BenchCouncil Transactions on Benchmarks, Standards and Evaluations 5 (2025) 100213

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-onbenchmarks-standards-and-evaluations/

Research Article

Predicting the number of call center incoming calls using deep learning

Armaghan Nikfar [a,b], Javad Mohammadzadeh [a,b,*]

[a] Department of Computer Engineering, Ka.C., Islamic Azad University, Karaj, Iran
[b] Institute of Artificial Intelligence and Social and Advanced Technologies, Ka.C., Islamic Azad University, Karaj, Iran

ARTICLE INFO

Keywords:
Call volume
Prediction
Bidirectional long short-term memory (BLSTM)
Feature extraction customer service
management holiday impact on call traffic

ABSTRACT

One of the main problems in call centers is the call queue. This can lead to long waiting times for customers, increased frustration and call abandonment. The important role that predictive analytics plays in optimizing call center operations is increasingly recognized. Advanced models can be trained by training datasets such as the number of calls that have occurred throughout history, and by estimating how religious and public holidays have affected the weight of hours and the number of calls, and this study provides an analysis of 4 years. Call center data from Shatel, an Internet service provider. Predictive deep learning models, specifically the Bidirectional Short-Term Memory Model (BLSTM), were used to predict the number of incoming calls, predict the number of calls to centers, and prevent call queues with an accuracy of 90.56.

1. Introduction

Today, managing incoming calls in call centers is one of the main challenges of the organization. Accurately predicting the number of calls can significantly help organizations improve service quality. In the call center of a company, various functions and tasks are performed to ensure effective communication with customers and provide effective services to them. These include dealing with customer questions, solving their problems, processing orders, order support, selling new products, introducing products, buying advice, technical support, etc. Answering customer calls can lead to call queues. In general, call queues occur when the number of incoming calls to the call center exceeds the number of available agents to answer them. This technology helps call centers avoid queues by accurately predicting the number of calls on an hourly basis. This prediction reduces waiting time and increases customer service. In addition, AI adapts in real time to dynamic factors such as queue length, thereby improving operational efficiency and saving money. This can lead to better customer satisfaction, more efficient operations and ultimately a more effective call center. Choosing the best architecture for predictive models, depends on the task and data characteristics [1].

A call center is a centralized office or facility where a company handles a large volume of telephone calls, primarily for customer service, support, sales, and other inquiries. Agents in a call center manage inbound and outbound calls, providing assistance, resolving issues, processing orders, and conducting telemarketing or market research. Call centers play a critical role in maintaining effective communication between a company and its customers, ensuring customer satisfaction, efficient problem resolution, and streamlined operations. These are the reasons for the importance of predicting call center calls:

Anticipating and managing the volume of calls helps companies at any given time and prevents overtime or shortage of human resources. This efficiency reduces operating costs and improves resource utilization. Lower call volume means shorter wait times and faster service, leading to higher customer satisfaction. Customers appreciate timely responses and quick solutions to their problems. Optimizing the volume of calls, reducing costs with employees, including the use of systems, headsets, etc., overtime and infrastructure maintenance. Analyzing traffic patterns provides insights into customer behavior, seasonal trends and fluctuations in service demand. This data can help with strategic decisions to improve service delivery and operational planning [2]. Overall, minimizing contact center traffic through accurate forecasting and management and timely strategic decisions not only improves operational efficiency, but also increases customer satisfaction, reduces costs, and provides strategic benefits to the organization. This project was implemented at Shatel, an Internet Service Provider (ISP), with excellent results. Accurately forecasting the number of calls significantly improved the efficiency of Shatel's call center, reduced customer waiting time, and increased overall customer satisfaction. It also made Shatel able to consider the experts needed for this number of calls based on the

Peer review under the responsibility of The International Open Benchmark Council.
* Corresponding author.
E-mail address: J.mohammadzadeh@kiau.ac.ir (J. Mohammadzadeh).

https://doi.org/10.1016/j.tbench.2025.100213
Received 30 January 2025; Received in revised form 24 April 2025; Accepted 14 May 2025
Available online 4 June 2025

prediction of the number of calls. There are different types of deep learning such as LSTM, GRU, BLSTM, CNN, …In this model, a Bidirectional short-term memory model (BLSTM) has been used in the AI system. The BLSTM model is particularly suitable for this application because it can process sequential data in forward and backward directions and capture dependencies over long periods. This capability is critical to accurate call count forecasting, as it allows the model to understand and incorporate historical data patterns, including seasonal trends and specific time-based behaviors. By using the advanced features of the BLSTM model, the AI can make highly accurate predictions and ensure that call centers are well prepared for different call volumes throughout the year. Reducing call center traffic is the purpose of this project.

2. Literature review

Recent developments in deep learning methods impacted various fields throughout these years. Call center management is one of the fields. A comprehensive review of call center management challenges has been reviewed. This section also contains history and applications of used methods for this paper. The following subsections provide a comprehensive analysis of the literature categorized into four key themes: Call Centers, LSTM, BiLSTM, and GRU.

2.1. Call centers

Call centers have such a huge impact in today's business world. Call centers have challenges in several main domains, including forecasting, capacity planning, queueing, and personnel scheduling [3]. As [4] says " Accurately modeling and forecasting future call arrival volumes is a complicated issue which is critical for making important operational decisions, such as staffing and scheduling, in the call center ".

In [5] authors demonstrated enhanced prediction performance compared to traditional models like CNN and Transformer by utilizing sequence data and improving the MMoE algorithm.

Another approach used a Long Short-Term Memory (LSTM) network to predict call traffic characteristics and capture non-linear patterns. These predictions are incorporated into a reinforcement learning-based model for optimizing agent schedules dynamically, improving efficiency and maintaining service quality [2].

A case study proved that applying RNN-based deep learning algorithms to automate customer complaint classification, validate the suitability of deep learning for call center complaint management [6].

A hierarchical LSTM architecture used for customer satisfaction estimation. Lower LSTM used for capturing sequential information within customer dialogue turns, while the upper LSTM estimated overall satisfaction using aggregated turn-level outputs [7].

The main goal of one of the studies was to predict call volumes over a 40-day horizon for better workforce management and LSTM models were tested for capturing temporal patterns in the data. However statistical models such as SARIMAX performed better in this study [8].

2.2. Long-short term memory networks (LSTM)

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) aimed at mitigating the vanishing gradient problem (Hochreiter, Sepp [9]). LSTM is an RNN sort in which other nodes in the same layer are linked to boost learning with the removal and retrieval of relevant knowledge [10]. Fig. 1 has shown a complete architecture of the LSTM network.

In [11] LSTM was used to handle long-term dependencies in the traffic data combined with the TCN for extracting short-term features.

In a study authors propose an objective framework based on speech signal processing to classify agent's productivity levels. The LSTM network combined with attention layers was used to capture the sequential nature of speech signals [12].

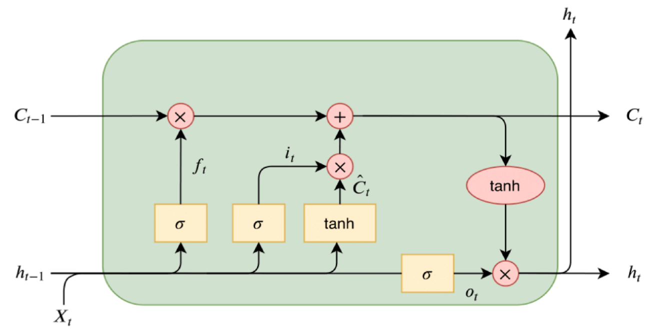

Fig. 1. Long-short term memory networks.

In several studies LSTM's ability to capture long term dependencies have been used in multiple domains such as predicting Traffic Speed [13] and stock market's price [14].

Analyzing time-series is another field that LSTM have shown promising results. Analyzing Missing and spatial data also have been studied and LSTM outperformed other algorithms [15].

2.3. Bidirectional LSTM (BLSTM)

Bidirectional LSTM (BLSTM) is a recurrent neural network used primarily on natural language processing. Unlike standard LSTM, the input flows in both directions, and it's capable of utilizing information from both sides. It's also a powerful tool for modeling the sequential dependencies between words and phrases in both directions of the sequence [16]. In Fig. 2 the full architecture of BLSTM has been shown.

Both forward and backward directions will be used for analyzing data in BLSTM networks. In [17] study, The BLSTM component enhanced prediction by analyzing data in both forward and backward directions, and achieved an average accuracy of 97.68 %.

A study compared a Uni-LSTM with BLSTM to create a model to predict short-term traffic and the BLSTM, which processed input data in both forward and backward directions, outperformed the Uni-LSTM for that matter [18].

Combination of CNN for extracting high-level features from input wind speed time series data and BLSTM for BLSTM processing these features bidirectionally as hybrid model, created a reliable model to predict wind speed with high accuracy [19].

BLSTM combined with Support Vector Machine (SVM) also had been used in language models for sentimental analysis. BLSTM network captured sequential dependencies and context in Arabic text. This sentimental analysis helped identify customer sentiments to assess customer satisfaction and improve services [20].

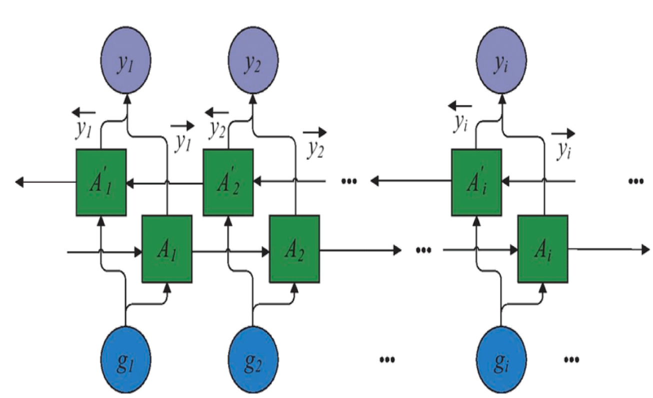

Fig. 2. Architecture from BLSTM model.

2.4. The gated recurrent unit (GRU)

Gated Recurrent Neural Network (RNN) have shown successful results in several applications including sequential data, natural language processing, machine translation, speech recognition [21], financial sequence predictions [22], Machine health monitoring [23], etc. In Fig. 3 complete architecture of GRU model has been shown.

A study compared traditional models like ARIMA and LSTM with GRU-based approaches for predicting traffic flow. GRU-based methods showed superior accuracy and lower computational requirements [24].

Variants of GRU models, such as regression GRU and stacked GRU have been studied and compared in a paper to create a prediction system for electricity generation's planning and operation, are applied to improve prediction accuracy. The multi-GRU model outperformed conventional approaches like RNN and LSTM [25].

Low computational advantage with faster training of GRU-based models had been used in a study for stock prediction. GRU-based model demonstrated its advantages, while LSTM excels in capturing long-term dependencies creating optimized hybrid model [26].

Applying hybrid GRU-LSTM models were also been used in other studies such as [13] to predict traffic speeds using data from multiple heterogeneous sources. As mentioned before, LSTM captured long-term temporal dependencies, while GRU handled short-term patterns, leading to improved performance. Proposed model demonstrated high accuracy in predicting urban traffic speed.

3. Previous research

Much literature has been developed in the forecasting of call arrivals.

According to [6] For service providers, a call center is a communication bridge between customers and providers that can handle any inquiries and requests.

(Ibrahim, R et al. [27]) said one important source of this uncertainty is the call arrival rate, which is typically time-varying, stochastic, dependent across time periods and call types, and often affected by external events. the specifics of the forecasting procedure need to be determined carefully, including the forecasting horizon (for time intervals, for a day, or for multiple days), and whether to combine arrivals from separate queues.

Also according to (Chacón, H et al. [28]) For any call center facility, the number of call arrivals represents a key component between customer satisfaction and budget constraints. Hence, the ability to accurately forecast the number of calls for a particular period of time is an effective measure in planning customer service and reducing any resource waste. This research presents a comparison between traditional time series forecasting methods and machine/deep learning techniques to predict the number of call arrivals for short and long term periods.

In (Kumwilaisak, W, et al. [2]) said Without effective staff allocation, improper workforce management can degrade service quality and reduce customer satisfaction. So they decided to create a deep neural network method 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 call center traffic prediction utilized a deep neural network containing the LSTM and the fully connected networks to predict call center traffic parameters.

In (Mohammed, R. A., & Pang, P [29]) said A call center operates with customers calls directed to agents for service based on online call traffic prediction. This paper proposes an agent call prediction method that uses agent skill information as prior knowledge for call prediction. This research shows that the use of deep neural networks and hybrid models can be very effective in predicting call traffic, customer anger, and telecommunications network traffic. These predictions help improve service quality, customer satisfaction, and resource efficiency. In the research we conducted, we will predict the number of incoming calls for better call center management based on the factors affecting the call.

Also in this paper, (Bugarčić, P et al. [30]) presented a solution using machine learning that predicts the number of calls entering the call center per hour using supervised machine learning. For the prediction, they used the WEKA machine learning software tool and the prediction results are verified using several methods, which show very good results.

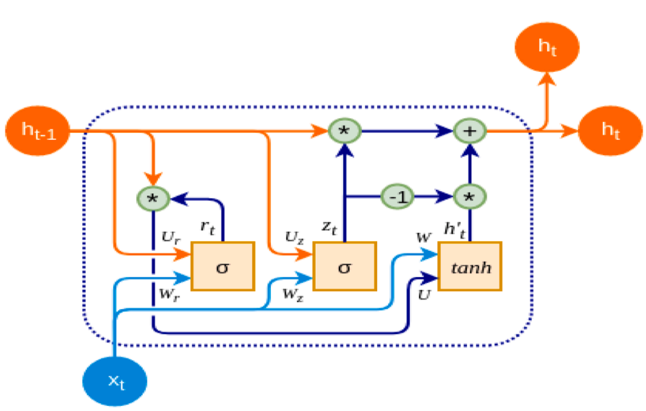

Fig. 3. Architecture of GRU model.

4. Predicting number of call's

In call centers, one of the most critical aspects is preventing the formation of long call queues. When no queues exist, customer satisfaction significantly increases because callers can quickly connect to agents and resolve their issues. On the other hand, when queues form, customers waiting on hold for extended periods may become frustrated, leading to dissatisfaction. To address this, we developed a predictive model capable of forecasting the number of calls based on various influencing factors.

Factors Influencing Call Queue Congestion

Several factors affect the number of calls a call center receives, which in turn affects the likelihood of a call queue forming. We will explore these factors below.

Holidays: During public or festive holidays, call volumes tend to decrease as people are often traveling or engaged in other activities.

Off-peak hours: During late-night hours or non-business hours, the number of calls naturally drops.

To improve the prediction process, we collected all relevant factors influencing call queues and incorporated them into the model as features. Each feature was assigned an appropriate weight to ensure the model could effectively learn their importance.

4.1. Model selection and approach

Given that our data is time-dependent and sequential, classification models were deemed the best fit for this task. Among them, LSTM (Long Short-Term Memory) networks were selected as the optimal choice due to their ability to handle sequential data and remember temporal dependencies.

Our LSTM model was trained on historical call volume data categorized by date and hour over several years. The model also accounts for nuances in the Iranian calendar, such as leap years, as well as official and unofficial holidays, including religious events, public holidays, and more. The model automatically preprocesses the data based on the input year, labels holidays, and prepares the dataset for prediction.

Once the data is prepared, the model predicts call volumes for each hour and date, enabling proactive planning.

4.2. Feature selection and performance improvement

Feature selection played a vital role in enhancing the accuracy of our predictions. By identifying the most relevant features and eliminating irrelevant ones, the model's precision improved. When we compared the model's accuracy with and without feature selection, the version with feature selection performed significantly better.

4.3. Predicting required staffing levels

In addition to predicting call volumes, our model also determines the optimal number of agents required to handle customer inquiries without causing delays. This calculation takes into account:

1. The predicted number of calls.
2. The likelihood of agents temporarily stepping away from their desks for short breaks.

Using historical data on agent availability during shifts, the model estimates the minimum number of agents required for each hour. To further improve accuracy, we added calculations to predict how many agents might step away from the queue during specific times.

As a result, the model provides highly accurate staffing recommendations for each shift. Managers can use this information to optimize resource allocation and ensure smooth operations during peak and non-peak hours. Additionally, this capability supports HR departments in workforce planning and recruitment by offering precise estimates of staffing needs.

5. Results and impact

We validated the LSTM model's predictions against real-world data, and the differences were minimal. For staffing predictions, we conducted trial runs using the model's recommendations. During these test shifts, no call queues were formed, demonstrating the effectiveness of the predicted staffing levels.

Benefits for Call Centers

Implementing this model in call centers offers several advantages, including:

Enhanced customer satisfaction: Reduced wait times and quicker issue resolution lead to happier customers.

Optimized workforce allocation: The model ensures sufficient agents are available without overstaffing.

Cost savings: By efficiently allocating human resources, unnecessary staffing costs are minimized.

Reduced agent stress: Avoiding overwhelming call volumes prevents burnout and ensures a healthier work environment.

In conclusion, this predictive model serves as a robust tool for call centers, streamlining operations, improving customer experience, and ensuring efficient use of resources.

The implementation of a predictive LSTM model for call centers marks a significant leap in improving operational efficiency and customer satisfaction. By accurately forecasting call volumes, our model empowers call centers to make informed, proactive decisions that prevent call queue congestion. This, in turn, enhances the customer experience, as callers are connected to agents promptly, ensuring faster resolution of their issues.

5.1. Key capabilities of the model

Accurate Call Volume Prediction: The model leverages historical data and temporal dependencies to forecast call volumes with high precision. It accounts for specific characteristics of the Iranian calendar, such as leap years, public and unofficial holidays, and religious events, ensuring that predictions are tailored to real-world conditions.

Feature Integration and Weight Assignment: By incorporating a wide range of influential factors such as holidays, off-peak hours, and historical call patterns, the model learns the impact of each feature. Assigning appropriate weights to these factors enhances the model's ability to recognize patterns and adapt to varying conditions.

Staffing Optimization: Beyond predicting call volumes, the model accurately calculates the number of agents required for each shift. It considers the dynamic nature of call centers, including factors like agent breaks and temporary unavailability. This capability ensures sufficient staffing to handle call volumes efficiently, reducing customer wait times and preventing agent burnout.

Real-Time Adaptability: The model dynamically adjusts predictions based on input data for a given year, pre-labeling holidays and special events. This adaptability ensures that predictions remain relevant, even as conditions change.

Scalability and Applicability: The model is not limited to a single context or region. It can be adapted to different calendars, cultural events, and operational setups, making it a versatile tool for call centers worldwide.

5.2. Methodological contributions and practical novelty

While this research utilizes established deep learning models such as LSTM, GRU, and BLSTM, its methodological contributions lie in the following novel aspects:

Integration of Iranian Calendar-Based Features: Our model incorporates culturally specific temporal features such as Nowruz, religious holidays, summer breaks, and leap years. These are encoded dynamically and allow the model to learn region-specific patterns that are not commonly addressed in previous literature.

Dynamic Year-Adaptive Preprocessing: Unlike static approaches, our model can adaptively generate holiday features for any target year (e.g., 1403) without manual data labeling. This automation adds a novel layer to the preprocessing pipeline, making the system scalable and usable across different years and calendars.

Real-World Deployment and Validation: This study goes beyond theoretical analysis and demonstrates real-world deployment within Shatel's call center. The performance of the model was evaluated in operational settings, revealing its ability to eliminate call queues and enhance workforce planning.

Predictive Staffing Mechanism: In addition to call forecasting, we propose a method to calculate the required number of agents per hour, incorporating factors like queue size, AHT, and allowable breaks. This integration of prediction with resource optimization adds a practical, business-driven value to the model.

6. Methods

6.1. Datasets process

Datasets: Data collection begins with the collection of historical call data from the Shatel call center. As we know, call centers receive a lot of calls every day. These calls will be stored in a database. To be precise, our dataset consists of the same number of incoming calls to the call center every hour, for the past 4 years. Below is a sample of these calls (due to Shatel's security policy, these numbers are not real) (Table 1).

Similarly, data will be collected hourly for 4 consecutive years and will enter the pre-processing stage.

Data preprocessing: Before training the BLSTM model, the collected data undergoes preprocessing steps such as:

Table 1
Sample of datasets.

DATE	HOURS	CALLS
3/20/2020	0	50
3/20/2020	1	35
3/20/2020	2	20
3/20/2020	3	10

6.1.1. **Normalization:** normalized value, x', is computed as

$$x' = \frac{x-\mu}{\sigma} \tag{1}$$

where x' is the original value, μ is the mean, and σ is the standard deviation of the feature.

This ensures all numerical inputs have zero mean and unit variance for better convergence.

6.1.2. Encoding categorical variables

Categorical variables (such as holidays) are converted to a numerical representation (binary or mono-hot encoding) that can be handled by the model. For binary encoding of holidays (e.g., Nowruz, weekends, etc.), assign:

$$Encoded\ Value = \{1,\ if\ holiday\ or\ weekend\ 0,\ otherwise$$

For mono-hot encoding (if categorical variables have more than two states):

A categorical feature F with k states is encoded as a vector of size k, where only one element is 1 and others are 0.

6.1.3. Sequence formation

Data is organized into overlapping input-output pairs. For example, given a sequence of hourly call volumes$[x_1,\ x_2, \dots\ x_n]$:

$$Input\ Sequence:\ [x_1,\ x_2, \dots\ x_n]$$

n: n is a hyperparameter that defines the length of the historical window used for prediction. A larger n allows the model to consider longer-term dependencies, while a smaller n focuses on more immediate trends. The choice of n is determined experimentally based on the dataset and model performance.

Output (Prediction Target): x_{t+1}

6.2. Feature extraction

To extract factors for predicting call volume, the model uses statistical statistics to determine whether the year is a leap year (using statistical rules), and what religious and administrative holidays have that this data set consists of two elements (0 or 1).

These features include:

6.2.1. Hours

The hours are so important because the weight of hours are effective in incoming calls, for example in nights the weight of calls are stronger the other hours because calls will be increase at nights.

6.2.2. Dates

dates are features too because it is important to know which dates have more calls and which day it is.

6.2.3. Nowruz holidays

These holidays are usually associated with a few calls in the morning when people communicate with family and friends.

6.2.4. Summer holidays

These can result in reduced working hours throughout the day due to longer holiday periods.

6.2.5. Holidays

These can directly impact the amount of time associated with performance issues.

6.2.6. Religious holidays

These holidays have a different impact on the amount of time, such as reduced morning visits on Eid days. each of these factors affects the weight of the watch. For example, a model that considers the Nowruz holidays as an important festival is less likely to predict fewer calls in the morning compared to the evening and night.by incorporating these sources and learning from historical data, the BLSTM model can accurately predict future call volume. This approach helps improve customer call center operations by anticipating and managing call traffic during various holiday periods.

6.2.7. Day off

in these days incoming calls will be less at morning and evening, because in day off works are closed and we don't have calls from companies at morning and evening.

6.3. Calculating features

In estimating the attributes, the model includes a variety of factors that affect call volume and can be used to predict hourly counts

6.3.1. LSTM cell has three main gates

Forget gate:

$$f_t = \sigma\left(W_f.\ [h_{t-1}, x_t] + b_f\right) \tag{2}$$

Input gate:

$$i_t = \sigma(W_t.\ [h_{t-1}, x_t] + b_i \tag{3}$$

Output gate:

$$O_t = \sigma(W_o.\ [h_{t-1}, x_t] + b_o) \tag{4}$$

Candidate memory cell:

$$C_t = \tanh(W_c.[h_{t-1}, x_t] + b_c) \tag{5}$$

Update memory cell:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C_t \tag{6}$$

Compute hidden state:

$$h_t = o_t.tanh\ (C_t) \tag{7}$$

6.3.2. Extension

BLSTM processes data in both forward and backward directions:

Forward pass is $\overrightarrow{h_t}$ and backward pass is $h_t^{\leftarrow}$.

Final output at time t:

$$h_t = \overrightarrow{h_t} \oplus h_t^{\leftarrow} \tag{8}$$

(where ⊕ is concatenation). with these computed features, the Bidirectional Long Short-Term Memory (BLSTM) model forecasts hourly call volumes. BLSTM is a type of recurrent neural network (RNN) that processes sequences bidirectionally. capturing dependencies in both forward and backward directions through time. This capability allows the model to learn from historical data patterns and make predictions based on the temporal relationships between the extracted features and call volumes. By leveraging these features, the BLSTM model enhances the accuracy of its predictions, enabling businesses to optimize resource allocation and improve customer service during various temporal and cultural contexts.

6.4. Training

6.4.1. Loss function

To minimize the prediction error:

$$\mathrm{L} = \frac{1}{n}\sum_{n}^{i=1}\left(y_i - \breve{y}_i\right) \tag{9}$$

6.4.2. *Algorithm*

Gradient-based optimization (e.g., Adam optimizer):

$$\theta_{t+1} = \theta_t - n.\nabla_\theta L \tag{10}$$

6.5. *Model architecture*

In the research we described, the BLSTM (Bidirectional Long Short-Term Memory) model serves as a crucial component of artificial intelligence. Specifically, it is used to predict the number of incoming calls for the upcoming year. The main features and functionalities of the BLSTM model in this project include:

6.5.1. *Data processing*

BLSTM is adept at processing sequential data, which is essential for analyzing historical call volume patterns over time.

Forward pass:

Processes the sequence from $t = 1t = 1t = 1$ to TTT, generating a hidden state at each time step:

$$\overrightarrow{h_t} = LSTM_{forward}\left(x_{t,}\overrightarrow{h_t} - 1\right) \tag{11}$$

Backward pass:

Processes the sequence in reverse, from $t = \mathrm{T}t = \mathrm{T}t = T$ to 111:

$$h_t^{\leftarrow} = LSTM_{\text{backward}}\ (x_t, h_t^{\leftarrow} - 1) \tag{12}$$

Final hidden state:

Combines the forward and backward hidden states at each time step:

$$h_t = \overrightarrow{h_t} \oplus h_t^{\leftarrow} \tag{13}$$

6.5.2. *Capturing long-term dependencies*

it can capture dependencies and trends in call volumes that span across longer periods, such as seasonal fluctuations or annual trends.

6.5.3. *Processing*

unlike traditional LSTM models, BLSTM processes data in both forward and backward directions. This bidirectional capability allows it to learn from past and future contexts simultaneously, enhancing its understanding of temporal dynamics in call patterns.

6.5.4. *Feature extraction*

by leveraging its multiple layers and memory cells, BLSTM can automatically extract relevant features from historical data, such as time of day, day of week, holidays, and special events, which are crucial for accurate call volume predictions.

6.5.5. *Prediction accuracy*

the advanced architecture of BLSTM enables it to provide more accurate forecasts compared to simpler models, as it can learn intricate patterns and dependencies present in call center data (Figs. 4 and 5).

6.6. *Calculation the number of agents*

6.6.1. *Predicted call volume*

The predicted number of calls for hour **h** on day **d** is denoted as $\mathscr{C}_{d,h}$. This value is provided by the BLSTM model.

6.6.2. *Average call handling time (AHT)*

The standard average call handling time is:

$$AHT = 7minutes/call(\approx 0.1167hours\,/\,call) \tag{14}$$

6.6.3. *Required agents without considering the queue*

To calculate the base number of agents required to handle the predicted call volume $\mathscr{C}_{d,h}$. within one hour, we use:

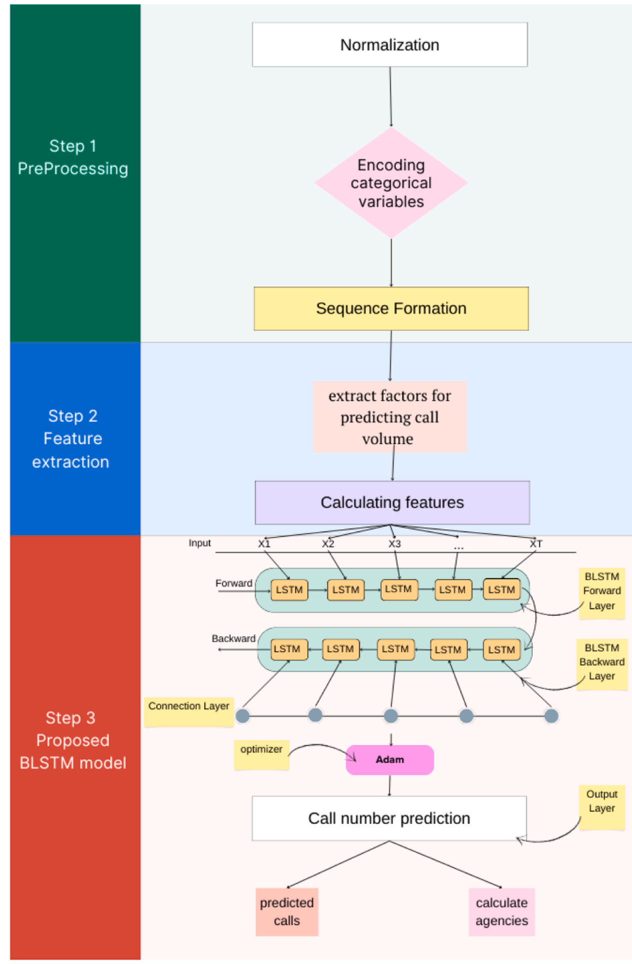

Fig. 4. Inputs and BLSTM architecture.

$$n_{\text{base}} = \frac{C_{d,h.} \times AHT}{1} \tag{15}$$

6.6.4. *Adjusting for the queue size (Q)*

The model adjusts the required number of agents based on the size of the queue Q_1 as follows:

Let k_{logoff} be the number of agents allowed to log off based on queue size (Q):

$$k_{logoff} = \{0\ if\ Q < 10\ 1\ if\ 10 \le Q < 20\ 2\ if\ 20 \le Q < 30 \tag{16}$$

6.6.5. *Adjusted number of agents*

The required number of agents is then:

$$n_{required} = n_{base} - k_{logoff} \tag{17}$$

6.7. *Ensuring adequate staffing levels*

To ensure agents are available to handle predicted call volumes effectively, the number of agents $n_{required}$ must meet or exceed a minimum threshold based on the target service level (SL):

$$n_{final} = max\left(n_{required}\ ,\ n_{min}\right) \tag{18}$$

Where n_{min} is the minimum number of agents needed to meet the desired service level (e.g., 80 % of calls answered within 30 s).

This method calculates the required number of agents per hour by factoring in the predicted call volumes, the average call handling time, and the probabilities of agents logging off based on current queue

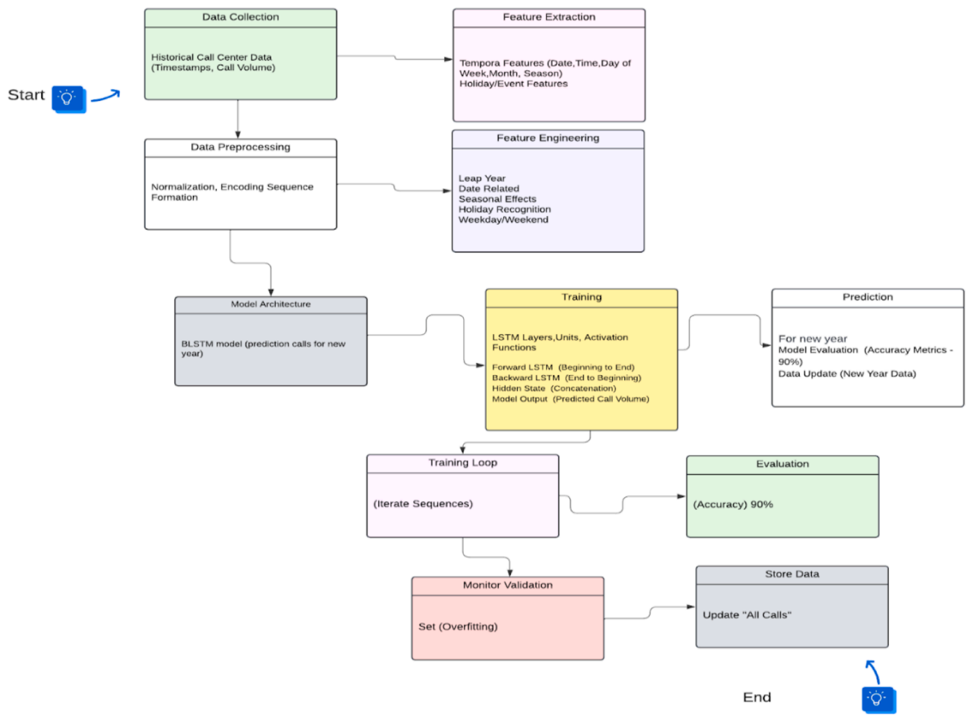

Fig. 5. step by step diagram of full model architecture.

conditions. This calculation is then displayed in Excel to ensure that sufficient agents are available during each time interval to handle incoming calls effectively. The goal is to ensure adequate staffing to respond to incoming calls within acceptable service levels (Table 2).

6.8. Train data using bidirectional long short-term memory (BLSTM)

Define the BLSTM architecture: Choose the number of LSTM layers (stacked layers can capture complex dependencies).

Set the number of units per layer (a balance between model capacity and complexity is needed).

Define activation functions for gates and hidden layers (commonly sigmoid or tanH for gates, tanH or ReLU for hidden layers). Split the preprocessed data into training, validation, and testing sets.

Train the BLSTM model using an optimizer (e.g., Adam) and a loss function (e.g., Mean Squared Error) to minimize the difference between predicted and actual call volumes.

During training, the model iterates through sequences:

At each time step (t) within a sequence: The input feature vector (containing past call volumes and other features) is denoted as e(t).

The forget gate (f(t)) determines information to retain from the previous cell's memory state (c(t-1)).

The input gate (i(t)) controls the information from e(t) to be incorporated into the cell's memory.

The candidate memory cell value (c^(t)) is calculated based on e(t) and the previous memory state.

The output gate (o(t)) determines the information from the current memory cell (c(t)) to be included in the hidden state (h(t)).

The hidden state (h(t)) at the current time step captures the temporal dependencies learned from the sequence so far.

A backward LSTM processes the sequence in reverse, generating hidden states (h^-(t)) capturing future context.

The final hidden state for each time step (t) is obtained by concatenating the forward and backward hidden states: h(t) = [h^(t), h^-(t)]

The final output of the BLSTM for a sequence is a series of hidden state vectors, h(1), h(2), …, h(T) (where T is the sequence length).

This output is then used to predict the call volume for the subsequent time window.

Monitor the model's performance on the validation set to prevent overfitting [31].

6.9. Prediction

The trained model is used to predict the calls count for the new year (e.g., 1403 in the Persian calendar), and the results are stored in an excel file.

This prediction was for a year ago and when it was compared with the test data, it showed an accuracy of 90.56 %, which was a very good prediction accuracy.

6.10. Model evaluation and improvement

At the end of each year, the model's accuracy is evaluated using the actual data from that year (for example, 1403 in the Persian calendar). Additionally, the actual data from the year (1403) is added to the model's memory, stored in an Excel file named "All Calls." This process ensures that the model is continually updated with current data, improving its performance over time by allowing it to adapt to new trends and patterns in the data, leading to more accurate predictions in subsequent years

Table 2
Sample pf model's prediction.

Date	Hour	Predicted calls	Queue	Required agents
17 Aug 2024	08:00	22	3	10
17 Aug 2024	09:00	30	0	18
17 Aug 2024	10:00	45	2	30

7. The results

The current prediction with BLSTM model has achieved an accuracy of 90 %. This accuracy was obtained by comparing its predictions of the number of calls for the year 1402 with the actual data from that year.

Considering the principle in artificial intelligence where an accuracy above 90 % is considered excellent, and noting that the model also incurs errors if it deviates by >10 calls in its predictions, it has performed precise predictions. With continued training on new data, it is expected that the prediction accuracy of this model will improve in future years.

This model earns knowledge by means of the data input and stores it in its own implanted memory. After that we can give the model last year and it will start predicting for current year and it outputs the chances of any year being a leap year and sets the holidays within the dataset equal to zeros and ones (the program decides the features on its own).

Then, it calculates the number of incoming calls per hour for that year, considering that each agent, on average, spends 7 min per call (based on the company's average call duration).

Taking into account the number of incoming calls and the allowable break time for each agent, which is 10 min, as well as the allowance for simultaneous breaks for multiple agents (determined by the number of incoming calls) for example, if there are 10 incoming calls, only 3 agents are allowed to take a break at the same time.

And finally the output will be an excel file with dates, hours, predicted calls, features, Agents.

This approach was useful in the avoidance of call traffic, the satisfaction of the customer, the correct placement of the number of agents and optimistic results in human resources.

In the continuation of this prediction, LSTM and GRU models were also performed in order to compare the accuracy of these two prediction models.

To enhance the practical application of this study, a real-time dashboard was developed to visualize and interact with the BLSTM model's predictions. This dashboard provides a user-friendly interface for call center managers to monitor and analyze predicted call volumes, allowing for dynamic adjustments in workforce allocation and scheduling.

The main features of this dashboard include:

7.1. Real-Time prediction display

The dashboard displays predicted call volumes on an hourly and daily basis through various charts (e.g., line and bar charts). This visual representation helps managers to quickly understand expected call trends and prepare resources accordingly.

7.2. Adjustable settings for prediction parameters

Users can interact with the dashboard to modify prediction parameters, such as considering specific holidays, working hours, or unique events that may impact call volumes. These adjustments allow the model to incorporate real-time data factors for improved accuracy.

7.3. Alert system for high call volumes

The dashboard includes an alert system that notifies managers when predicted call volumes exceed a certain threshold. This feature enables the call center to proactively adjust staffing levels to meet increased demand and minimize customer wait times.

7.4. Analytical reports and comparison

The dashboard provides daily, weekly, and monthly analytical reports, comparing the predicted and actual call volumes. These reports help managers assess the model's accuracy and track call patterns over time, aiding in strategic decision-making.

7.5. Data upload and model update

For continuous improvement, the dashboard allows for new data uploads, updating the model with recent call history to maintain high prediction accuracy. This function supports ongoing model optimization as the call center adapts to changing conditions.

7.6. The results of call prediction vs real calls in BLSTM, LSTM & GRU models

To further evaluate the significance of our model's performance, we conducted a comparative analysis not only in terms of raw accuracy percentage, but also in terms of practical impact in a real-world scenario.

While the difference in accuracy between BLSTM (90.56 %), LSTM (89.07 %), and GRU (88.33 %) may seem incremental at first glance, this translates to a substantial operational benefit in a call center context. For instance, considering a call center that handles 10,000 calls per day, a 1.5 % improvement means approximately 150 additional calls per day are correctly predicted, allowing for more efficient staffing and reducing missed or delayed calls significantly.

Moreover, we deployed each of the three models (BLSTM, LSTM, and GRU) in test simulations using real Shatel call center historical data. The simulations tracked actual call queues versus predicted agent allocations:

GRU and LSTM-based predictions resulted in 3–5 queue formation events per week during high-volume hours, requiring reactive staff reallocation.

In contrast, BLSTM-based planning led to zero queue events during the same period, showing better alignment between predicted volume and staffing needs.

These results emphasize that even a marginal improvement in model accuracy can lead to tangible operational enhancements, especially in large-scale environments like call centers (Figs. 6-9).

8. Comparing incoming calls before and after prediction

This graph shows the number of incoming calls to the call center over a 30-day period (horizontal axis). There are two lines in the graph:

Red graph (Before AI Prediction):

This line shows the number of incoming calls before the AI model was used. In this situation, the number of calls is relatively high, reaching >120 calls on some days. This high value can lead to congestion and reduced service quality.

Green graph (After AI Prediction):

This line shows the number of calls after the AI model predicted them. By predicting call patterns and optimizing workforce management, the AI model has led to a reduction in the number of incoming calls. The reduction in calls can be due to proactive measures such as automated answering, better queue management, or solving common problems before the customer calls (Fig. 10).

9. Conclusion

Based on the results obtained from the GRU, LSTM, and BLSTM models, the BLSTM model demonstrated higher accuracy.

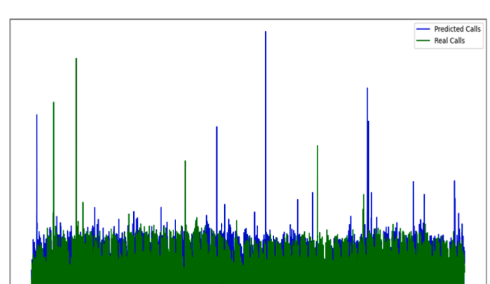

Fig. 6. BLSTM model with 90.56 % accuracy.

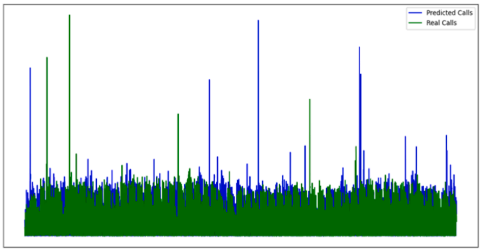

Fig. 7. LSTM model with 89.07 % accuracy.

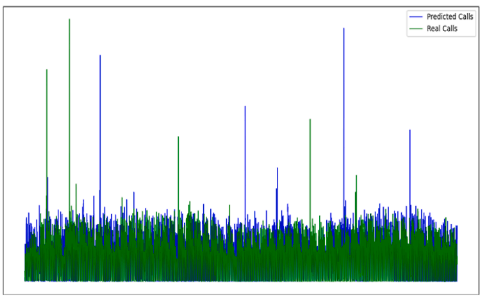

Fig. 8. GRU model with 88.33 % accuracy.

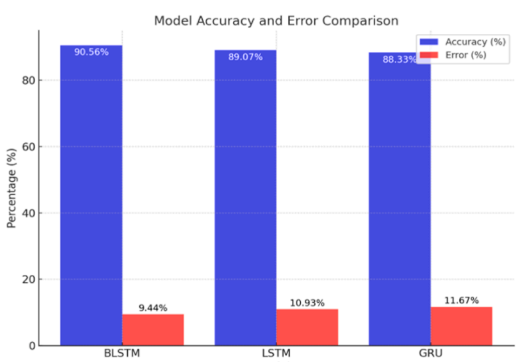

Fig. 9. Chart of model's accuracy and error comparison.

The Bidirectional Long Short-Term Memory (BLSTM) model outperformed the other models due to its ability to capture dependencies in both forward and backward directions. Unlike standard LSTM, which only processes data in one direction, BLSTM processes the input sequence from the beginning to the end and from the end to the beginning. This bidirectional approach allows the BLSTM model to have a more comprehensive understanding of the sequential data, capturing more context and relationships within the data.

The improved accuracy of the BLSTM model can be attributed to:

Enhanced Contextual Understanding: By considering both past and future contexts, the BLSTM can capture intricate patterns and dependencies in the data that might be missed by unidirectional models. Better handling of long-term dependencies: BLSTM's dual-layer processing makes it more effective at learning long-term dependencies compared to GRU and unidirectional LSTM.

Increased Robustness: The ability to analyze data from two directions provides a more robust feature extraction, leading to better performance on complex tasks.

As a result, the BLSTM model showed superior performance, making it more suitable for tasks requiring high accuracy in understanding and predicting time series data, such as forecasting incoming calls in our scenario.

In this study, the BLSTM model achieved a prediction accuracy of 90.56 %, outperforming other models such as LSTM and GRU. A similar study by Kumwilaisak et al. [2] utilized an LSTM model to forecast call volumes in a call center, achieving an accuracy of approximately 89.07. Compared to our results, the BLSTM model demonstrated superior performance due to its bidirectional processing capability, which allows it to capture dependencies in both forward and backward directions, enhancing its accuracy in complex, time-dependent data.

In the feature extraction phase, this study incorporated factors such as official holidays, religious events, hours, and days of the week. Comparison with Kumwilaisak et al. [2] reveals that our inclusion of these temporal and cultural features significantly improved prediction accuracy. While the study by Kumwilaisak et al. considered similar temporal variables, it utilized a unidirectional LSTM model, which limits its ability to capture bidirectional dependencies.

Finally, our research successfully improved the operational efficiency of Shatel's call center, enhancing customer satisfaction through the use of the BLSTM model. Similar outcomes were observed in the study by Kumwilaisak et al. [2], where deep learning was used to optimize call center operations. However, the higher accuracy achieved by our BLSTM model indicates its effectiveness for managing complex, variable data scenarios in real-world applications.

10. Future works

In future research, we aim to enhance the accuracy and performance

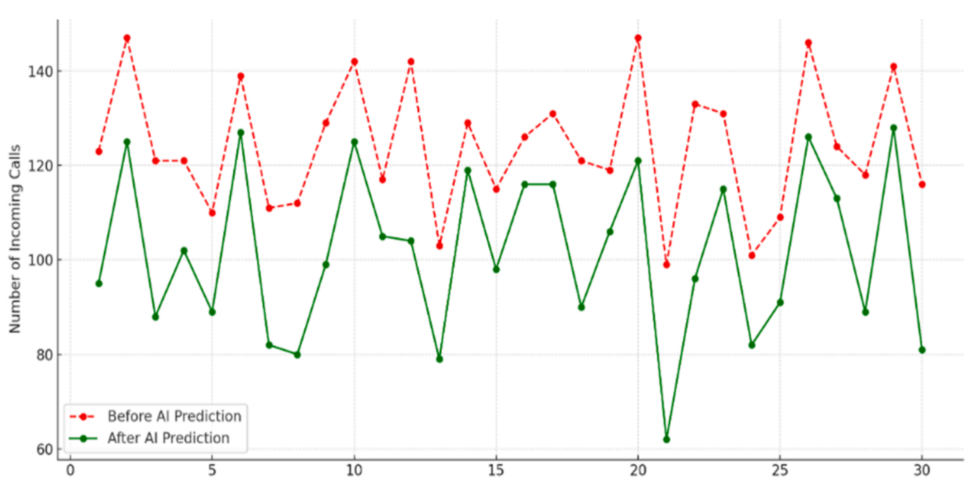

Fig. 10. The number of incoming calls of call center over a 30-day period.

of our predictive model by incorporating more advanced deep learning architectures, particularly transformers, which have demonstrated superior capabilities in sequential data modeling.

Additionally, we plan to explore different optimization algorithms beyond the ones currently used. Optimizers play a crucial role in training deep learning models, and experimenting with alternatives may lead to better convergence and generalization.

Another important aspect of our future work is the incorporation of additional domain-specific features to enrich the input data. While our current model already considers key factors such as holidays and seasonal trends, we intend to integrate customer behavioral patterns, network traffic data, sentiment analysis from support interactions, and real-time external factors that may influence call volume. By leveraging a more comprehensive feature set, we expect the model to capture nuanced variations in call center demand more effectively.

These advancements will contribute to a more accurate, scalable, and real-world-applicable forecasting system, ultimately leading to better resource allocation and customer experience optimization in call centers.

CRediT authorship contribution statement

Armaghan Nikfar: Writing – original draft, Software. **Javad Mohammadzadeh:** Supervision.

Declaration of competing interest

The authors declare that they have no conflict of interest.

Acknowledgements

The authors would like to thank Shatel team from Iran Tehran who's supporting made this research possible.

References

[1] H. Al-Selwi, F. Alharbi, M.A. Azam, M.A. Alharbi, M.M. Hassan, A cybersecurity framework for DDoS mitigation in IoT and cloud computing using RPL and deep learning, Electron. (Basel) 13 (4) (2024) 818, https://doi.org/10.3390/electronics13040818.
[2] W. Kumwilaisak, S. Phikulngoen, J. Piriyataravet, N. Thatphithakkul, C. Hansakunbuntheung, Adaptive call center workforce management with deep neural network and reinforcement learning, IEEE Access 10 (2022) 35712–35724, https://doi.org/10.1109/access.2022.3160452.
[3] Z. Aksin, et al., The modern call center: a multi-disciplinary perspective on operations management research, Prod. Oper. Manag. 16 (6) (2007) 665–688.
[4] R. Ibrahim, H. Ye, P. L'Ecuyer, H. Shen, Modeling and forecasting call center arrivals: a literature survey and a case study, Int. J. Forecast. 32 (3) (2016) 865–874.
[5] L. Cheng, et al., Multi-modal fusion for business process prediction in call center scenarios, Inf. Fusion 108 (2024) 102362.
[6] S. Lukitasari, F. Hidayat, Deep learning-based complaint classification for Indonesia telecommunication company's call center, in: Proceedings of the 7th Mathematics, Science, and Computer Science Education International Seminar, MSCEIS 2019, Bandung, West Java,Indonesia, 2020, 12 October 2019.
[7] A. Ando, et al., Hierarchical LSTMs With Joint Learning For Estimating Customer Satisfaction from Contact Center Calls, INTERSPEECH, 2017.
[8] D. Leszko, PhD thesis, 2020.
[9] Sepp Hochreiter, Untersuchungen Zu Dynamischen Neuronalen Netzen (PDF) (diploma thesis), Technical University Munich, Institute of Computer Science, 1991.
[10] M. Phridviraj, et al., A bi-directional long short-term memory-based diabetic retinopathy detection model using retinal fundus images, Healthc. Anal. 3 (2023) 100174.
[11] J. Bi, et al., A hybrid prediction method for realistic network traffic with temporal convolutional network and LSTM, IEEE Trans. Autom. Sci. Eng. 19 (3) (2021) 1869–1879.
[12] A. Ahmed, et al., Agent productivity modeling in a call center domain using attentive convolutional neural networks, Sensors 20 (19) (2020) 5489.
[13] N. Zafar, et al., Applying hybrid LSTM-GRU model based on heterogeneous data sources for traffic speed prediction in urban areas, Sensors 22 (9) (2022) 3348.
[14] D.M. Nelson, et al., Stock market's price movement prediction with LSTM neural networks, in: 2017 International joint conference on neural networks (IJCNN), Ieee, 2017.
[15] D.-H. Shin, et al., Prediction of traffic congestion based on LSTM through correction of missing temporal and spatial data, IEEE Access 8 (2020) 150784–150796.
[16] C.C. Aggarwal, Neural Networks and Deep Learning: A Textbook, Springer, 2018, https://doi.org/10.1007/978-3-319-94463-0.
[17] W.A. Aziz, et al., Content-aware network traffic prediction framework for quality of service-aware dynamic network resource management, IEEE Access 11 (2023) 99716–99733.
[18] R.L. Abduljabbar, et al., Unidirectional and bidirectional LSTM models for short-term traffic prediction, J. Adv. Transp. 2021 (1) (2021) 5589075.
[19] A. Lawal, et al., Wind speed prediction using hybrid 1D CNN and BLSTM network, IEEE Access 9 (2021) 156672–156679.
[20] H.E. Alfar, Sentiment analysis for arabic call center notes using machine learning techniques: a case study of Jordanian dialect, Tech. Bus. Manag. 1 (1) (2022).
[21] R. Dey, F.M. Salem, Gate-variants of gated recurrent unit (GRU) neural networks, in: 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS), IEEE, 2017.
[22] G. Shen, et al., Deep learning with gated recurrent unit networks for financial sequence predictions, Procedia Comput Sci 131 (2018) 895–903.
[23] R. Zhao, et al., Machine health monitoring using local feature-based gated recurrent unit networks, IEEE Trans. Ind. Electron. 65 (2) (2017) 1539–1548.
[24] B. Hussain, et al., Intelligent traffic flow prediction using optimized GRU model, IEEE Access 9 (2021) 100736–100746.
[25] W. Li, et al., Multi-GRU prediction system for electricity generation's planning and operation, IET Gener. Transm. Distrib. 13 (9) (2019) 1630–1637.
[26] Y. Gao, et al., Stock prediction based on optimized LSTM and GRU models, Sci. Program. 2021 (1) (2021) 4055281.
[27] R. Ibrahim, et al., Modeling and forecasting call center arrivals: a literature survey and a case study, Int. J. Forecast. 32 (3) (2016) 865–874.
[28] H. Chacón, V. Koppisetti, D. Hardage, K.K.R. Choo, P. Rad, Forecasting call center arrivals using temporal memory networks and gradient boosting algorithm, Expert Syst. Appl. 224 (2023) 119983.
[29] R.A. Mohammed, P. Pang, Agent personalized call center traffic prediction and call distribution, in: Neural Information Processing: 18th International Conference, ICONIP 2011, Shanghai, China, 2011, pp. 1–10. November 13-17, 2011, Proceedings, Part II 18Springer Berlin Heidelberg.
[30] P. Bugarčić, S. Janković, S. Mladenović, Forecasting number of calls to the call center using machine learning, Transp. Today's Soc. (2021).
[31] D. Uliyan, A.S. Aljaloud, A. Alkhalil, H.S.A. Amer, M.a.E.A. Mohamed, A.F. M. Alogali, Deep learning model to predict student's retention using BLSTM and CRF, IEEE Access 9 (2021) 135550–135558, https://doi.org/10.1109/access.2021.3117117.

Further reading

[32] S.M. Al-Selwi, M.F. Hassan, S.J. Abdulkadir, A. Muneer, E.H. Sumiea, A. Alqushaibi, M.G. Ragab, RNN-LSTM: from applications to modeling techniques and beyond—Systematic review, J. King Saud. Univ. - Comput. Inf. Sci. (2024) 102068, https://doi.org/10.1016/j.jksuci.2024.102068.
[33] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Netw 18 (5–6) (2005) 602–610, https://doi.org/10.1016/j.neunet.2005.06.042.
[34] S. Hochreiter, J. Schmidhuber, Long Short-Term memory, Neural Comput 9 (8) (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.
[35] R. Huan, Z. Zhan, L. Ge, K. Chi, P. Chen, R. Liang, A hybrid CNN and BLSTM network for human complex activity recognition with multi-feature fusion, Multimed. Tools Appl. 80 (30) (2021) 36159–36182, https://doi.org/10.1007/s11042-021-11363-4.
[36] Javad Mohammadzadeh, Abadi, Ali Fallahi Rahmat (2023); "Leveraging deep learning techniques on collaborative filtering recommender systems". doi:10.22105/jarie.2021.275620.126.
[37] Z.C. Lipton, J. Berkowitz, C. Elkan, A Critical Review of Recurrent Neural Networks for Sequence Learning, Cornell University, 2015 arXiv, doi:10.48550/arxiv.1506.00019.
[38] C. Janiesch, P. Zschech, K. Heinrich, Machine learning and deep learning, Electron. Mark. 31 (3) (2021) 685–695, https://doi.org/10.1007/s12525-021-00475-2.
[39] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN Encoder-Decoder for statistical machine translation. arXiv (Cornell University), 2014. doi:10.48550/arxiv.1406.1078.
[40] Felix Gers, Jürgen Schmidhuber, Fred Cummins, Learning to forget: continual prediction with LSTM, 9th International Conference on Artificial Neural Networks: ICANN '99, 1999, 1999, pp. 850–855 ISBN 0-85296-721-7, doi:10.1049/cp:19991218.
[41] "Recurrent Neural Network Tutorial, Part 4 – implementing a GRU/LSTM RNN with python and theano – WildML". Wildml.com. 2015-10-27. Archived from the original on 2021-11-10. Retrieved May 18, 2016.
[42] M. Ravanelli, P. Brakel, M. Omologo, Y. Bengio, Light gated recurrent units for speech recognition, IEEE Trans. Emerg. Top. Comput. Intell. 2 (2) (2018) 92–102, https://doi.org/10.1109/tetci.2017.2762739.
[43] Y. Su, C.J. Kuo, On extended long short-term memory and dependent bidirectional recurrent neural network, Neurocomputing 356 (2019) 151–161, https://doi.org/10.1016/j.neucom.2019.04.044.

[44] Chung, J., Gülçehre, Ç., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1412.3555.
[45] N. Gruber, A. Jockisch, Are GRU cells more specific and LSTM cells more sensitive in motive classification of text? Front. Artif. 3 (2020) https://doi.org/10.3389/frai.2020.00040.
[46] M.E. Manaa, S.M. Hussain, N.S.A. Alasadi, N.H.a.A. Al-Khamees, DDoS attacks detection based on machine learning algorithms in IoT environments, Intel. Artif. 27 (74) (2024) 152–165, doi:10.4114/intartif.vol_27iss74pp152-165.
[47] W. Kumwilaisak, S. Phikulngoen, J. Piriyataravet, N. Thatphithakkul, C. Hansakunbuntheung, Adaptive call center workforce management with deep neural network and reinforcement learning, IEEE Access 10 (2022) 35712–35724, https://doi.org/10.1109/ACCESS.2022.3160452.
[48] M.S.B. Phridviraj, R. Bhukya, S. Madugula, A. Manjula, S. Vodithala, M.S. Waseem, A bi-directional long short-term memory-based diabetic retinopathy detection model using retinal fundus images, Comput. Biol. Med. 146 (2020) 2022 105580, doi:10.1016/j.compbiomed.2022.105580.
[49] W. Kumwilaisak, et al., Adaptive call center workforce management with deep neural network and reinforcement learning, IEEE Access 10 (2022) 35712–35724.
[50] P. Kaushik, S. Singh, P. Yadav. Traffic prediction in telecom systems using deep learning, 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future, IEEE, 2018, pp. 302–307.