg, J., Brown, L.D., Stroud, J.R.: Bayesian forecasting of an inhomogeneous poisson process with applications to call center data. J. Am. Stat. Assoc. 102(480), 1185–1198 (2007)
12. Dietterich, T.G., et al.: Ensemble learning, The Handbook of Brain Theory and Neural Networks 2, 110–125 (2002)
13. Ghimire,D., Lee, J.: Extreme learning machine ensemble using bagging for facial expression recognition. JIPS 10(3), 443–458 (2014)
14. Polikar, R.: Ensemble based systems in decision making. IEEE Circuits Syst. Mag. 6(3), 21–45 (2006)
15. Breiman, L.: Bagging predictors. Mach. Learn. 24(2), 123–140 (1996)
16. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. 1189–1232 (2001)
17. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
18. Allen, A.O.: Probability, Statistics, and Queuing Theory. Academic Press, Cambridge (2014)
19. Angus, I.: An introduction to Erlang b and Erlang c. Telemanagement 187, 6–8 (2001)
20. Barunik, J., Hartmann, M., Polacek, R.: Properties of a modelled call centre. Int. J. Electr. Eng. 71(2), 96–102 (2020)
21. Mandelbaum, A., Zeltyn, S.: Service engineering in action: The palm/Erlang-a queue, with applications to call centers. In Advances in Service Innovations. Springer, Berlin pp. 17–45 (2007)
22. Gurvich, I., Armony, M., Mandelbaum, A.: Service-level differentiation in call centers with fully flexible servers. Manage. Sci. 54(2), 279–294 (2008)
23. Kendall, D.G.: Stochastic processes occurring in the theory of queues and their analysis by the method of the imbedded markov chain. Ann. Math. Stat. 338–354 (1953)
24. Madan, K.C.: An M/G/1 queue with second optional service. Queueing Syst. 34, 37–46 (2000)
25. Pollaczek, F.: Problèmes stochastiques posés par le phénomène de formation d’une queue d’attente à un guichet et par des phénomènes apparentés (1957)
26. Bianchi, L., Jarrett, J., Hanumara, R.C.: Improving forecasting for telemarketing centers by arma modeling with intervention. Int. J. Forecast. 14(4), 497–504 (1998)
27. Shen, H., Huang, J.Z.: Interday forecasting and intraday updating of call center arrivals. Manuf. Serv. Oper. Manag. 10(3), 391–410 (2008)
28. Barrow, D.K.: Forecasting intraday call arrivals using the seasonal moving average method. J. Bus. Res. 69(12), 6088–6096 (2016)
29. Cao, L., et al.: Forecasting long-term call traffic based on seasonal dependencies. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing. Springer, Berlin pp. 231–246 (2019)
30. Paprzycki, M., et al.: Data mining approach for analyzing call center performance. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer pp. 1092–1101 (2004)
31. Onan, A.: An ensemble scheme based on language function analysis and feature engineering for text genre classification. J. Inform. Sci. 44(1), 28–47 (2018)
32. Onan, A., Korukoğlu, S., Bulut, H.: Classifier and feature set ensembles for web page classification. J. Inform. Sci. 42(2), 150–165 (2016)
33. Onan, A., Korukoğlu, S., Bulut, H.: Ensemble of keyword extraction methods and classifiers in text classification. Expert Syst. Appl. 57, 232–247 (2016)
34. Onan, A., Korukoğlu, S., Bulut, H.: A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification. Expert Syst. Appl. 62, 1–16 (2016)
35. Onan, A.: Hybrid supervised clustering based ensemble scheme for text classification. Kybernetes 46(2), 330–348 (2017)
36. Brunese, L., et al.: An ensemble learning approach for brain cancer detection exploiting radiomic features. Comput. Methods Programs Biomed. 185, 105134 (2020)
37. Onan, A.: Two-Stage Topic Extraction Model for Bibliometric Data Analysis Based on Word Embeddings and Clustering. IEEE Access 7, 145614–145633 (2019)
38. Onan, A.: Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. Concurr. Comput. Practice and Experience, e5909 (2020)
39. Onan, A., Toçoğlu, M.A.: A term weighted neural language model and stacked bidirectional lstm based framework for sarcasm identification. IEEE Access 9, 7701–7722 (2021)
40. Onan, A., Korukoğlu, S.: A feature selection model based on genetic rank aggregation for text sentiment classification. J. Infor. Sci. 43(1), 25–38 (2017)
41. Onan, A.: A fuzzy-rough nearest neighbor classifier combined with consistency-based subset evaluation and instance selection for automated diagnosis of breast cancer. Expert Syst. Appl. 42(20), 6844–6852 (2015)

How to cite this article: Hou, C., Cao, B., Fan, J.: A data-driven method to predict service level for call centers. IET Commun. 1–12 (2021).
https://doi.org/10.1049/cmu2.12192