Adaptive Call Center Workforce Management With Deep Neural Network and Reinforcement Learning

WUTTIPONG KUMWILAISAK[1], **(Member, IEEE), SARAVUT PHIKULNGOEN**[1],
JITPINUN PIRIYATARAVET[1], **NATTANUM THATPHITHAKKUL**[2], **AND**
CHATCHAWARN HANSAKUNBUNTHEUNG[2]

[1]Department of Electronics and Telecommunication Engineering, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand
[2]Assistive Technology and Medical Devices Research Center, National Science and Technology Development Agency, Pathum Thani 12120, Thailand

Corresponding author: Nattanum Thatphithakkul (nattanun.tha@nstda.or.th)

ABSTRACT Workforce management is one of several critical issues in a call center. A call center supervisor must assign an adequate number of call agents to handle a high volume of time-variant incoming calls. Without effective staff allocation, improper workforce management can degrade service quality and reduce customer satisfaction. This paper presents a novel call center workforce management based on a deep neural network and reinforcement learning (RL). The proposed method first uses a deep neural network to learn and predict call center traffic characteristics. The deep neural network consists of a Long-Short Term Memory (LSTM) network and a Deep Neural Network (DNN) capturing non-linear call traffic behaviors. The expected traffic parameters are supplied into the Erlang A model, which calculates important service metrics, including a call abandonment probability and the average response time. This paper applies a reinforcement learning framework using the Q-learning algorithm to establish the optimal starting times of call agent shifts and their associated call agent numbers by maximizing a defined reward function to handle dynamic call center traffic. The objective of these findings is to maintain the quality of service of a call center throughout working hours. The proposed method surpasses experienced human supervisors and previous workforce management schemes in terms of achieved qualities of service and average waiting time from experimental results under actual call center data.

INDEX TERMS Reinforcement learning, Q-learning algorithm, fully connected network, long short-term memory network, Erlang A, Thai telecommunication relay services (TTRS).

I. INTRODUCTION

Operating a call center is a very challenging issue because of its operating nature 24 hours and seven days a week. Incoming calls from customers require actual assistance day and night, and call agents must be present to provide services. Instead of hiring a large number of call agents, call center supervisors must design a workforce management scheme to schedule working hours and to allocate a proper call agent number for each service-time interval [2], [3]. This

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa Rahimi Azghadi.

task becomes more complicated when a call center possesses several services. For example, in Thai Telecommunication Relay Services or TTRS [1], a multimedia call center providing relay services to Thai hearing impaired people, supervisors must allocate call agents for different types of services, including real-time video relay services, real-time text relay services, and short message relay services. In general, the objective of workforce management in a call center including TTRS is to leverage customer satisfaction under limited human resources. Designing a workforce management procedure for a call center must consider the following challenges.

- High turnover rates: A call center often has a high rate of employee turnover all over the world. A call center's functioning is impacted by high turnover rates since new call agents must be recruited and trained to replace resigned staff. When a call center has to deal with highly competent call agents, the situation becomes much more challenging. For example, TTRS call agents must manage relay services between the hearing impaired and regular individuals using sign language. Replacing resigned employees would necessitate a lengthy period of training.

- Workload forecasting: When a call center is understaffed, customer satisfaction generally drops, while call agents are overwhelming, leading to high turnover rates. In contrast, when a call center is overstaffed, many call agents are idle, and a call center will waste human resources. Workload forecasting is generally carried out by supervisors' experiences in some call centers, such as TTRS, resulting in inaccurate workload forecasting and overloading supervisors when a call center has several services and many call agents.

- Human errors: Supervisors and call agents who are overworked frequently make mistakes in planing a call center, preserving client information, and providing services. When all operations are performed manually, the situation becomes even worse. For instance, in TTRS, supervisors normally spend two or three consecutive days for human resource planing. Supervisors are unable to work on other things during this planning period since they must focus on this laborious duty. Despite the fact that they put forth a lot of effort, human resource planning is prone to blunders.

- Service level agreement: The majority of call centers have their own set of quality-of-service targets, with some performance indicators recommended by industry standards. For example, TTRS adopts the recommendations from the International Telecommunication Union (ITU) [39], [40] as a guideline to set up its service level agreement. In general, there are multiple performance metrics used to assess call center performance such as customer's call waiting time, call abandonment rate, and average speed of answer. For example, TTRS sets up different service level agreements for their video relay services, text relay services, and short message relay services. It is very difficult for human supervisors to incorporate all these metrics from different service types and to derive the best workforce management scheme that can satisfy both available call center resource and customer satisfaction.

In practice, human supervisors commonly employ a trunking theory via Erlang-B and Erlang-C [4] together with their experiences to manage human resources. However, these conventional tools do not consider some vital call center performance metrics such as abandoned calls. As a result, an improved version of Erlang-C called Erlang-A [5] was proposed by including the rate of abandoned calls in its formulation. Unfortunately, these formulations are relied on the call center's static statistics and may not well represent actual dynamic traffics [6]. There were some researches proposed to be alternative tools for workforce management and resource allocations in various applications. The work from [7] introduced a Decision Support System (DSS) to handle a workforce management problem using linear optimization. The DSS outputs an appropriate number of service intervals (shift) and their corresponding agent numbers. However, the call center's parameters may not have a linear relation with the outputs. Hence, the DSS possibly can not provide the best solution. In addition, the DSS is still based on static statistics, which may not be efficient when we employ it in a dynamic environment. In practice, there are many factors affecting call center operations, such as different durations within a working day, holidays, and a number of call agents. Integrating several factors to derive suitable operating measures is not straightforward since we need to consider many variables, which may not be linearly related. Machine learning is widely known for producing a non-linear relationship between many inputs and outputs [8]–[11], [37]. To be more specific, it can extract hidden clues from available data that human supervisors may miss. A neural network [12] with Tabu Search [13] was proposed to derive a job schedule. However, this work was operated under a small number of parameters and constraints, which may not be suitable to apply to a call center's workforce management problem. In [14], reinforcement learning was introduced to manage a maintenance schedule. A neural network and correlative variables were adopted for workforce management in banking operation centers [15]. Similarly, some other studies including Deep Q-learning [16] Even though the works from [14]–[16] tackled practical problems containing many parameters and constraints, their formulations were not designed to handle dynamic statistics, and the derived schedules were considered as suboptimal solutions. In [17], deep Q-learning was adopted to solve the problem of multi-resource multi-machine job scheduling. Moreover, the authors in [18] proposed the optimized production systems and global production scheduling. However, this research was not fully automatic and required human experts to identify partial labeling.

This paper presents a novel call center workforce management based on a deep neural network and reinforcement learning (RL) to solve the limitations of existing resource allocation and workforce management methods. A deep neural network is used to learn and predict call center traffic characteristics. It consists of a Long-Short Term Memory (LSTM) network and a Deep Neural Network (DNN) capturing non-linear traffic characteristics. The expected traffic parameters are supplied into the Erlang-A model, which calculates important service metrics, including a call abandonment probability and the average response time. To handle dynamic traffic, this study applies an RL using the Q-learning algorithm for automatic workforce allocation. An RL establishes the optimal starting time of each call agent shift and the number of its associated call agents by maximizing a defined

reward function. The objective of these findings is to maintain the quality of service of a call center throughout working hours under various service types and multiple service level agreements. The contributions of this paper can be summarized as follows.

1) We propose a new method for estimating call center traffic using a combination of a recurrent neural network and a fully connected network.
2) We incorporate multiple constraints in actual call center operations to our proposed traffic prediction framework.
3) We formulate an adaptive workforce management method based on an RL and employ the Q-learning algorithm to search for the optimal workforce management solutions.

This paper is organized as follows. Section II describes the proposed call center traffic forecasting with the proposed deep neural network and the techniques in preparing data for traffic forecasting. Section III describes the dynamic workforce management with reinforcement learning framework. This section includes a policy searching technique of the formulated problem. The experimental results to access the performance of the proposed method from the actual call center data are in Section IV. Finally, concluding remarks are in Section V.

II. CALL CENTER TRAFFIC PREDICTION WITH LSTM AND FULLY CONNECTED NETWORKS

A. DATA PRE-PROCESSING MODULE

In a call center, a number of available channels is restricted by a number of working call agents, $n_a(t)$, at a specific time t. If all call agents are occupied, incoming calls need to wait in a queue. Let $n_q(t)$ is a number of calls that a call center's queuing system can handle at time t. As a result, the maximum number of channels that a call center can provide services is equal to $n_a(t) + n_q(t)$. If there are some available channels in a queuing system, an incoming call triggers a ringing notification, and it is transferred to an available call agent. A call agent picks up a call and gives a service to a customer. After completing a service, a call agent and a customer release an occupied channel. Then, an awaiting call in a queue can proceed to receive a service. However, if all call agents and a queuing system are fully occupied, a customer will receive a busy signal. The described call center procedure can be illustrated in Fig.1.
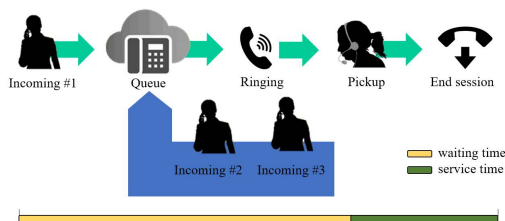

FIGURE 1. A general process of a call center.

TABLE 1. An example of the dataset from three services of the TTRS call center.

No.	TTRS dataset [1]
L_1	Number of incoming calls of video relay service on mobile phones
L_2	Number of received calls of video relay service on mobile phones
L_3	Service time of video relay service on mobile phones
L_4	Waiting time of video relay service on mobile phones
L_5	Number of incoming calls of text relay service
L_6	Number of received calls of text relay service
L_7	Service time of text relay service
L_8	Waiting time of text relay service
L_9	Number of incoming calls of video relay service on kiosks and video phones
L_{10}	Number of received calls of video relay service on kiosks and video phones
L_{11}	Service time of video relay service on kiosks and video phones
L_{12}	Waiting time of video relay service on kiosks and video phones

Service time T_s is defined as a time interval starting from a call agent picking up a call to when both a call agent and a customer release a channel. Waiting time T_w is a time interval starting from when a call from a customer enters a call center channel to when a call agent picks up a call. The waiting time consists of three-time intervals. The first time interval (t_{w1}) is the time that a customer spends calling a call center. The second time interval (t_{w2}) is when a customer has to wait in a queue before a call triggers a ringing notification. Finally, the third time interval (t_{w3}) is when a ringing notification appears to a call agent to the time a call agent picks up a call. As a result, waiting time can be defined by

$$T_w = t_{w1} + t_{w2} + t_{w3}. \tag{1}$$

Elements of a call center dataset consist of a number of incoming calls, a number of receiving calls, each call's service time, and each call's waiting time. Suppose there are k services in a call center. Then, we will have a total of $4k$ datasets. For example, TTRS has three real-time services: video relay service on mobile phones, text relay service, and video relay service on kiosks and video phones. Therefore, TTRS call center has 12 datasets as shown in Table 1.

To obtain meaningful data, we remove false and unintended calls by discarding the calls with the waiting time T_w of lesser than eight seconds and the calls with no responses

TABLE 2. An Example of the data from the TTRS video service.

Date/Time	L_1	L_2	L_3	L_4
01/01/2020, 08:01-08.30	1	1	51	0
01/01/2020, 08:31-09.00	3	3	911	0
01/01/2020, 09:01-09.30	4	4	342	0
01/01/2020, 9:31-10.00	8	8	1056	0
01/01/2020, 10:01-10.30	6	6	1468	0
01/01/2020, 10:31-11.00	9	7	1124	2
01/01/2020, 11:01-11.30	5	5	297	0
01/01/2020, 11:31-12.00	18	16	3474	2

from callers. Then, the data are grouped into chunks of the 30-minute time interval. By dividing a whole day to be several 30 minutes time intervals, we obtain 48 intervals. However, if a dataset contains information for n_d days, the total number of time intervals is calculated by

$$T_{nd} = n_d \times 48. \tag{2}$$

Table 2 shows an example of collected data from the TTRS video service that are grouped into 30-minute time interval chunks.

In general, call center traffic affects by several factors such as weather, time interval, and holidays. For example, the number of incoming calls of TTRS significantly increases during rainy days, breaking times, and holidays. Hence, we incorporate a set of time-related features as shown in Table 3 to characterize call center traffic and call center's collected data.

TABLE 3. Descriptors for predicting call center traffic.

Var	Description	Min	Max
F_1	Day	1	31
F_2	Year	2018	2021
F_3	Is Monday	0	1
F_4	Is Tuesday	0	1
F_5	Is Wednesday	0	1
F_6	Is Thursday	0	1
F_7	Is Friday	0	1
F_8	Is Saturday	0	1
F_9	Is Sunday	0	1
F_{10}	Is Weekday	0	1
F_{11}	Is Weekend	0	1
F_{12}	Is January	0	1
F_{13}	Is February	0	1
F_{14}	Is March	0	1
F_{15}	Is April	0	1
F_{16}	Is May	0	1
F_{17}	Is June	0	1
F_{18}	Is July	0	1
F_{19}	Is August	0	1
F_{20}	Is September	0	1
F_{21}	Is October	0	1
F_{22}	Is November	0	1
F_{23}	Is December	0	1
F_{24}	Is First workday of Month	0	1
F_{25}	Is Middle workday of Month	0	1
F_{26}	Is Last workday of Month	0	1
F_{27}	Is Summer	0	1
F_{28}	Is Rainy	0	1
F_{29}	Is Winter	0	1
F_{30}	Is Public Holiday	0	1
F_{31}	Is Before a Public Holiday	0	1
F_{32}	Is Promote	0	1
F_{33}	1st of 120 minutes of time interval	0	1
F_{34}	2nd of 120 minutes of time interval	0	1
F_{35}	3rd of 120 minutes of time interval	0	1
F_{36}	4th of 120 minutes of time interval	0	1
F_{37}	5th of 120 minutes of time interval	0	1
F_{38}	6th of 120 minutes of time interval	0	1
F_{39}	7th of 120 minutes of time interval	0	1
F_{40}	8th of 120 minutes of time interval	0	1
F_{41}	9th of 120 minutes of time interval	0	1
F_{42}	10th of 120 minutes of time interval	0	1
F_{43}	11th of 120 minutes of time interval	0	1
F_{44}	12th of 120 minutes of time interval	0	1

In this paper, there are 44 time-related features. The first 32 features characterize call center's operating day. The last 12 features (from the 33th feature to 44th feature) represent a service-time interval of the length of 120 minutes within a day. These time-related features F_i, where $1 \leq i \leq 44$, and traffic parameters, L_i, where $1 \leq i \leq 12$ of each call center service are used as inputs of our proposed traffic forecasting, which be described in the following section.

Collected call center data normally possess various statistical properties. Different data may differ in data types, data formats, and data scales. We normalize different data to be on the same scale. The normalization will shorten a training period since the lost function will be close to being symmetric, make easier to optimize. To normalize data, the min-max normalization is utilized, which can be expressed as

$$z_n = \frac{z - z_{min}}{z_{max} - z_{min}}, \tag{3}$$

where z is a data sample, z_n is the normalized sample of z, z_{max} is the data sample with the maximum value, and z_{min} is the data sample with the minimum value. These normalized data will be used as inputs of a traffic forecasting module.

B. CALL CENTER TRAFFIC FORECASTING MODULE

We use a deep neural network [38] to predict call center traffic since it can possibly derive non-linear relation among several call center factors. Our proposed neural network is based on a combination of Long Short-Term Memory (LSTM) networks and Fully Connected (FC) networks, as illustrated in Fig. 2. The deep neural network contains two layers of LSTM and six layers of FCs. Since traffic data of a call center correlates with themselves in a temporal domain and with other call center descriptors, we employ the LSTM networks in the first two layers to extract temporal features from call center data. Then, correlations among features are computed from the FCs. Finally, the output from the DNN is the call center traffic statistics such as the number of incoming calls, the number of received calls, the service time, and the waiting time.
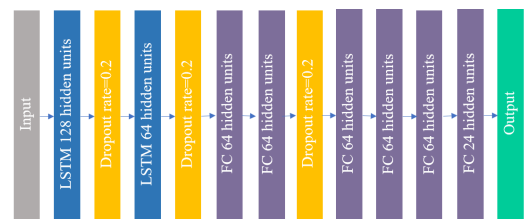
FIGURE 2. The proposed traffic forecasting network.

The pre-processed data are fed to the LSTM network with 128 hidden units. A Rectified Linear Unit (ReLU) is used as an activation function for all network layers. The activations from layer one are fed as inputs to layer two, which is the LSTM network with 64 hidden units. The activations of layer two are fed to be inputs of layer three, which is a fully connected layer with 64 hidden units. The activations from

layer three are fed to layer four, which is a fully connected network with 64 hidden units. The activation of layer four is fed to the fully connected network of layer five with 64 hidden units. Layer six and seven possess the same structure as layer five. The dropout rate is set to be the same value as layer one during the training process. The activations of layer seven are fed to the fully connected network with 24 hidden units of layer eight, and the prediction output is the product from the final layer. The dropout rate during the training process is set to be 0.2 to avoid over-fitting. These predicted call center statistics will be used in an adaptive workforce management algorithm, which will be described in the next section.

III. DYNAMIC WORKFORCE MANAGEMENT WITH Q-LEARNING ALGORITHM

This section addresses a new workforce management method with reinforcement learning (RL). RL offers several advantages in the workforce allocation problem as follows.

- The RL can handle a complex task involving several variables, including incoming traffic, SLA, and starting time of each service interval.
- The customer behaviors dynamically change over time. The conventional method may not learn from changing data, resulting in the human resource policy that fails to achieve better call center services. For example, the RL-based human resource management can outperform the supervised learning-based human resource management under dynamic environments because the training data may not reflect the changing customer behaviors.

The following subsections will present the formulation of the workforce management problem as Markov decision process and demonstrate the use of Q-learning algorithm to search for the best solution of human resource allocation.

A. WORKFORCE MANAGEMENT AS MARKOV DECISION PROCESS

We formulate our workforce management problem using a Markov Decision Process (MDS). The MDS consists of a set of environment and agent states, a set of actions of the RL agent, the transition probability from state S to state S', and the immediate reward from state S to state S' with action a. The MDS's objective in our problem is to learn the optimal or nearly optimal workforce management policy that maximizes the reward function. The reward function directly reflects call center service qualities. The service qualities are measured via Service Level Agreements (SLA), such as a dropped call rate or waiting time.

An RL agent interacts with a dynamic call traffic environment in discrete time steps. At each time step t, an agent possesses state S_t and reward r_t. An agent chooses an action a_t from all possible actions. This action is sent to the environment before moving to the next state S_{t+1} and the reward r_{t+1} corresponding with the transition (S_t, a_t, S_{t+1}) is computed. An agent observes and predicts the current call traffic environment state in our formulating problem via our

proposed deep learning network. Our formulated RL action, a_t, is defined as an ordered pair of the starting time of each working shift and a number of call center agents working in that shift. For example, if each consecutive starting time is 30 minutes apart, within one day, there are a total of 48 possible starting times (i.e., 00.00, 00.30,.., 23.30). The number of call center agents ranges from one to the whole number of call agents in a call center. For instance, if a call center possesses 24 call agents, call agents range from one to 24. Let l_t be the starting time, and g_t is a corresponding number of call agents. The ordered pair (l_t, g_t) can represent an action of RL. A number of possible actions is dependent on the possible number of starting times and a number of call center agents. In our example, there are possible 48 possible starting times and 24 call agents. Hence, there are a total of $48 \times 24 = 1152$ actions.

Define agent state t, $S_t(\gamma, \beta)$, characterized by working shift γ and service type β. For example, suppose that we have five working shifts and three services. In this case, γ can be either working shift $R_1, R_2, R_3, R_4,$ or R_5 and β is either service $C_1, C_2,$ or $C3$. There are a total of 15 states, as shown in Fig.3. The dwell time of the specific state is equal to the length of the corresponding shift. The state only changes within the same service type. We preset the dwell time of each state based on its service time and its service type. Hence, it is certain that state $S_t(R_t, C_k)$ will move to state $S_{t+1}(R_{t+1}, C_k)$. For instance, in Fig.3, we will change from state $S_1(R_1, C_1)$ to state $S_2(R_2, C_1)$ after staying in state $S_1(R_1, C_1)$ for its state dwell time $\tau_1(C_1)$. Since the starting time of each state or shift is dictated by the starting time from the selected action, it is possible that consecutive working shifts may overlap. In this case, the number of agents working during the overlapping shifts is equal to working call agents summation from consecutive states. The number of working call agents for each service can be expressed as

$$\sum_{k \in K} g_k w_k, \qquad (4)$$

where K denotes the state S_t within the same service type k that represents (R_t, C_k) at each time step t. w_k is set to be one for a consecutive state and zero for a non-consecutive state. Fig.3 shows an example of overlapping shifts when state $S_1(R_1, C_1)$ overlaps with state $S_2(R_2, C_1)$. The number of working call agents during these overlapping states is equal to $g_1 + g_2$. Notice that when state $S_2(R_2, C_1)$ does not overlap with state $S_1(R_1, C_1)$, a number of working call agents is equal to g_2. Our proposed learning agent learns a policy

$$\pi(S_{t+1}(R_{t+1}, C_k)) = a_{t+1}(l_t, g_t), \qquad (5)$$

which maximizes the accumulative reward. The immediate reward is positive and negative when the action gives call center performance above and below the preset SLA.

B. REWARD FUNCTION AND SERVICE LEVEL AGREEMENT

To apply the immediate reward to an RL agent, it is needed to define the SLA of a call center. We measure the SLA of
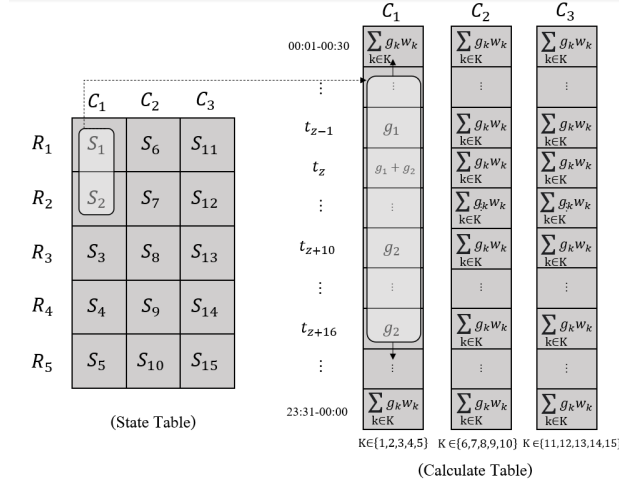
FIGURE 3. An example of Q-learning states.

a call center based on the average speed of answer and the probability that the actual speed of answer is less than or equal to the required average speed of answer. To compute the SLA, define A as a traffic intensity [19] computed as

$$A = \lambda \times H = \frac{\lambda}{\mu}, \tag{6}$$

where λ is an arrival rate of calls, H is the average service time per call, and μ is a service rate, which is equal to $\frac{1}{H}$. Suppose that there are n_a agents. A load per agent [19] can be calculated as

$$\rho = \frac{\lambda}{\mu \times n_a}. \tag{7}$$

Define a ratio between a call abandonment probability and the average waiting time, $\bar{T}_w$, as

$$\theta = \frac{P_{ab}}{\bar{T}_w}. \tag{8}$$

The Erlang-A model [5] is used to compute the performance metrics of a call center. We first define parameters J, ε, and K via [20]

$$J = \frac{e^{\frac{\lambda}{\theta}}}{\theta} \cdot \left(\frac{\theta}{\lambda}\right)^{\frac{n}{H \cdot \theta}} \cdot \gamma\left(\frac{n}{H \cdot \theta}, \frac{\lambda}{\theta}\right), \tag{9}$$

$$\varepsilon = \frac{\sum_{j=0}^{n-1}\left(\frac{1}{j!}\right)\left(\frac{\lambda}{\mu}\right)^j}{\frac{1}{(n-1)!}\left(\frac{\lambda}{\mu}\right)^{n-1}}, \tag{10}$$

and

$$K = \frac{\left(\frac{n}{H \cdot \theta}\right) \cdot e^{\frac{\lambda}{\theta}} \cdot \gamma\left(\frac{n}{H \cdot \theta}, \frac{\lambda}{\theta}\right)}{\frac{\lambda}{\mu}^{\frac{n}{H \cdot \theta}}}. \tag{11}$$

The parameters J, ε and K can be deployed to calculate the probability of waiting $P\{T_w > 0\}$ and the probability of abandonment P_{ab} as

$$P\{T_w > 0\} = \frac{\lambda J}{\varepsilon + \lambda J} \cdot (1 - \theta), \tag{12}$$

and

$$P_{ab} = P\{T_w > 0\}\left(\frac{1}{\rho K} + 1 - \frac{1}{\rho}\right). \tag{13}$$

The above equations lead us to performance metrics of a call center via the Erlang-A model as

$$T_{asa} = \frac{P_{ab}}{\theta}, \tag{14}$$

where T_{asa} is the average speed of answer. A probability that the speed of answer is less than or equal to the average speed of answer can be expressed as

$$P\{t \leq T_{asa}\} = 1 - \left[\frac{\lambda J(t)}{\varepsilon + \lambda J(t)} \cdot (1 - \theta)\right]. \tag{15}$$

In general, any call center sets its own SLA. In the case of the TTRS call center, it sets the required average speed of answer for video relay service on mobile phones, text relay service, and video relay service on kiosks and video phones to be 15, 12, and 15 seconds, respectively. Required $P\{t \leq T_{asa}\}$ of all three services are set to be 0.85. With predicted call center traffics from our proposed deep learning model, we can estimate T_{asa} and $P\{t \leq T_{asa}\}$ relied on the Erlang-A model and the human resource strategy from the RL. If an RL action satisfies the preset SLA, the immediate reward will be granted to the RL. In contrast, the negative reward is applied to the RL if its action can not meet the SLA.

C. Q-LEARNING ALGORITHM
We employ the Q-learning algorithm [21] to search for the optimal workforce management policy. The Q-learning algorithm computes a quality of a state-action combination via a mapping

$$Q : S \times A \rightarrow R, \tag{16}$$

where $S \times A$ is a Cartesian product between a set of states and a set of actions. R is a set of a real number. Before learning starts, the mapping Q is initialized to a random number. When an RL agent chooses an action a_t, the Q-learning algorithm observes a reward r_t. Then, the state changes from state S_t to state S_{t+1}. The Q-value is updated via [22], [23]

$$Q^{new}(S_t, a_t) = Q(S_t, a_t) + \alpha(r_t + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, a_t)), \tag{17}$$

where r_t is a reward when moving from state S_t to state S_{t+1}. γ is a discount factor and α is a learning rate. The algorithm will end when S_t is the terminal state, where the Q value is no longer updated. We describe the use of the Q-learning algorithm in the following sections.

D. Q-TABLE INITIALIZATION
To perform the Q-learning algorithm, a Q-table must first be initialized. Rows and columns of the Q-table correspond to states and actions. For instance, if we have 15 states due to five working shifts and three services, and 1152 actions from all 48 starting times and 24 call agents, the Q-table can be shown in Table 4.

TABLE 4. Q-table initialization.

	a_1	a_2	a_3	$\cdots$	a_{1152}
S_1	$Q^{\Delta t}_{1,1}$	$Q^{\Delta t}_{1,2}$	$Q^{\Delta t}_{1,3}$	$\cdots$	$Q^{\Delta t}_{1,1152}$
S_2	$Q^{\Delta t}_{2,1}$	$Q^{\Delta t}_{2,2}$	$Q^{\Delta t}_{2,3}$	$\cdots$	$Q^{\Delta t}_{2,1152}$
S_3	$Q^{\Delta t}_{3,1}$	$Q^{\Delta t}_{3,2}$	$Q^{\Delta t}_{3,3}$	$\cdots$	$Q^{\Delta t}_{3,1152}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
S_{15}	$Q^{\Delta t}_{15,1}$	$Q^{\Delta t}_{15,2}$	$Q^{\Delta t}_{15,3}$	$\cdots$	$Q^{\Delta t}_{15,1152}$

The Q-values in the table are set to all zeros. Each shift's starting time and a call agent number are set to be random and one, respectively. We also initialize important parameters, including a discount factor (γ) and a learning rate (α). In this paper, we set them to 0.8 and 0.0001, respectively.

E. ACTION SELECTION

The epsilon greedy algorithm [24] is used to search for the optimal policy in the Q-learning algorithm. The RL agent selects the action corresponding to the highest reward with probability $1 - \epsilon$. Otherwise, the RL agent will obtain the action from the rest with probability ϵ. Define the initial value of ϵ as ϵ_s, the final value of ϵ as ϵ_s, and the decay rate of ϵ as ϵ_d. The value of ϵ is updated for the next time step Δt as

$$\epsilon = \epsilon_e + \frac{(\epsilon_s + \epsilon_e)}{\epsilon^{\epsilon_e \cdot \Delta t}}. \tag{18}$$

The greedy algorithm to search for the optimal policy can be described step-by-step as follows.

- Step 1: Set $\Delta t = 0$. Initialize a number of call agents to be one and set the starting times randomly for all actions.
- Step 2: For each iteration of Δt and state S_k, compute ϵ from 18 and select randomly ζ between zero to one with a uniform distribution. Select a policy via

$$\pi(S_k) = \begin{cases} a_r(l_r, g_r), & \text{if } \zeta < \epsilon, \\ \max_{\forall a_i} Q^{\Delta t}_{k,i}(S^{\Delta t}_{k+1}, a^{\Delta t}_i), & \text{otherwise.} \end{cases} \tag{19}$$

- Repeat Step 2 for T_f iterations and return the optimal actions of all states.

The greedy algorithm can be written in an algorithm form in Algorithm 1. Note that even though the selected actions are optimal in the Q-learning sense, they may not meet some constraints imposed by a call center. In this case, the selected actions will not be passed to the RL agents. Instead, the greedy algorithm continues on the next step until finding the optimal policy does not violate the constraints. Examples of restrictions imposed by a call center are the total available of call agents, the minimum number of call agents allocated to different services, and starting time of each shift. It is possible that if there are too many constraints, the greedy algorithm may not converge. In this case, we may need to relax some constraints.

Algorithm 1 The Epsilon-Greedy Action Selection

Initialize the time step to be zero, the number of call agents to be one and randomly the starting times for all actions
for Each iteration of Δt and S_k **do**
    Compute ϵ as in (Eq. 18)
    select randomly r between zero to one with a uniform distribution
$$\pi(S_k) = \begin{cases} a_r(l_r, g_r) & r = \epsilon \\ \max_{\forall a_i} Q^{\Delta t}_{k,i}(S^{\Delta t}_{k+1}, a^{\Delta t}_i) & r = 1 - \epsilon \end{cases}$$
end for
return π, a

TABLE 5. Hyperparameters used in deep learning prediction module and reinforcement learning.

Model	Parameter	Value
Deep learning	Batch size	32
	Activation	ReLU
	Optimizer	ADAM
	Dropout	0.2
	Learning rate	0.0001
	Epochs	10,000
	Loss function	Mean Square Error
Reinforcement Learning	Episode	100,000
	Discount factor	0.8
	Learning rate	0.0001

IV. EXPERIMENTAL RESULTS

This section evaluates the proposed adaptive human resource management algorithm. We deploy the actual call center data from TTRS ranging from 1 January 2018 to 31 January 2021 in our experiments [1]. The TTRS data are based on 24 call agents and three services containing TTRS video relay service on mobile phones, TTRS text relay service, and TTRS video relay service on kiosks and video phones. Call center data from 1 January 2018 to 31 December 2020 are used as the training set, whereas the rest are used as a test set. Table 5 shows the hyperparameters utilized in both deep learning prediction module and reinforcement learning. The hidden node number of each fully connected layer is 64, except the last fully connected layer that has 24 nodes as shown in .

A. CALL CENTER TRAFFIC PREDICTION

We first assess the performance of the proposed traffic prediction algorithm with DNN in this section. We compare the experimental results of our algorithm with previous works including Auto-Regressive Integrated Moving Average (ARIMA) [25], LSTM [26], Exponential Smoothing (ES) [27], and LSTM Deep Autoencoder (LSTM+DAE) [28]. We predict various call center traffic parameters, which are a number of incoming calls, a number of received calls, service time, and waiting time. The Mean Absolute Error (MAE) between the predicted and the actual traffic parameters are used as a metric for comparison among

TABLE 6. A comparison of the MAEs of the TTRS video relay service between the proposed method and the existing methods.

TTRS video relay service on mobile				
Method	Incoming Calls	Received Calls	Service Time	Waiting Time
ARIMA [25]	6.531	6.434	6.564	6.442
LSTM [26]	5.318	5.467	5.497	5.538
ES [27]	2.004	2.385	2.473	2.144
LSTM +DAE [28]	0.953	1.276	1.193	1.531
Proposed Method	**0.832**	**0.966**	**0.991**	**0.981**

TABLE 7. A comparison of the MAEs of TTRS text relay service between the proposed method and the existing methods.

TTRS text relay service				
Method	Incoming Calls	Received Calls	Service Time	Waiting Time
ARIMA [25]	6.434	6.442	6.561	6.432
LSTM [26]	5.317	5.511	5.486	5.476
ES [27]	2.311	2.298	2.484	2.204
LSTM +DAE [28]	0.954	**0.915**	1.563	1.112
Proposed Method	**0.817**	1.386	**1.464**	**0.962**

TABLE 8. A comparison of the MAEs of TTRS video relay service in kiosks and video phones between the proposed method and the existing methods.

TTRS video relay service with kiosks and video phones				
Method	Incoming Calls	Received Calls	Service Time	Waiting Time
ARIMA [25]	6.528	6.442	6.621	6.478
LSTM [26]	5.412	5.512	5.445	5.531
ES [27]	2.255	2.424	2.633	2.124
LSTM +DAE [28]	1.133	**0.843**	1.168	1.275
Proposed Method	**0.927**	0.910	**0.938**	1.138

different algorithms. The MAE can be expressed as

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |p_i - \hat{p}_i|, \qquad (20)$$

where p_i and $\hat{p}_i$ are the ith actual and predicted traffic values. N is the total number of traffic data.

Tables 6,7, and 8 compare the results obtained from three services of TTRS, TTRS video relay service on mobile phones, TTRS text relay service, and TTRS video relay service on kiosks and video phones, respectively. The MAE results indicate that the proposed traffic forecasting based on deep learning can surpass all previous predictive techniques and can give higher predictive accuracy. The superior prediction accuracy of our algorithm is based on the exploitation of LSTM and fully connected layers. The LSTM layers are utilized to learn temporal dependency among time-series call center traffic. Then, six fully connected layers are employed to derive non-linear mapping between LSTM outputs and the forecasted traffic. As a result, it outperforms the scheme that

uses LSTM alone [26], and LSTM+DAE [28], which uses only three layers of fully connected layers.

Fig.4 compares the actual call center metrics with their predicted values from deep learning. We can observe that call center metrics are dynamically changing along with time intervals. Each time interval corresponds to a 30-minute call center service. From our experimental results, the proposed traffic forecasting method can track and predict call center metrics with high accuracy due to the non-linear properties of deep learning. The high accuracy of traffic prediction will enhance the performance of human resource management evaluated in the next section.

B. CALL CENTER WORKFORCE MANAGEMENT WITH REINFORCEMENT LEARNING

We verify the proposed human resource management with reinforcement learning in this section. We divide working hours within a day to be 48 intervals of 30 minutes. We set the total number of call agents to be 24 referenced from TTRS. Therefore, we have a total of $24 \times 48 = 1152$ actions in the Q-learning algorithm. Each action represents the starting time and the number of call agents allocated to various services. For each episode of the Q-learning algorithm, we average Q-values from all actions. First, we study the effects of different reward functions [29]–[31] on the convergence of the Q-learning algorithm. We map the value y of the SLA to a new number via

$$x = y \times \frac{x_{max} - x_{min}}{y_{max} - y_{min}} + y_{min} \times (1 + \frac{x_{max} - x_{min}}{y_{max} - y_{min}}), \qquad (21)$$

where x_{max} and x_{min} are the upper and lower limits of the x value, and y_{max} and y_{min} are the upper and lower limits of the SLA. The unit of the SLA is the percentage that call agents can provide the service within 15 seconds. To be specific, in this paper, $y_{min} = 0$ and $y_{max} = 100$. Based on a new value of x, we compute a reward via the following functions [32]− [36].

- Sigmoid [32]:

$$r(x) = \frac{1}{1 + e^{-x}}. \qquad (22)$$

- SiLU [33]:

$$r(x) = \frac{x}{1 + e^{-x}}. \qquad (23)$$

- tanh [34]

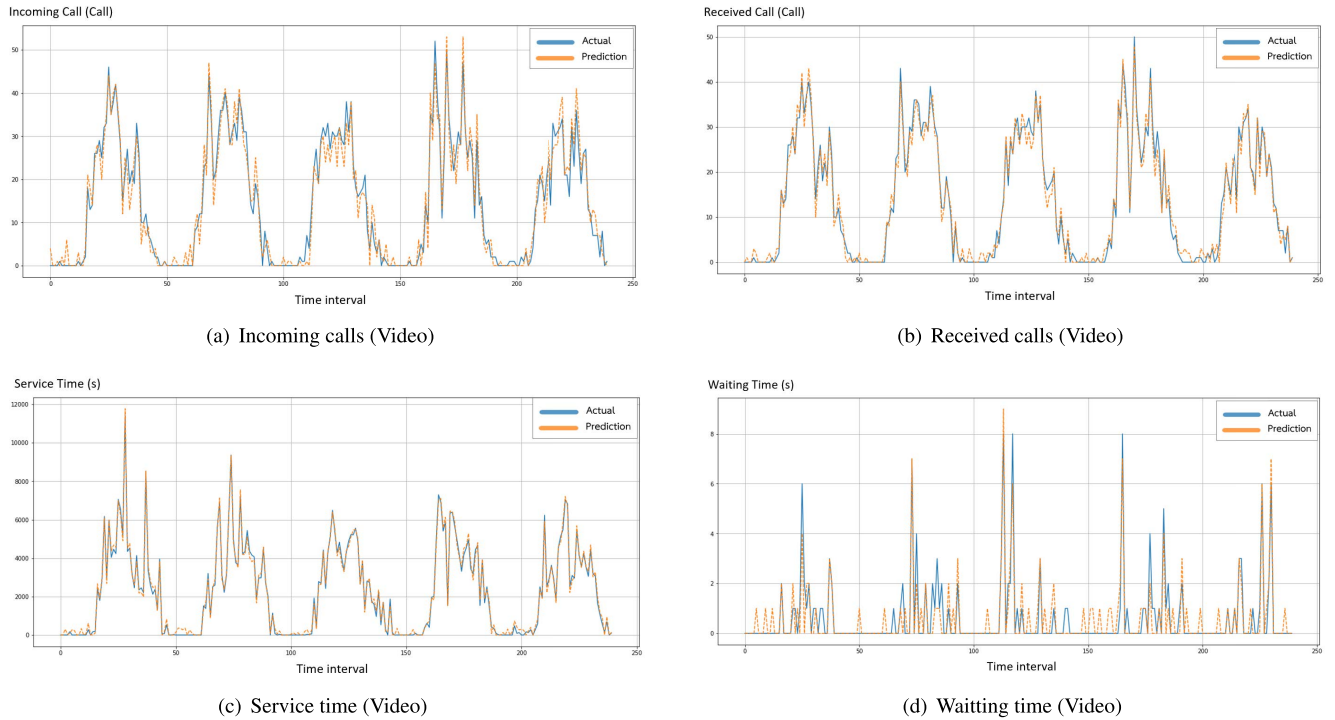$$r(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \qquad (24)$$

- Softplus [35]
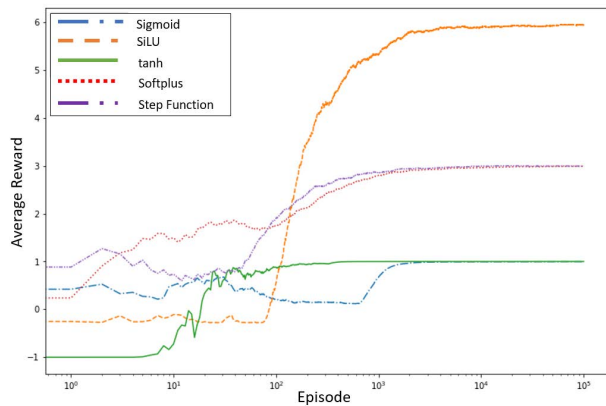
$$r(x) = ln(1 + e^x). \qquad (25)$$

- Step function [36]

$$r(y) = \begin{cases} 30 & y \geq t_{SLA}, \\ -30 & y < t_{SLA}, \end{cases} \qquad (26)$$

(a) Incoming calls (Video)

(b) Received calls (Video)

(c) Service time (Video)
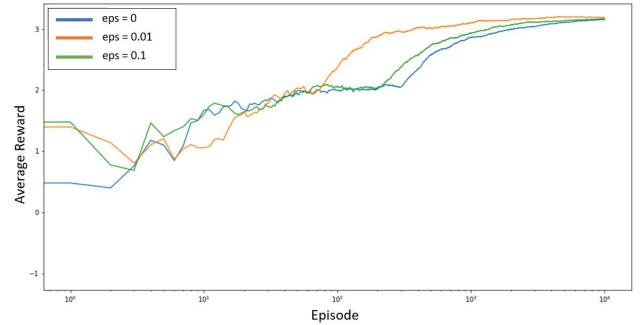

(d) Waitting time (Video)

FIGURE 4. Comparison between the actual and predicted call center traffic values using the proposed network.

FIGURE 5. A comparison of average rewards from different reward functions.

FIGURE 6. A comparison of average rewards from different ϵ values.

where t_{SLA} is the preset threshold of the SLA equaling to 85.

If the Q-learning algorithm gives the human resource violating the preset conditions of a call center, the reward will be set to be -30. The constraints can be customized to match the requirements of different call centers. Fig.5 shows the average reward along with the episode from 01/01/2021 to 05/01/2021. We average over five working shifts and three services. We can see that the tanh function gives the fastest convergence. In contrast, the sigmoid reward function has the slowest convergence rate. Fig.6 shows the effects from different ϵ values of the greedy epsilon algorithm to the average reward. We can see that when we set the value of

ϵ to be 0.01, the searching algorithm for the best strategy converges fastest. This implies that the tanh reward function should be employed in practice, and the value of ϵ should be set to 0.01 when we deploy the Q-learning algorithm for human resource management.

We next evaluate the performance of our proposed method for adaptive workforce management using RL. The results of workforce management are based on the call center traffics from our proposed DNN. We compare the results of our proposed method to the outcomes of workforce management by experienced human supervisors and the Decision Support System (DSS) [7] as shown in Table 9. It is obvious to see that these three human resource management schemes produce different human resource management solutions. For example, on January 1, 2021, a human supervisor allocated three call agents for video relay service on mobile phones,

TABLE 9. A comparison of the predicted number of minimum required agents for each service between the proposed method and the existing methods. In this table, the notations of Video, Text, and VP represent video relay service on mobile phone, text relay service, and video relay services on kiosks and video phones.

		Proposed method				Experienced human supervisor				Decision Support System [7]			
		Time	Video	Text	VP	Time	Video	Text	VP	Time	Video	Text	VP
01/01/2021		07.01-15.30	2	1	1	07.01-15.30	3	1	1	07.01-15.30	1	1	1
		08.01-16.30	2	2	1	08.31-17.00	4	2	1	09.01-17.30	2	2	2
		12.31-21.00	4	2	1	12.31-21.00	3	1	1	12.01-20.30	2	3	1
		14.01-22.30	3	1	1	14.31-23.00	2	1	1	20.01-04.30	2	2	1
		22.31-07.00	1	1	1	22.31-07.00	1	1	1	22.31-07.00	1	3	1
02/01/2021		07.01-15.30	2	1	1	07.01-15.30	1	1	1	07.01-15.30	1	1	1
		08.01-16.30	2	2	1	08.31-17.00	4	2	1	09.01-17.30	2	2	2
		10.01-18.30	3	3	1	12.31-21.00	3	1	1	12.01-20.30	3	1	2
		14.31-22.30	3	1	1	14.31-23.00	2	1	1	15.01-23.30	2	2	1
		22.31-07.00	1	1	1	22.31-07.00	1	1	1	22.31-07.00	1	2	1
03/01/2021		07.01-15.30	1	1	1	07.01-15.30	1	1	1	07.01-15.30	1	1	1
		08.31-17.00	2	3	1	08.31-17.00	4	2	1	08.01-16.30	2	2	2
		11.31-20.00	3	3	1	12.31-21.00	3	1	1	11.01-19.30	2	2	2
		15.31-24.00	3	1	1	14.31-23.00	2	1	1	14.31-23.00	2	2	2
		22.31-07.00	1	1	1	22.31-07.00	1	1	1	22.31-07.00	1	1	1
04/01/2021		07.01-15.30	1	1	1	07.01-15.30	1	1	1	07.01-15.30	1	1	1
		08.31-17.00	2	3	1	08.31-17.00	4	2	1	08.01-16.30	2	2	2
		11.31-20.00	3	3	1	12.31-21.00	3	1	1	11.01-19.30	2	2	2
		15.31-24.00	3	1	1	14.31-23.00	2	1	1	14.31-23.00	2	2	2
		22.31-07.00	1	1	1	22.31-07.00	1	1	1	22.31-07.00	1	1	1
05/01/2021		07.01-15.30	1	1	1	07.01-15.30	1	1	1	07.01-15.30	1	1	1
		08.31-17.00	2	3	1	08.31-17.00	4	2	1	08.01-16.30	2	2	2
		11.31-20.00	3	3	1	12.31-21.00	3	1	1	11.01-19.30	2	2	2
		15.31-24.00	3	1	1	14.31-23.00	2	1	1	14.31-23.00	2	2	2
		22.31-07.00	1	1	1	22.31-07.00	1	1	1	22.31-07.00	1	1	1

TABLE 10. Comparison of the average performances of workforce allocation methods within one month. In this table, the notations of Video, Text, and VP represent video relay service on mobile phone, text relay service, and video relay service on kiosks and video phones.

Result	Proposed method			Experienced Human Supervisor			Decision Support System [7]		
	Video	Text	VP	Video	Text	VP	Video	Text	VP
SLA (percent)	**98.456**	**99.921**	**98.936**	97.666	99.51	97.143	97.138	95.906	95.466
ASA (seconds)	**8.57**	**0.06**	**6.53**	47.61	1.73	25.70	107.30	27.36	19.74

whereas the DSS and our proposed algorithm suggested one and two agents, respectively. To assess how these solutions affect the call center's quality of services, we measure the percentages that call agents can provide services within the desired thresholds. Specifically, we adopt the service level agreement (SLA) of TTRS that call agents must provide

TABLE 11. Comparison of the percentage of the SLA and ASA failures from the workforce allocation methods.

Methods	SLA_f (percent)
Experienced Human Supervisor	9.00
Decision Support System [7]	23.33
Proposed method	**4.93**

video relay service on mobile phones, text relay service, and video relay service on kiosks and video phones within 15, 12, and 15 seconds after a customer starts making a call. Moreover, the average speed of answer (ASA) is also utilized as our comparison metric. Table 10 shows the average performance within one month. The proposed method surpasses other methods in both SLA and ASA for all services. The results indicate that when our proposed algorithm is utilized, customers spend less waiting time, leading to better customer satisfaction. In contrast, human resource management solutions from human supervisors and the DSS cause customers to wait for services longer and not maintain the required SLA in some services. For example, in video relay service on mobile phones, the ASAs of video relay service from human supervisors and DSS are equal to 47.61 and 107.3 seconds, exceeding the preset ASA at 15 seconds. In addition, the percentages that the proposed method can maintain the SLA overcomes human supervisors and the DSS.

In addition, we define a metric measuring a frequency that the SLA is not met based on various human resource management schemes. Define L_t as the total number of the 30-minute service interval. Each operating day contains $24 \times 2 = 48$ intervals. Therefore, each month will contain 1440 service intervals. We categorize this interval as an SLA-failure interval if the SLA is not met in a service interval regardless of the SLA failing duration. Define L_f as the total number of an SLA-failure interval. The SLA failure rate can be defined as

$$SLA_f = \frac{L_f}{L_t} \times 100. \qquad (27)$$

For example, if the total number of SLA-failure intervals is equal to 144 intervals within one month, the SLA failure rate will be equal to 10 percent. We use this metric to assess the SLA-failure rates of the proposed algorithm, human supervisors, the DSS. Table 11 compares the SLA-failure rate among the proposed algorithm, human supervisors, and the DSS. We can see that the proposed method provides less SLA-failure frequency than other methods. This is based on the fact that the human resource allocation from our proposed algorithm most likely corresponds to the actual call volume. Hence, the SLA failure rate has decreased since we have sufficient call agents to pick up incoming calls in every service interval. This also implies that the proposed algorithm can better maintain the SLA along with service times.

Finally, we investigate the influences of traffic prediction algorithms on the proposed human resource management method. We adopt the above SLA failure rate as an evaluation metric. Table 12 shows the SLA failure rates of different traffic prediction algorithms. When we incorporate the deep

TABLE 12. Comparison of the percentage of the SLA and ASA failures from the traffic prediction methods between the proposed DNN-based method and the existing methods.

Methods	SLA_f (percent)
ARIMA [25] + RL	31.46
LSTM [26] + RL	30.00
ES [27] + RL	21.60
LSTM +DAE [28] + RL	10.07
DNN + RL	**5.69**

neural network with the RL, the SLA failure rate drops to 5.69 percent, which is much lower than those of other algorithms such as ARIMA, LSTM, and ES. The lower SLA failure rate comes from the fact that traffic prediction based on deep learning gives accurately predicted parameters. As a result, the RL can precisely allocate human resources to all services.

V. CONCLUSION

This paper presented a novel learning-based method for adaptive call center workforce management. The proposed system was designed to have two stages. The first stage was the call center traffic prediction, and the second stage was the automatic workforce management with reinforcement learning. The call center traffic prediction utilized a deep neural network containing the LSTM and the fully connected networks to predict call center traffic parameters. Because traffic data of a call center possessed high temporal correlation, we applied the LSTM to extract temporal features and derived correlation among features with the fully connected network. These predicted traffic parameters were fed to the Erlang-A formulation to compute the expected service level agreement. Under changing traffic statistics, the reinforcement learning with the Q-learning algorithm automatically found the best workforce allocation solution under some constraints, such as the total available call agents. From the experimental results with actual data, the proposed human resource management scheme with the deep learning-based traffic prediction surpassed other methods regarding the desired SLA and ASA and the SLA failure rate. Without human intervention, the proposed human resource management method could maintain the quality of service, provided high customer satisfaction, and offloaded human supervisors from tedious workforce management duties. The proposed framework can be utilized to replace the traditional human resource allocation based on the trunking theory in all kinds of call centers.

REFERENCES

[1] *Thai Telecommunication Relay Service*. Accessed: Nov. 2, 2021. [Online]. Available: https://www.ttrs.or.th/

[2] M. A. Huselid, "The science and practice of workforce analytics: Introduction to the HRM special issue," *Human Resource Manage.*, vol. 57, no. 3, pp. 679–684, May 2018.

[3] B. Dallago, "The budget constraint in the governance of organizations," *Corporate Ownership Control*, vol. 5, no. 4, pp. 219–232, Jun. 2008, doi: 10.2139/ssrn.1031888.

[4] I. Angus, "An introduction to Erlang B and Erlang C," *Telemanagement*, to be published.

[5] M. Avishai and S. Zeltyn, "Service engineering in action: The palm/Erlang-a queue, with applications to call centers," in *Advances in Services Innovations*, Berlin, Germany: Springer, 2007, pp. 17–45.

[6] G. Koole and S. Li, "A practice-oriented overview of call center workforce planning," 2021. [Online]. Available: https://arxiv.org/abs/2101.10122

[7] A. Erdem and B. Gedikoglu, "A DSS for shift design and workforce allocation in a call center," in *Proc. Technol. Manage. Global Future (PICMET Conf.)*, Jul. 2006, pp. 1279–1289, doi: 10.1109/PICMET.2006.296696.

[8] A. Tomita, "A managing framework for artificial intelligence to supplement human workforce in a digital economy," in *Proc. Portland Int. Conf. Manage. Eng. Technol. (PICMET)*, Aug. 2019, pp. 1–7, doi: 10.23919/PICMET.2019.8893902.

[9] I. Tewari and M. Pant, "Artificial intelligence reshaping human resource management: A review," in *Proc. IEEE Int. Conf. Advent Trends Multidisciplinary Res. Innov. (ICATMRI)*, Dec. 2020, pp. 1–4, doi: 10.1109/ICATMRI51801.2020.9398420.

[10] R. K. Daya, M. Jeyasurya, M. R. R. Kumar, S. Priyadharshini, and R. G. V. Vyshali, "Machine learning's role in the workplace of the future," in *Proc. Int. Conf. Digitization (ICD)*, Nov. 2019, pp. 60–63, doi: 10.1109/ICD47981.2019.9105676.

[11] G. Metan, I. Sabuncuoglu, and H. Pierreval, "Real time selection of scheduling rules and knowledge extraction via dynamically controlled data mining," *Int. J. Prod. Res.*, vol. 48, no. 23, pp. 6909–6938, Dec. 2010, doi: 10.1080/00207540903307581.

[12] C. Chenqi, "Job scheduling using neural network in environment inspection," M.S. Thesis, Disciplinary Domain Sci. Technol., Math. Comput. Sci., Dept. Inf. Technol., Uppsala Univ., Uppsala, Sweden, 2012

[13] F. Glover and M. Laguna, "Tabu search—Part I," *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, 1989.

[14] M. Knowles, D. Baglee, and S. Wermter, "Reinforcement learning for scheduling of maintenance," in *Research and Development in Intelligent Systems XXVII (SGAI)*, M. T. Bramer, M. Petridis, and A. Hopgood, Eds. London, U.K.: Springer, 2010, pp. 409–422.

[15] S. I. Serengil and A. Ozpinar, "Workforce optimization for bank operation centers: A machine learning approach," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, no. 6, pp. 81–87, Nov. 2017.

[16] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.

[17] W. Chen, Y. Xu, and X. Wu, "Deep reinforcement learning for multi-resource multi-machine job scheduling," 2017, *arXiv:1711.07440*.

[18] B. Waschneck, A. Reichstaller, L. Belzner, and T. Altenmuller, "Optimization of global production scheduling with deep reinforcement learning," *Proc. CIRP*, vol. 72, pp. 1264–1269, Aug. 2018, doi: 10.1016/j.procir.2018.03.212.

[19] K. Nag and M. Helal, "Evaluating Erlang C and Erlang a models for staff optimization: A case study in an airline call center," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*, Singapore, Dec. 2017, pp. 1–5, doi: 10.1109/IEEM.2017.8289839.

[20] M. Avishai and S. Zeltyn, "Call centers with impatient customers: Many-server asymptotics of the $M/M/n+G$ queue," *Queueing Syst.*, vol. 51, pp. 361–402, Dec. 2005, doi: 10.1007/s11134-005-3699-8.

[21] B. Jang, M. Kim, G. Harerimana, and J. W. Kim, "Q-learning algorithms: A comprehensive classification and applications," *IEEE Access*, vol. 7, pp. 133653–133667, 2019, doi: 10.1109/ACCESS.2019.2941229.

[22] H. Ge, Y. Song, C. Wu, J. Ren, and G. Tan, "Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control," *IEEE Access*, vol. 7, pp. 40797–40809, 2019, doi: 10.1109/ACCESS.2019.2907618.

[23] A. Kushwaha, M. Gopal, and B. Singh, "Q-learning based maximum power extraction for wind energy conversion system with variable wind speed," *IEEE Trans. Energy Convers.*, vol. 35, no. 3, pp. 1160–1170, Sep. 2020, doi: 10.1109/TEC.2020.2990937.

[24] H. A. Tran, S. Hoceini, A. Mellouk, J. Perez, and S. Zeadally, "QoE-based server selection for content distribution networks," *IEEE Trans. Comput.*, vol. 63, no. 11, pp. 2803–2815, Nov. 2014, doi: 10.1109/TC.2013.33.

[25] G. Box, G. Jenkins, and G. Reinsel, *Time Series Analysis: Forecasting and Control* (Forecasting and Control Series). Hoboken, NJ, USA: Wiley, 1994.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[27] R. Fried and A. C. George, "Exponential and holt-winters smoothing," in *Proc. Int. Encyclopedia Stat. Sci.*, 2011, pp. 488–490.

[28] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *J. Comput. Commun.*, vol. 6, no. 7, pp. 504–507, Jul. 2006.

[29] Z. Hu, K. Wan, X. Gao, and Y. Zhai, "A dynamic adjusting reward function method for deep reinforcement learning with adjustable parameters," *Math. Problems Eng.*, vol. 2019, Nov. 2019, Art. no. 7619483, doi: 10.1155/2019/7619483.

[30] J. Xie, Z. Shao, Y. Li, Y. Guan, and J. Tan, "Deep reinforcement learning with optimized reward functions for robotic trajectory planning," *IEEE Access*, vol. 7, pp. 105669–105679, 2019, doi: 10.1109/ACCESS.2019.2932257.

[31] W. Jin, R. Gu, and Y. Ji, "Reward function learning for Q-learning-based geographic routing protocol," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1236–1239, Jul. 2019, doi: 10.1109/LCOMM.2019.2913360.

[32] P. W. Zaki, A. M. Hashem, E. A. Fahim, M. A. Masnour, S. M. ElGenk, M. Mashaly, and S. M. Ismail, "A novel sigmoid function approximation suitable for neural networks on FPGA," in *Proc. 15th Int. Comput. Eng. Conf. (ICENCO)*, Dec. 2019, pp. 95–99, doi: 10.1109/ICENCO48310.2019.9027479.

[33] A. Wang, X. He, P. Ghamisi, and Y. Chen, "LiDAR data classification using morphological profiles and convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 774–778, May 2018, doi: 10.1109/LGRS.2018.2810276.

[34] B. Pasca and M. Langhammer, "Activation function architectures for FPGAs," in *Proc. 28th Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2018, pp. 437–443, doi: 10.1109/FPL.2018.00015.

[35] Y. Guo, L. Sun, Z. Zhang, and H. He, "Algorithm research on improving activation function of convolutional neural networks," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 3582–3586, doi: 10.1109/CCDC.2019.8833156.

[36] Y. Duan and J. Wu, "Optimizing rebalance scheme for dock-less bike sharing systems with adaptive user incentive," in *Proc. 20th IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2019, pp. 176–181, doi: 10.1109/MDM.2019.00-59.

[37] W. Kumwilaisak, T. Piriyatharawet, P. Lasang, and N. Thatphithakkul, "Image denoising with deep convolutional neural and multi-directional long short-term memory networks under Poisson noise environments," *IEEE Access*, vol. 8, pp. 86998–87010, 2020, doi: 10.1109/ACCESS.2020.2991988.

[38] W. Kumwilaisak, *Intelligent Visual Coding*, 11st ed. Bangkok, Thailand: King Mongkut's Univ. Technology Thonburi, 2019. [Online]. Available: https://drive.google.com/file/d/1vEwUbdYDoOtiwmqscHlkuVpRdnoX5hRu/view?usp=sharing

[39] *Framework a Service Level Agreement*, document ITU-T Recommendation E-860, Jun. 2002.

[40] *Multimedia Telecommunication Relay Services*, document ITU-T Recommendation F.930, Mar. 2018.

WUTTIPONG KUMWILAISAK (Member, IEEE) received the B.E. degree in electrical engineering from Chulalongkorn University, in 1995, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, in 2003, with the support from Thai Government Scholarship.

From May 2001 to August 2001, he was a Research Intern with Ericsson Eurolab, Aachen, Germany, and with Microsoft Research Asia, Beijing, China, in 2002. From April 2003 to August 2004, he was a Senior Engineer and a Project Leader of the Mobile Platform Solution Team and the Multimedia Laboratory, Samsung Electronics, Suwon, South Korea. He was a Postdoctoral Fellow with the Thomson Research Laboratory, Princeton, NJ, USA, from March 2006 to November 2006. He has been an Associate Professor with the Electronics and Telecommunication Department, King Mongkut's University of Technology Thonburi, Bangkok, Thailand. His research interests include optimization and algorithmic design for wireless communications, multimedia communication systems, multimedia communication, multimedia compression and processing, and deep learning.

SARAVUT PHIKULNGOEN received the B.E. degree in electrical engineering from the Suranaree University of Technology (SUT), in 2018, and the M.E. degree from the King Mongkut's University of Technology Thonburi (KMUTT), Thailand, in 2021. His research interests include image processing, computer vision, deep learning, object detection, and learning-based depth estimation.

NATTANUM THATPHITHAKKUL received the B.E. and M.E. degrees from Suranaree University, Thailand, in 2000 and 2002, respectively, and the Ph.D. degree in computer engineering from the King Mongkut's Institute of Technology Ladkrabang, in 2008. He is currently the Chief of the Accessibility and Assistive Technology Research Team, National Science and Technology Development Agency, Thailand. His research interests include speech and speaker recognition, speech synthesis, natural language processing, human–machine interaction, and assistive technology.

JITPINUN PIRIYATARAVET received the B.E. degree in computer engineering and the M.E. degree in telecommunication engineering from the King Mongkut's University of Technology Thonburi (KMUTT), Thailand, in 2018 and 2021, respectively. Her research interests include deep learning, artificial intelligence, data analysis, image processing, computer vision, and light-weight deep learning for real-time analysis of traffic flow information.

CHATCHAWARN HANSAKUNBUNTHEUNG received the B.E. and M.E. degrees in electrical engineering from Chulalongkorn University, Thailand, in 1998 and 2000, respectively, and the Ph.D. degree in global information and telecommunication studies from the Waseda University, Tokyo, Japan, in 2010. He is currently the Researcher of the Accessibility and Assistive Technology Research Team, National Science and Technology Development Agency, Thailand. His research interests include speech and speaker recognition, speech synthesis, natural language processing, human–machine interaction, and assistive technology.

● ● ●