ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20240305001374
Original scientific paper

# A Sporadic Classification and Regression-Based Approach to Intermittent Demand Forecasting in Smart Supply Chain

Praveena S.*, Prasanna Devi S.

**Abstract:** Intermittent demand forecasting presents a distinct problem in supply chain management, as it requires accurate prediction of demand in order to minimize costs and enhance operational efficiency for businesses. The present study introduces a novel data-driven approach for handling the problem of forecasting intermittent demand combinations across several time horizons. The approach involves building an efficient sporadic classification model and using regression techniques to predict the quantity of non-zero demand for future time horizons. This unique two-stage forecasting framework is based on the implementation of the best threshold classification methods using LGBM. The results show a significant improvement in classification accuracy for splitting intermittent requests. The output from the first phase has been given to the multimodal temporal attention-based Seq2seq approach, which prioritizes various aspects of the past in order to predict multiple future time series over different time horizons. The experiment results were obtained using the Corporación Favorita dataset, which was made publicly available for a Kaggle competition. Our approach has demonstrated good performance when compared to state-of-the-art techniques. The findings demonstrate that this study can also offer precise Smart inventory relies on upstream inputs to ensure accurate and efficient decision-making in the smart supply chain.

**Keywords:** attention learning; intermittent demand forecasting; sequence to sequence; supply chain

## 1 INTRODUCTION

Time series forecasting is a crucial task in supply chain management, aiming to accurately predict future events based on past observations [1]. This precision enhances operational efficiency across various domains of society. Data-driven demand forecasting methods help online merchants understand market demands, improving supply chain operations like delivery speed and product availability. Inventory control is essential for managing stored components and finite goods, with the primary objective being to decrease total cost of ownership [2]. Intermittent Demand (ID), or zero inflation, is a phenomenon characterized by sporadic or irregular patterns of demand [3, 4]. Supply chain demand forecasting faces significant challenges due to the lack of focus on predicting sporadic demand [5, 6]. As a result, predicting for Sporadic demand has been identified as a difficult process by researchers [7-9]. Researchers have explored various techniques, such as Croston's method [10], which divides sporadic demand into two consecutive series: total demand and demand interval. Neural Network (NN) methodology has been used to improve inventory service levels while maintaining a stable inventory holding rate [11]. Common methods for forecasting time series data include the Holt-Winters technique and ARIMA approach, but these models struggle to accurately express extremely irregular temporal data. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (*LSTM*s) have been recommended for modeling complex sequential time-related data, including NLP, acoustic waveforms, and audiovisual segments. Multi-horizon forecasting [12] is often chosen in real-world scenarios for managing resource allocation and making informed decisions over long periods.

Recent research shows that time series models commonly make one-way forecasts. This will significantly increase inventory and backlog expenses for firm replenishment. Thus, the proposed framework enhancing demand categorization precision is essential in real-world supply chains. Categorization study excludes demand variables. It highlights time series features and uses temporal correlations to predict future trends using real data.

## 2 RELATED WORK

Researchers suggest that precise demand estimation is crucial for effective forecasting strategies. The ABC classification Approach, also known as the Pareto classification technique, is widely used in various industries. The 80/20 Pareto principle is reliable in inventory management. However, forecasting is challenging due to intermittent demand, making traditional methods difficult. Alternative methods like Croston's approach [13] and Syntetos & Boylan's SBA [14] technique have been proposed to address this issue. These methods aim to make predictions more unbiased and accurate, considering non-zero demand occurrences. In this technique *ADI* (Average Demand Interval) and *CV* (Coefficient of Variation) has to be calculated over non-zero demand occurrences shown in the equation below.

$$\begin{aligned} ADI &= \frac{\text{Total number of demand periods}}{\text{Total number of non-zero demand occurences}} \\ CV &= \frac{\text{Standard deviation of all demands}}{\text{Average of demands}} \end{aligned} \tag{1}$$

As a result, it is unable to differentiate between periods of demand and periods of no demand, and it does not provide guidance for replenishment. This can lead to additional costs associated with inventory and backlog. Furthermore, other bootstrapping techniques have been employed for classification issues; however they solely depend on the intrinsic properties of the time series. Current research fails to fully exploit the benefits of machine learning in order to encompass a wider range of influential elements and generate more precise forecasts. The proposed research intends to explore the constraints of the time series-based approach in the context of the classification problem, with the objective of improving the accuracy of demand forecasts [15]. To estimate demand and sales in the supply chain domain deep learning approaches are utilized. The multiple multi-layer perceptrons (MLP) construct a consolidated forecast and contrast it with traditional methodologies. The primary drawback of this strategy is the methodology not well-suited for handling time series data, necessitating extensive feature engineering. Given the substantial volume of data typically accessible, it is not feasible solution for a supply chain problem. The time series data in terms of supply chain logistics [16] examines the performance of different deep learning architectures, such as Convolutional Neural Networks (CNN), Long-Short Term Memory (*LSTM*), and Multi-Layer Perceptron, for the purpose of forecasting outbound demand. A comparative analysis of a deep learning model in the retail domain [17] that employs from the retailer data, implements Long Short-Term Memory (*LSTM*) with peephole associations and conventional tree-based models. This study [18] examined the efficacy of various multi horizon forecasting techniques using synthetic datasets with varying parameters, including the size of the time series and the number of horizons. The DeepAR model [19] is suggested that utilizes a predefined distribution to generate probabilistic forecasts for time-series data. DeepAR utilizes multi-layer perceptrons (MLPs) to estimate the distribution parameters at any given point, to generate probability density functions for the target variables. The assumption of distributional homogeneity is frequently too stringent to apply to datasets in real-world scenarios. The *LSTM* encoder-decoder model with position-based attention captures the trends of periods in sequential data. The attention mechanism was utilized to examine analogous local patterns in past data for the purpose of making predictions about the future. Examining the complete historical record of time series is not feasible, and determining which portion of the past history data to focus on depends on human expertise.

## 3 METHODOLOGY

### 3.1 Demand Patterns Classification: Light GBM Model

Xgboost and lightGBM are frequently employed in GBM frameworks. LightGBM is the decision tree-based distributed gradient boosting framework which is a common algorithm of machine learning [21]. LightGBM is very suitable for constructing sporadic classification models that can efficiently manage large-scale supply chain datasets due to its efficiency, scalability, support for varied data types, and model interpretability. This framework helps firms analyse large scale supply chain data and make smart decisions to optimize operations, reduce risks, and boost performance. The Leaf Splitting technique used by LGBM and LGBM utilizing a Histogram-based approach is shown in Fig. 1 and Fig. 2.

LightGBM uses a decision tree algorithm based on a Histogram, dividing features into small bins, which can reduce storage and computational costs by minimizing data quantity per leaf node. LightGBM enables technique to control and optimization through the use of specific parameters.

*num_leaves:* Quantity of leaf nodes in each tree.
*learningrate:* Parameter used in machine learning process.
*min_data_in_leaf:* Minimal number of data points allowed in a leaf node.
*feature_fraction:* Regulates proportion of selected features.
*bagging_fraction:* Ratio of a subset of data to overall data.
*metric:* Mathematical function measuring distance or similarity between two objects.

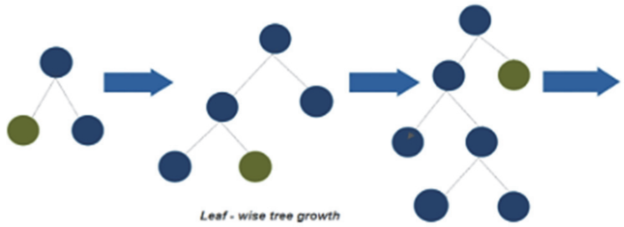

**Figure 1** The leaf splitting technique used by LGBM

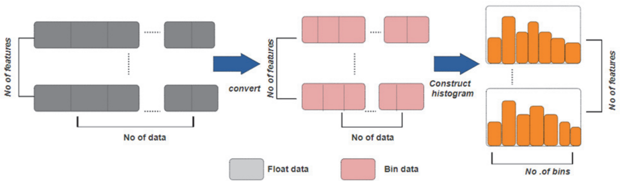

**Figure 2** LGBM utilizing a histogram-based approach

### 3.2 Demand size prediction (Regression): Temporal Attention based *LSTM* Encoder-Decoder Structure for Multi-horizon Forecasting

#### 3.2.1 Sequence to Sequence using *LSTM*

Time series data on sales, stocks, and transit volumes are commonly found in extensive supply chain datasets. Seq2Seq models are well-suited for the analysis of sequential data due to their ability to capture temporal dependencies. The *LSTM*-based encoder-decoder model is popular in machine learning and time series forecasting. It encodes past and future variables using a sequence-to-sequence learning pipeline and decodes them into future predictions. The encoder phase includes a Bi-layer *LSTM*, transforming past sequences into hidden representations and shown in Eq. (2).

$$\begin{aligned}
i_t &= \sigma\left(W_{ix}x_t + W_{im}x_{t-1}\right)\\
f_t &= \sigma\left(W_{fx}x_t + W_{fm}x_{t-1}\right)\\
o_t &= \sigma\left(W_{ox}x_t + W_{om}x_{t-1}\right)\\
C_t &= f_t C_{t-1} + i_t \tan h\left(W_{cx}x_t + W_{cm}x_{t-1}\right)\\
h_t &= o_t \tan h\left(c_t\right)
\end{aligned} \tag{2}$$

The equation is abbreviated as $h_t^E = LSTM^E\left(x_t, h_{t-1}\right)$, where the superscript $E$ denotes that the hidden states belong to the encoder. In the Eq. (3), we represent the hidden states generated during a given time t from the forward *LSTM* as $hs_t^{Forward}$ and from the backward *LSTM* as $hs_t^{Backrward}$. We also mark the concatenation of these states as $hs_t$ .

$$
\begin{aligned}
hs_t^{Forward} &= LSTM^F\left(x_t, h_{t-1}\right)\\
hs_t^{Backrward} &= LSTM^B\left(x_t, h_{t+1}\right)\\
hs_t &= \begin{bmatrix} hs_t^{Forward} \\ hs_t^{Backrward} \end{bmatrix}
\end{aligned}
\tag{3}
$$

The formulation is abbreviated as $h_t^{Decoder}$, which represents the hidden states of the decoder. It is defined as $h_t^{Decoder} = biLSTM^d\left(x_t, h_{t-1}, h_{t+1}\right)$ where $x_t$ is the input and $h_{t-1}$, $h_{t+1}$ are the previous and next hidden states, respectively. To obtain a quantile forecasts from hidden states, a linear layer is appended to the hidden states at each time step. The Eq. (4) represents the linear layer that generates K quantile predictions simultaneously, as depicted in Fig. 3. The *K* quantile forecasts for encoder and decoder stages are denoted as $qp_t^{encoder}$ and $qp_t^{decoder}$.

$$
\begin{aligned}
qp_t^{encoder} &= W_{encoder} h_t^{encoder} + b_{encoder}\\
qp_t^{decoder} &= W_{decoder} h_t^{decoder} + b_{decoder}
\end{aligned}
\tag{4}
$$

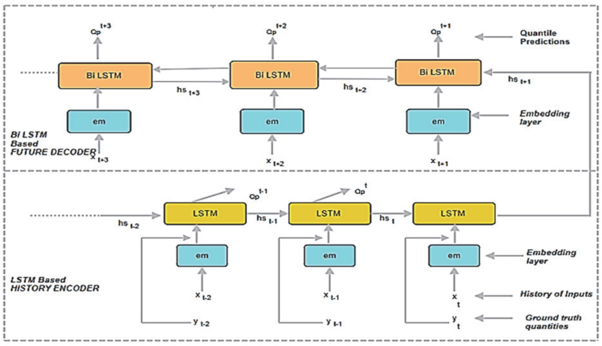

**Figure 3** The architecture of the encoder decoder model is based on long short-term memory (*LSTM*)

### 3.2.2 Feature Learning using Embedding

Embedding is a technique that converts categorical data into quantifiable vectors of features, such as word embedding. It helps analyze external factors and self-regulating characteristics in retail environments. Embedding layers in models allow simultaneous learning of time series-related features and encoding categorical features like store IDs, holidays, weather, and geography represented in Fig. 4. Conventional encoding methods have limitations, so embedding is justified due to its compactness and density.

### 3.2.3 Multimodal Temporal Attention Mechanism

The *LSTM*-based encoder decoder model struggles to capture long-term dependencies due to its lack of memory and tendency to overwrite prior information. A Position-based Attention Model addresses this issue by capturing times series patterns in past data as in Fig. 4. However, this model experiences error accumulation when ignoring dynamic future information. The proposed architecture consists of a historical encoder, a BiLSTM Future decoder, and an attention mechanism to identify similar patterns.

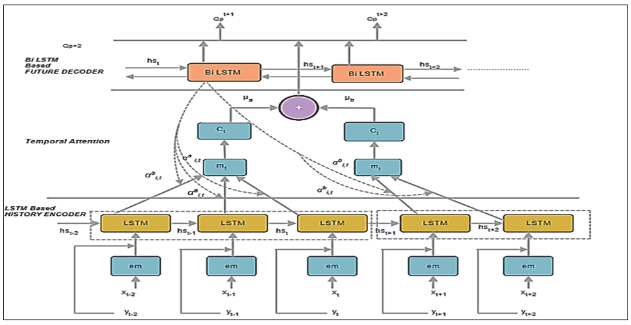

**Figure 4** Proposed multimodal temporal attention for the Seq2seq based LSTM encoder and decoder

This mechanism is led by the BiLSTM hidden states generated at each future time step *t*.

History Encoder:

The online sales dataset comprises a collection of product records that can be effectively analyzed using *LSTM* based history encoder. At each time step *t*, the input sequence can be expressed as a function of the hidden state from the previous time step and the current input. The final hidden state is represented in the Eq. (5). $h(t)_{encoder}$ is a composite function of encoded temporal information derived from the preceding hidden states $w \cdot h_{t-1}$ and previous inputs $x_{t-1}$. This can be clearly presented as seen below:

$$
h(t)_{encoder} = F\left(w \cdot h_{t-1} + x_{t-1}\right)
\tag{5}
$$

The attention weight is generated using a context vector, combining multiple attentions on past data to predict future time steps. This method is particularly useful for time series data like E-Commerce sales datasets with lengthy historical records. The decoder network can adjust the period duration to coincide with industry cycles, such as calendar months or quarters.

Every group of time step i of the encoder and the time stept of the decoder, the so called context scores $c_{i,t}$ are computed using weighted sum and it is given by the Eq. (6),

$$
c_{i,t} = F_c \tan h\left(h_t^{decoder} s(t-1) w_{encoder} \cdot h_i\right)
\tag{6}
$$

This equation involves trainable weights denoted as $F_c, h_t^{decoder}, w_{encoder}$ which are referred to as attention weights. The weights $w_{encoder}$ correspond to the hidden states of the encoder, the weights $h_t^{decoder}$ are associated to the hidden states of a decoder, and the weights $F_c$ define the function that calculates the context score. At each time step *t*, the scores $c_{i,t}$ are normalized using the softmax function across the encoder time steps *i*, resulting in the multimodal attention weights $\alpha_{i,t}$ is represented in the Eq. (7).

$$
\alpha_{i,t} = \frac{\exp \boxed{fo}\left(\alpha_{i,t}\right)}{\sum_{j=1}^{M} \exp \boxed{fo}\left(\alpha_{i,t}\right)}
\tag{7}
$$

The attention weight $\alpha_{i,t}$ represents the importance of the input at time step i in influencing the output at time step

$t$. The Eq. (8) indicates that the context vector $c_{i,t}$ is computed by taking the weighted sum of all the hidden states of the encoder, based on the attention weights.

$$c_t = \sum\nolimits_{i=1}^{T} \alpha_{i,t} h(i) \tag{8}$$

The modified output probability $m_t$ can be calculated using the Eq. (9) by attended weights generated with respect to context vector $c_t$, shown in the Eq. (9)

$$m_t = soft\max\left(w_i c_t + b_i\right) \tag{9}$$

Future Decoder: The decoder receives the context vector $c_t$ and calculates the probability distribution for the subsequent potential output. This decode process is applied to all time steps in the input. The context vector $c_t$ is given as input which helps to compute the current hidden state $h_t$ based on the recurrent unit function; the hidden state $h_{t-1}$ and output $y_{t-1}$ of the previous time step are given by Eq. (10)

$$h_t = F\left(h_{t-1}, y_{t-1}, c_t\right) \tag{10}$$

Our suggested method expands the attention-based technique to multimodal attention by incorporating the present decoder state. The decoding architecture can selectively focus on particular kinds of input modalities in order to forecast the next time step. We can utilize temporal attention on $M$ periods of historical data, where $M$ is equal to 2, as depicted in the image. Additionally, we can combine this with multi-modal attention weights $\mu_t^{1\ldots M}$ given in the Eq. (11) and Eq. (12), which are created by integrating with the prior hidden state $h_{t-1}$ along with modified content vectorsm$_\mathrm{t}$

$$Q_t^m = F_c \tan h\left(W_t^D s_{t-1} + F_c^M + m_t + b_t\right) \tag{11}$$

$$\mu_t^m = \frac{\exp\boxed{fo}\left(Q_t^m\right)}{\sum\nolimits_{i=1}^{M} \exp\boxed{fo}\left(Q_t^I\right)} \text{ For } m = 1, \ldots, M \tag{12}$$

The integrated information $x_t$ is calculated by adding the values $m_t$ with the multimodal attention weights $\mu^m$ represented in the Eq. (13)

$$x_t = \sum\nolimits_{m=1}^{M} \mu_t^m m_t \tag{13}$$

## 3.3 Evaluation Metrics for Demand Pattern Classification and Prediction

The list of Performance evaluation metrics is used in order to accurately quantify the effectiveness of demand forecasting models.

### 3.3.1 Root Mean Squared Logarithmic Error (*RMSLE*)

*RMSLE* and The Root Mean Squared weighted Logarithmic Error (*RMSWLE*) are computed by taking the logarithm of both the actual and predicted values, and then finding the difference between them. *RMSLE* and *RMSWLE* is represented in the Eq. (13) and Eq. (14) which is resistant to the influence of outliers, as it treats both major and minor errors uniformly.

$$RMSLE = \sqrt{\frac{1}{M} \sum\nolimits_{i=1}^{M} \left(\log\left(\widehat{y_i} + 1\right) - \log\left(y_i + 1\right)\right)^2} \tag{13}$$

$$RMSWLE = \sqrt{\frac{\sum\nolimits_{i=1}^{M} W_i \left(\log\left(\widehat{y_i} + 1\right) - \log\boxed{fo}\left(y_i + 1\right)\right)^2}{\sum\nolimits_{i=1}^{M} W_i}} \tag{14}$$

### 3.3.2 Mean Absolute Logarithmic Error (*MALE*)

In order to compute the Mean Absolute Logarithmic Error (*MALE*), we first determine the natural logarithm of each for the values of the prediction. The process involves determining values and computing the mean squared error, which is specifically referred to as the Mean absolute Logarithmic Error (MSLE) represented in the Eq. (15).

$$MALE = \sqrt{\frac{1}{M} \sum\nolimits_{i=1}^{M} \left|\log\left(\widehat{y_i} + 1\right) - \log\boxed{fo}\left(y_i + 1\right)\right|} \tag{15}$$

In the previous equations $\widehat{y_i}$ denotes the predicted sales of the shops, $y_i$ denotes the actual sales and $M$ denotes the total number of products.The adoption of the logarithmic component in the error measurements was motivated by the fact that products from various shops can have significantly varying demand levels

### 3.3.3 Mean Absolute Deviation (*MAD*)

We assess the performance of forecasting method using the mean absolute deviation (*MAD*) metric shown in the Eq. (16). The determination of the specific quantile for every item is based on industry logic, as well as the provision of the whole quantile forecast is highly advantageous for this reason. A decreased quantile loss signifies an improved prediction. *MAD* is calculated by summing the quantile loss for any given time related feature i for every upcoming time step t, according to the following formula:

$$Q_L^{MAD} = \sum\nolimits_i \sum\nolimits_Q \frac{\sum\nolimits_t \left[Q\left|\left(y_t^i - \left(\hat{y}_t^i\right)^Q\right)\right| + (1-Q)\left|\left(\hat{y}_t^i\right)^Q - y_t^i\right|\right]}{T} \tag{16}$$

### 3.3.4 Metrics for Classification

Furthermore, this study requires further evaluation of classification prediction, in contrast to the metrics of assessment used for regression prediction. The confusion matrix, as illustrated in Tab. 1, is the most commonly utilized tool in data mining theory to assess the predictive capability of a classification model.

The Individual evaluation metrics facilitates on Accuracy, precision, and recall whereas the detailed evaluation focuses primarily on the $F1$ score, Receiver

Operating Characteristic (ROC), and Area under the Curve ($AUC$).

**Table 1** Confusion matrix

| Actual value | Predicted Value | |
|---|---|---|
| | Positive | Negative |
| Positive | True Positive (TP) | False Negative (FN) |
| Negative | False Positive (FP) | True Negative (TN) |

## 4 PROPOSED TWO STAGE DEMAND FORECASTING FRAMEWORK

This suggested two-stage retail sales forecasting method utilizes demand pattern classification and demand prediction (Regression). The framework is implemented using LGBM and seq2seq with temporal attention for multi-horizon prediction is illustrated in Fig. 5. The suggested forecasting framework aims to offer a versatile solution for different demand patterns in retail sales supply chains. The Corporacion Favorita Grocery Sales data set is suitable for both classification and prediction objectives, specifically for demand/sales forecast goal.

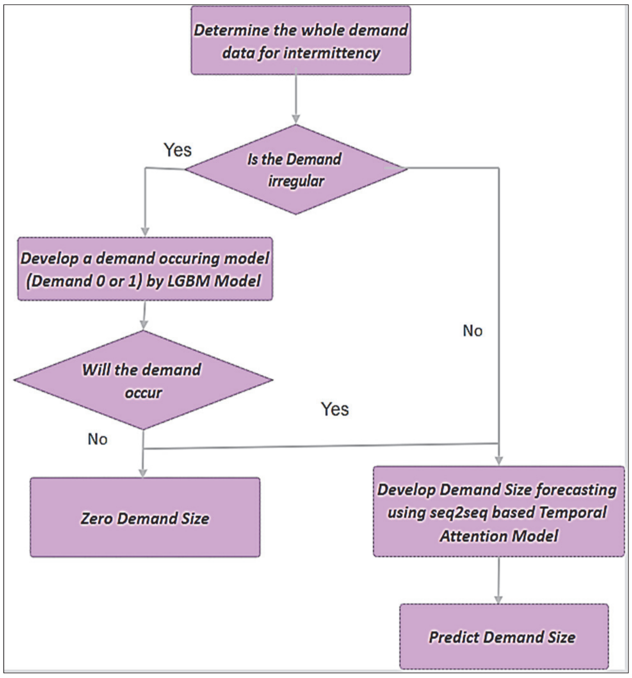
Determine the whole demand data for intermittency
Is the Demand irregular
Yes
No
Develop a demand occuring model (Demand 0 or 1) by LGBM Model
Will the demand occur
Yes
No
Zero Demand Size
Develop Demand Size forecasting using seq2seq based Temporal Attention Model
Predict Demand Size

**Figure 5** Overall proposed workflow

The proposed forecasting framework can seamlessly integrate with existing smart inventory systems, enhancing decision-making capabilities in the smart supply chain and enabling organizations to optimize inventory management, reduce stockouts, and improve overall operational efficiency.In this section, we present the combined demand forecasting approach that primarily utilizes the enumerated list of fundamental notations provided below.

General notations:

Let $x_i$ be the feature space, often known as objective (feature) space.

The class space, denoted as $c_d = \{0, 1\}$, pertains to the classification problem of identifying the existence or non-existence of demand, a crucial aspect to be defined within the scope of this study.

Let $y_i \in p$ denote the integer space.

The classification data points $C \subseteq \{x_i, c_d\}$, obtained from the feature space.

The Regression data points $R \subseteq \{x_i, y_i\}$, obtained from true category $c_{x_i} \in c_d$.

The function for mapping $F_C : x_i \rightarrow c_d$, assigns to its true category $c_{x_i} \in c_d$.

The function for mapping $F_R : x_i \rightarrow y_i$, assigns the data point to its $r_{x_i} \in y_i$.

Unlabeled Test data is denoted as $test_{x_i}$ such that $t_1 = test_{x_i}$ where $i = 1, 2, \dots, n$.

Test prediction data is denoted as $testP_{x_i}$ such that $t_2 = test2_{x_i}$ where $i = 1, 2, \dots, n$.

Validation data is denoted as $val_{x_i}$ such that $V_i = val_{x_i}$ where $i = 1, 2, \dots, n$.

Classification threshold $\tau_i \in \{0.4, 0.6\}$ such that $\tau_i = \tau_c$ where $i = 1, 2, \dots, k$.

Training data is divided for processing both classification and regression and is denoted as $train_{x_i}^{C} = \{x_i, c_{x_i}\}$ and $train_{x_i}^{R} = \{x_i, r_{x_i}\}$ where $c_{x_i}$ denotes true class sample of $x_i$ and $r_{x_i}$ denotes the sales feature sample of $x_i$.

Based on the provided notations, the forecasting framework can be implemented by Algorithm 1 in which training a classifier on the given training dataset $train_{x_i}^{C}$ and a regressor on the dataset $train_{x_i}^{R}$.

The goal is to minimize the errors of both classification and regression on the unlabeled dataset $test_{x_i}$ and $testP_{x_i}$.

Algorithm 1: Two stage retail sales forecasting (Using Classification and Regression)

```
Input: Training data $train_{x_i}^{C}$, $train_{x_i}^{R}$; unlabeled Test data $test_{x_i}$; classification learner - $C_L$ and regression Learner - $R_L$
Training: Utilize the training classification sample $train_{x_i}^{C}$; to invoke the classification approach $C_L$ and train the classification model $M_C \sim f_{x_i}^{C}$.
Apply the regression training data $train_{x_i}^{R}$ to invoke the regression approach $R_L$ and train the Regression model $M_R \sim f_{x_i}^{R}$.
Result:
For $i$ =1, 2, …, $n$
    Call $M_C : c_{x_i} = f_{x_i}^{C}$; (Classification)
    If $c_{x_i} = 0$, then $f_{x_i} = 0$ // checking for non-zero and zero demands
    else $c_{x_i} = 1$, then call $M_R : c_{x_i} = f_{x_i}^{R}$; (Regression)
end for
Output the forecast result set $F$
```

Algorithm 2: Best Demand Threshold for Demand Classification.

```
Input: Training data - $train^C_{x_i}$ , $train^R_{x_i}$ ; Unlabeled Test data- $test_{x_i}$ ; validation data - $val_{x_i}$ ; classification learner - $C_L$ and regression Learner - $R_L$; Classification threshold values - $\tau_i$
Training: Utilize the training classification sample $train^C_{x_i}$ to invoke the classification algorithm $C_L$ and to train the classification model $M_C \sim f^C_{x_i}$ to find out the probability of demand occurrence (0 or 1)
Utilize the training data points for regression $train^R_{x_i}$ to invoke the regression technique $R_L$and train the Regression model $M_R \sim f^R_{x_i}$ to find out the demand size
Finding of best demand threshold:
for $i = 1, 2, 3, ..., n$
$V_i < -val_{x_i}$
for $i = 1, 2, …, m$
    Call $M_C : D_m = f^C_{x_i}$ ;
    If $D_m > \tau_i$ then $V_i = 1$
    Else $V_i = 0$
end for
Generate the demand classification outcome and compute the area under the curve $AUC_i$ of the classification model using a threshold of $\tau_i$ as incorporating the true class of the validation data $V_i$
Output $\tau_i$ where as $i = argmax_i AUC_i$
Output:
for $i = 1, 2, …., k$
By optimal threshold $\tau_i$, call $M_C : D_m = f^C_{x_i}$
If $D_m < \tau_i$ then $f_{x_i} = 0$;
else call $M_R : c_{x_i} = f^R_{x_i}$;
Output the forecast result set $F$
```

Data from the training set is utilized to create a classification model, $M_C$, using the LightGBM machine learning algorithm. This model estimates demand occurrences. A range of classification thresholds from 0.4 to 0.6 is defined with incremental steps as shown in the Algorithm 2. Run $M_C$, on the validation set to calculate demand probability. If demand likelihood exceeds classification threshold, 1 is expected. If the demand probability is below the categorization threshold, 0 is anticipated. Systematically assess validation set classification at various thresholds. Determine the threshold with the largest Area under the Curve (*AUC*) value, indicating the best categorization. The test set should use the optimal categorization threshold from the validations that match data attributes. Once categorization predictions are made, the $M_R$ regression model estimates non-zero demand. Both sets of findings are combined to determine the final result. The suggested framework may forecast long-term supply chain dynamics and market conditions. It often gathers weeks, months, or quarters of data. Historical data can show seasonal, cyclical, and other long-term demand fluctuations. The system can dynamically update forecasting models based on incoming data to adapt to changing conditions and demand pattern shifts.

## 5 EXPERIMENTS

### 5.1 Data Sets

CorporaciónFavorita, an Ecuadorian firm that owns many supermarkets around Latin America, published this dataset in about 2017 as a Kaggle competition, with the aim of challenging the community to predict their sales. The dataset comprises daily sales data for 4400 distinct items, across 54 Ecuadorian retailers, spanning from January 1st, 2013 to August 15th, 2017. The dataset for Corporacion Favorita has 125497040 observations for training and 3370464 observations for testing as shown in Tab. 2. The datasets consist of sales data categorized by date, store number, item number, and promotion details. In addition, the system offered information on transactions, oil prices, store details, and public holidays.

**Table 2** Description about the corporacion favorita dataset

| .csv files | Features | Variable count |
|---|---|---|
| train | id, date, store.nbr, item.nbr, unit.sales, onpromotion | 6 |
| oil | Data, dcoilwtico | 2 |
| holidays | Date, type, locale, locale.name, description, transferred | 6 |
| items | item.nbr, family, class,perishable | 4 |
| stores | store.nbr, city, state, type,cluster | 5 |
| transactions | Date, store.nbr, transactions | 3 |

### 5.2 Feature Engineering

The primary conversion involved populating the zero sales data sets, since the datasets was initially lacking this information. The developers of the Kagglecompetition dataset, Corporación Favorita (2018), recommended normalizing the target variable with a logarithmic transformation. The feature engineering part is done by means of the aggregating the time series features (Discrete) and historical sales. The preprocessing of the grocery data can be done in the training data and distribution of the grocery sales is illustrated in Fig. 6. A logarithmic adjustment was employed to consolidate the grocery sale counts in order to enhance data presentation and interpretation. The analysis of sales is important for the prediction. So we preprocess and visualize the total sales for yearly, monthly and daily basis over 2013 to 2017 illustrated in Fig. 7. Then, dataset undergoes extraction based on the SBC demand categorization method introduced by Syntetos. The coefficient of variation ($CV^2$) quantifies the level of volatility in the demand for an item, whereas the Average Demand Interval (*ADI*) characterizes the demand characteristics by measuring the size of the items' demand over a specific time period

A logarithmic adjustment was employed to consolidate the grocery sale counts in order to enhance data presentation and interpretation. The analysis of sales is important for the prediction. So we preprocess and visualize the total sales for yearly, monthly and daily basis over 2013 to 2017 as illustrated in Fig. 7. Then, dataset undergoes extraction based on the SBC demand categorization method introduced by Syntetos. The coefficient of variation ($CV^2$) quantifies the level of volatility in the demand for an item, whereas the Average Demand Interval (*ADI*) characterizes the demand characteristics by measuring the size of the items' demand over a specific time period.

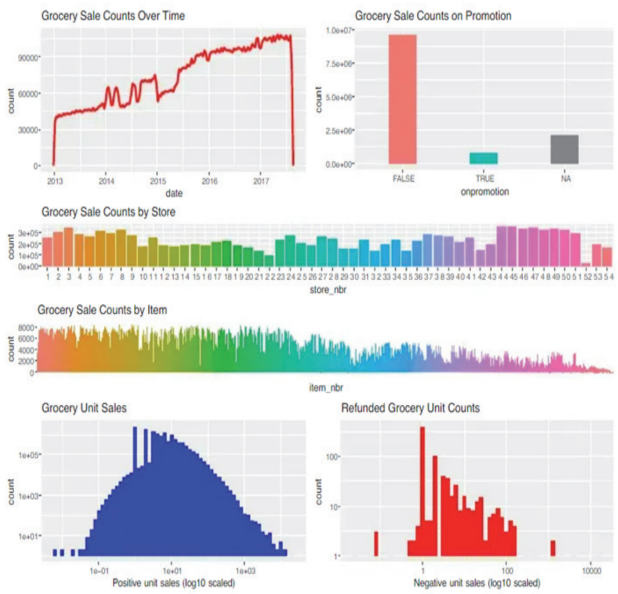

**Figure 6** Training data distribution for grocery sales

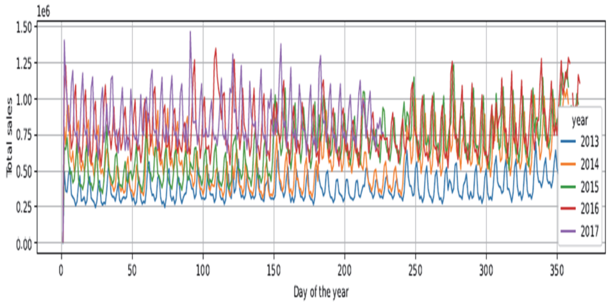

**Figure 7** Total number of sales on daily basis over 5 years

Syntetos proposes a demand classification hypothesis. The syntetos approach added a demand label to the training data to make retail sales data supervised. To convert the dataset to supervised, we utilize 0.49 and 1.32 demand thresholds. Demand 1 for smooth and irregular demand patterns, demand 0 for intermittent and lumpy demand patterns. The average demand interval (*ADI*) is 10.56, above the 1.32 criterion, suggesting significant intermittency. Tab. 3 shows that $CV^2$ is 9.95, above the 0.49 threshold. As seen in Tab. 3, most products have inconsistent prior demand, making forecast difficult.

**Table 3** SBC summary statistics of corporacion favorita dataset

| ID | Average Demand Interval (*ADI*) | $CV^2$ |
|---|---|---|
| Avg | 10.5623 e + 08 | 9.95751 e + 08 |
| Standard Deviation | 9.729693 e + 08 | 9.928462 e + 08 |
| Min | 1.254970 e + 08 | 0.751245 e + 08 |
| 25% | 1.263397 e + 08 | 1.054658 e + 08 |
| 50% | 1.271823 e + 08 | 1.504578 e + 08 |
| 75% | 1.280249 e + 08 | 1.005488 e + 08 |
| Max | 1.288675 e + 08 | 2.465846 e + 08 |

## 5.3 Baseline Methods

This study discusses various deep learning techniques for forecasting future values using hidden attributes and variables. Position-based attention models, MQ_RNN, gradient boosting, grid search, and MC_Bootstrapping are used. The effectiveness of these techniques is assessed using a service-driven inventory system. Classification Accuracy for the proposed framework and graphical illustrations is shown in Fig. 8 and the numerical results are shown in Tab. 4.

**Table 4** Classification Accuracy for 3 different time periods for the Corporación Favorita_2018

| Model | Forecasting Horizon Period - 1 | Forecasting Horizon Period - 2 | Forecasting Horizon Period - 3 |
|---|---|---|---|
| | *AUC* | | |
| Two stage Forecasting | **0.891** | **0.894** | **0.901** |
| Best threshold combination | **0.924** | **0.912** | **0.924** |
| XGBoost [21] | 0.814 | 0.824 | 0.829 |
| MC_Boot_ Strapping [19] | 0.631 | 0.628 | 0.612 |
| pos_content Attention_ RNN [7] | 0.871 | 0.864 | 0.854 |
| MQ_RNN [20] | 0.725 | 0.742 | 0.743 |

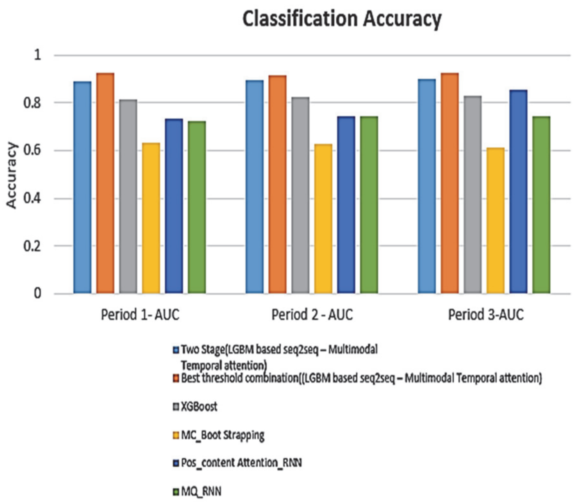

**Figure 8** A Comparative Analysis of Classification Accuracy among Two Proposed Variants with Forecast Horizons of 1, 2, and 3 Periods.

## 6 RESULTS AND DISCUSSION

The Bi-LSTM encoder decoder models have $h = 1$ and $h = 3$ variants, derived from the basic Enc_Dec model. They consist of two layers with 50 hidden state sizes and 20 embedding sizes for each categorical variable. The models use one or three periods of historical data, with Single Attention using a single period and Multimodal Temporal Attention using three consecutive periods. Two variants of a two-stage forecasting model have been developed to provide accurate predictions at various forecasting horizons shown in the Tab. 5.

Fig. 8 clearly exhibits that the first variation achieves an *AUC* accuracy of 89% across all forecasting horizons. Tab. 6 clearly exhibits that the *RMSLE*, *MALE*, and *RMSWLE* measures results lower loss errors compared to the existing forecasting techniques.

**Table 5** Proposed multimodal temporal attention prediction results compared with baseline models for corporación favorita 2018

| Techniques | | Quantile Loss (QL) | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | Avg |
| Baseline Models | Pos_content Attention_ RNN [7] | 3.43 | 3.56 | 3.72 | 2.58 | 2.45 | 3.95 |
| | MQ_RNN [20] | 2.44 | 2.53 | 2.78 | 2.87 | 1.80 | 2.48 |
| | Gradient Boosting [21] | 5.89 | 4.56 | 4.78 | 4.21 | 4.26 | 4.74 |
| | MC_Boot_ strapping [19] | 5.42 | 5.47 | 5.23 | 4.97 | 4.89 | 5.19 |
| Proposed Approach | BiLSTM _Encoder_ Decoder (h=1) | 2.22 | 2.59 | 2.41 | 1.80 | 1.36 | 2.07 |
| | BiLSTM _Encoder_ Decoder (h=3) | 2.31 | 2.39 | 2.42 | 1.92 | 1.44 | 2.10 |
| | Single Attention (h=1) | 2.45 | 2.37 | 2.46 | 1.75 | 1.38 | 2.01 |
| | Mulitmodal_A ttention (h=3) | 2.21 | 2.23 | 2.44 | 1.88 | 1.38 | **2.02** |

**Table 6** Prediction Loss for the proposed variants with state of art methods

| Model | Prediction Loss | | |
|---|---|---|---|
| | *RMSLE* | *RMSWLE* | *MALE* |
| Two stage Forecasting | **0.513 ± 0.0012** | **0.524 ± 0.0013** | **0.514 ± 0.0014** |
| Best threshold combination | **0.490 ± 0.0013** | **0.501 ± 0.0015** | **0.482 ± 0.0013** |
| XGBoost [21] | 0.768 ± 0.0024 | 0.754 ± 0.0021 | 0.744 ± 0.0021 |
| MC_Boot_ Strapping [19] | 1.574 ± 0.0004 | 1.554 ± 0.0023 | 1.345 ± 0.001 |
| pos_content_Attention RNN [7] | 0.527 ± 0.0018 | 0.548 ± 0.0018 | 0.334 ± 0.0023 |
| MQ_RNN [20] | 0.6290 ± 0.0026 | 0.631 ± 0.027 | 0.5912 ± 0.0012 |

The study uses PyTorch and a dedicated GPU server to predict daily sales for 4400 products. Network parameters are modified using the Adam solver. Inference time is evaluated using a standard GPU server with NVidia Tesla V100. The Hidden Size of *LSTM*s remains constant across models. Single-Attention takes 50.41 seconds, Multimodal-Temporal takes 90.1 seconds, and MQ_RNN has a shorter processing time of 18.7 seconds.

## 7 CONCLUSION

This paper presents a two-stage deep learning framework for forecasting intermittent demand in multiple time horizons using an LGBM-based Seq2seq Temporal Attention learning approach. The framework divides the problem into predicting demand presence and quantity, reducing inventory costs. The optimization technique prioritizes categorization, enhancing data asset utilization and accuracy. The method integrates attention learning frameworks to capture temporal contexts in historical data, achieving the highest performance with Corporación Favorita_2018. Future scope can be done by extending with External factors including market trends, weather patterns, and promotions that might improve predicting accuracy. Data validation is only done for grocery sales. Different fields can use this phenomenon to assess concept feasibility.

## 8 REFERENCES

[1] Mitra, A., Jain, A., Kishore, A., & Kumar, P. (2022). A Comparative Study of Demand Forecasting Models for a Multi-Channel Retail Company: A Novel Hybrid Machine Learning Approach. *Oper. Res. Forum, 3*(4), 58. https://doi.org/10.1007%2Fs43069-022-00166-4

[2] Dan, C., Yan, W., & Rui, F. (2022). Forecast of Large Earthquake Emergency Supplies Demand Based on PSO-BP Neural Network. *Technical Gazette, 29*(2), 561-571. https://doi.org/10.17559/TV-20211120092137

[3] Nikolopoulos, K. I. & Thomakos, D. D. (2019). Forecasting with the theta method: theory and applications. *John Wiley & Sons*. https://doi.org/10.1002/9781118445112.stat08270

[4] Lu, X. (2022). A Human Resource Demand Forecasting Method Based on Improved BP Algorithm. *Computational Intelligence and Neuroscience, 2022*, 3534840. https://doi.org/10.1155/2022/3534840

[5] Rožanec, J. M., Blaž F., & Dunja, M. (2022). Reframing demand forecasting: a two-fold approach for lumpy and intermittent demand. *Sustainability, 14*(15), 9295. https://doi.org/10.3390/su14159295

[6] Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., & Jenssen, R. (2017). An overview and comparative analysis of recurrent neural networks for short term load forecasting. *Neural and Evolutionary Computing*. https://doi.org/10.1007/978-3-319-70338-1

[7] Cinar, Y. G., Mirisaee, H., Goswami, P., Gaussier, E., Ait-Bachir, A., & Strijovet, V. (2017). Position-based content attention for time series forecasting with sequence-to-sequence RNNs. *Neural Information Processing: International Conference on Neural Information Processing*, 14-18. https://doi.org/10.1007/978-3-319-70139-4_54

[8] Yasemin, A., Tuğba, T., & Orhan, T. (2021). The Impact of Information Sharing on Different Performance Indicators in a Multi-Level Supply Chain. *Technical Gazette, 28*(6), 1960-1974. https://doi.org/10.17559/TV-20200108205821

[9] Ke, G., Guolin, K., Qi, M., Thomas, F., Taifeng, W., Wei, C., Weidong, M., Qiwei, Y., & Tie-Yan, L. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems, 30*.

[10] Kilimci, Z. H., Okay Akyuz, A., Mitat, U., Akyokus, S., Ozan Uysal, M., Bulbul, B. A., & Ekmis, M. A. (2019). An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain. *Complexity*. https://doi.org/10.1155/2019/9067367

[11] Talupula, A. (2019). Demand forecasting of outbound logistics using machine learning. *Digitala Vetenskapliga Arkivet, 36*.

[12] Helmini, S., Jihan, N, Jayasinghe, M., & Perera, S. (2019). Sales forecasting using multivariate long short term memory network models. *Peer J. Pre Prints,* 7, e27712v1. https://doi.org/10.7287/peerj.preprints.27712v1

[13] Chen, T. & Carlos, G. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acmsigkdd international conference on knowledge discovery and data mining*. 2016. https://doi.org/10.1145/2939672.2939785

[14] Deng, X., Liu, Q., Deng, Y., & Mahadevan, S. (2016). An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences, 340*, 250-261. https://doi.org/10.1016/j.ins.2016.01.033

[15] Taieb, S. B. & Amir F. A. (2015). A bias and variance analysis for multistep-ahead time series forecasting. *IEEE transactions on neural networks and learning systems, 27*(1), 62-76. https://doi.org/10.1109/TNNLS.2015.2411629

[16] Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski T. (2020). Deep AR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting, 36*(3), 1181-1191.

https://doi.org/10.1016/j.ijforecast.2019.07.001

[17] Xu, Q., Liu, X., Jiang, C., & Yu, K. (2016). Quantileautoregression neural network model with applications to evaluating value at risk. *Applied Soft Computing, 49*, 1-12. https://doi.org/10.1016/j.asoc.2016.08.003

[18] Türkmen, A. C., Januschowski, T., Wang, Y., & Cemgil, A. T. (2021). Forecasting intermittent and sparse time series: A unified probabilistic framework via deep renewal processes. *PLoS One, 16*(11), e0259764. https://doi.org/10.1371%2Fjournal.pone.0259764

[19] Hasni, M., Aguir, M. S., Babai, M. Z., Jemai, Z. (2019). On the performance of adjusted bootstrapping methods for intermittent demand forecasting. *International Journal of Production Economics, 216*, 145-153. https://doi.org/10.1016/j.ijpe.2019.04.005

[20] Wen, R., Torkkola, K., Narayanaswamy, B., & Madeka, D. (2017). A Multi-Horizon Quantile Recurrent Forecaster. *31st Conference on Neural Information Processing Systems.*

[21] Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics, 29*(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

**Contact information:**

**Praveena S.**
(Corresponding Author)
Department of Computer Science and Engineering,
College of Engineering and Technology,
SRM Institute of Science and Technology, Vadapalani Campus,
Chennai - 600026, Tamil Nadu, India
E-mail: ps4851srmist@outlook.com

**Prasanna Devi S.**
Department of Computer Science and Engineering,
College of Engineering and Technology,
SRM Institute of Science and Technology, Vadapalani Campus,
Chennai - 600026, Tamil Nadu, India
E-mail: hod.cse.vdp@srmist.edu.in