A Hybrid Deep Learning Based Deep Prophet Memory Neural Network Approach for Seasonal Items Demand Forecasting

Praveena S * and Prasanna Devi S

Department of Computer Science and Engineering, College of Engineering and Technology,
SRM Institute of Science and Technology, Vadapalani Campus, Chennai, India
Email: ps4851@srmist.edu.in (P.S.); hod.cse.vdp@srmist.edu.in (P.D.S.)
*Corresponding author

***Abstract*—Accurate sales forecasting is essential for any successful retail company in the competitive environment we live in today, where sales are of utmost importance to companies. By limiting overstock and preventing overproduction, it may aid in inventory management. Future sales are affected by a number of significant variables. A retail store's overall sales trends or the sales of a particular product may be examined to determine these aspects. With the use of temporal, historical, trend and seasonal data, this study develops a deep learning-based Deep Prophet Memory Neural Network (DPMNN) forecasting approach. Using M5 Forecasting and Predict Future Sales datasets in a Python context, the built system is used and evaluated. Extensive testing and comparisons to state-of-the-art research show that the suggested demand forecasting method achieves notable outcomes by obtaining lower Root Mean Squared Error (RMSE) and Mean Squared Error (MSE) rate.**

***Keywords*—demand forecasting, deep prophet memory neural network, linear clipping data normalization, bivariate wrapper forward elimination, sequential Bayesian inference optimization**

I. INTRODUCTION

Today, retail businesses all around the world face intense levels of competition. It is expected that the use of big data and predictive analytics will continue to expand as businesses turn to them in order to develop profitable strategies [1]. As a result, developing viable supply chain strategies is essential for facilitating market sustainability, enhancing customer relationship management, and boosting operational efficiency. Supply Chain Management (SCM) relies heavily on inventory control, which entails keeping tabs on stock levels, the number of products ready for sale, product orders, supplier deliveries, and customer fulfilment. But Out-of-Stock and Overstock are the two major issues that beset inventory management [2]. Having things that customers need temporarily unavailable is known as "Out-of-Stock", and it may negatively impact service quality. That might cause consumers to defect to the competition. The same can be said about Overstocking, which also has its share of problems. It causes a rise in operational and labor expenditures as well as a shortage of storage space and, depending on the nature of the goods being stored, a decline in product quality, all of which contribute to dwindling profits. Therefore, in order to resolve these challenges, it is crucial to have well established inventory management methods in place. The end result of this procedure should be a boost in operational efficiency and the end of needless stockpiling.

For many retails inventory management decision-making processes, information from demand forecasting systems is crucial. Many stores, especially those operating only online, have increased their investment in demand predicting technology in recent years. A retailer's success can't be guaranteed without accurate demand forecasting tactics. Supply chain management, in particular, benefits greatly from an accurate and practical demand forecasting system, which may help businesses stand out from the competition by enhancing their customer service and decreasing the costs associated with supply-demand mismatches. Additionally, the SCM process relies heavily on this factor. Using analytics to estimate demand fluctuations, businesses may improve stock projections by combining multiple factors, such as sales histories, temporal, historical, trend, and seasonal data. It is a well-studied topic to determine how best to predict future demand for a certain product. Several cutting-edge methods for predicting time series are widely used in this area, including Autoregressive Moving Average model (ARMA), Autoregressive Integrated Moving Average model (ARIMA), Seasonal Autoregressive Integrated Moving Average model (SARIMA), etc. Patterns in the past data are analyzed by these models, and the results are used to predict consumer demand. Recently, however, a number of machine learning methods like K-Nearest Neighbors, Gaussian Naive Bayes, Decision Tree, etc. have been used for this purpose. Recurrent Neural Networks (RNNs), Long Short-Term Memories (LSTMs), Gated Recurrent Units (GRUs), Auto Encoders (Encoders), and Convolutional Neural Networks (CNNs) are just a few

Manuscript received September 19, 2023; revised November 10, 2023; accepted November 27, 2023; published June 20, 2024.

of the deep learning methods that are gaining popularity. It turns out that deep learning methods do work, and their forecasts are far more reliable than those of any other method. In addition, several other Ensemble learning methods have been shown to provide reliable predictions and useful outcomes, such as a decrease in stock-outs and stock-days via fewer missed sales, which in turn increases an enterprise's total income. This study explores the use of a Deep Prophet Memory Neural Network, a deep learning-based model, to predict seasonal product sales.

Here are some of the study's most important findings:

- To apply forecasting techniques to predict the future sales.
- We apply optimization models for improving the process of feature selection.
- The results are compared using accuracy measurement methods such as Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE).

The rest of this document is outlined as follows. In Section II, a literature review is provided which discusses different approaches for forecasting. Section III describes the process of methodology involved for sales demand. In Section IV, the performances of different approaches are reported, and the results are compared. Finally, Section V is dedicated to the conclusions and future work.

II. Related Works

The increased need for footwear such as boots in the winter is an example of a product whose demand varies with the seasons. Multivariate ARIMA was used well in [3] to forecast demand for perishable goods. Recently, a neural network was utilized in addition to ARIMA for traffic prediction in networks [4]. Tandon *et al.* [5] used the ARIMA model to forecast the value of cryptocurrencies while accounting for the impact of social media. The SARIMA model is another common tool for forecasting. Predictions concerning tourist interest and traffic patterns are examples of how this strategy has been successfully implemented. Nonetheless, a number of authors have cautioned that SARIMA's linear structure and inability to distinguish nonlinear and highly volatile patterns may hamper prediction. In Ref. [6], a whole new method for making predictions called Seasonal Support Vector Regression (SSVR) is introduced. The results showed that SSVR is superior than SARIMA for forecasting seasonal products.

There is a wide range of flavors of exponential smoothing methods. Triple exponential smoothing (or the Holt-Winters (HW) technique) was employed by Tratar *et al.* [7] to account for seasonality in the dataset. Exponential smoothing has been shown to be a useful tool in the field of forecasting. The extremely variable daily sales of a supermarket were predicted using a seasonal exponential smoothing method in [8]. Several probes have made use of more advanced techniques. Li *et al.* [9] offered a Greedy Aggregation Decomposition (GAD) strategy for clothing stores to use when their daily SKU requirements exceed their available storage space. In Ref. [10], we see a variety of forecasting techniques at work. Various tried-and-true techniques for forecasting are used, such as Seasonal Autoregressive Integrated Moving Average (SARIMA) and Triple Exponential Smoothing. Next, we use state-of-the-art resources including Convolutional Neural Networks (CNNs), Long Short-Term Memories (LSTMs), and Prophets.

Since the Prophet method was just recently deployed as one of the forecasting approaches at the Facebook company [11], its full potential has not yet been exploited. The results of a study by Papacharalampous *et al.* [12] comparing the accuracy of the Prophet's weather predictions to those of traditional methods were quite close. Another research on predicting hospital discharge volumes found better accuracy than the SARIMA model [13]. Positive results were also shown when using a Prophet time series model to predict website traffic [14]. Predictions for coronavirus disease in Indonesia were made using ARIMA and Prophet [15]. Artificial neural networks are another method of prediction that has been studied and compared to the more traditional methods of predicting retail sales [16]. Aggregate time-series, which are large collections of time-series integrated using several approaches, provide more precise forecasts than disaggregate time-series [2]. Verstraete *et al.* [17] developed an ANN-based system to estimate the impact of weather uncertainty on retail goods in the near and far future. Zekic-Susac *et al.* [18] used ANN with other methods, like as regression trees, to predict the price of energy for public buildings.

Swarm intelligence and neural networks are only two examples of bio-inspired algorithms addressed in [19]. An extensive literature review of Swarm intelligence was undertaken in [20]. Applications and potential future directions for the research were discussed. Kar and Dwivedi [21] go into the advantages of employing big data. Both future paths and research requirements were provided in this work. Kaneko and Yada [22] detail a model that uses deep learning to predict a store's income. They trained and tested their system on three years' worth of grocery Point of Sale (POS) data. Their results suggest that deep learning is preferable than logistic regression. In Ref. [23], a novel method is shown for applying deep learning algorithms to assign ratings to products based on their massive number of online reviews. Another method that has gained popularity in recent years is the Recurrent Neural Network (RNN). To recall past events, RNNs rely on a network structure composed mostly of loops. Thus, they may out to be highly useful for time series forecasting [24]. One sort of RNN, called a Long Short-Term Memory (LSTM) method, may help with memory problems in the near future. Anomaly detection in time series using LSTM is described in [25]. Predicting sales on the M5 Kaggle dataset with the use of deep learning and statistical learning is the focus of [26]. Lakshmanan *et al.* [27] compare the LSTM model to various machine learning algorithms for demand forecasting and demonstrate its use in predicting market sales.

III. Problem Statement

The complex nature of consumer buying patterns during holidays, promotions, and other seasonal events make it difficult for traditional approaches to estimate sales with any degree of accuracy. Retail sales can be significantly impacted by natural catastrophes, financial crises, and other unanticipated occurrences, but traditional models cannot account for these outside influences well enough. These unanticipated occurrences can be considered as exogenous variables for seasonal demand prediction and also Conventional models frequently rely on human feature engineering, which might not be able to account for all pertinent variables influencing sales.

The aim of this study is to perform a forecasting on a seasonal item with the help of deep learning based DPMNN model. This model overcomes all the limitations has been discussed above and performing reliable prediction with history of sales. The process of forecasting is demonstrated in Fig. 1.

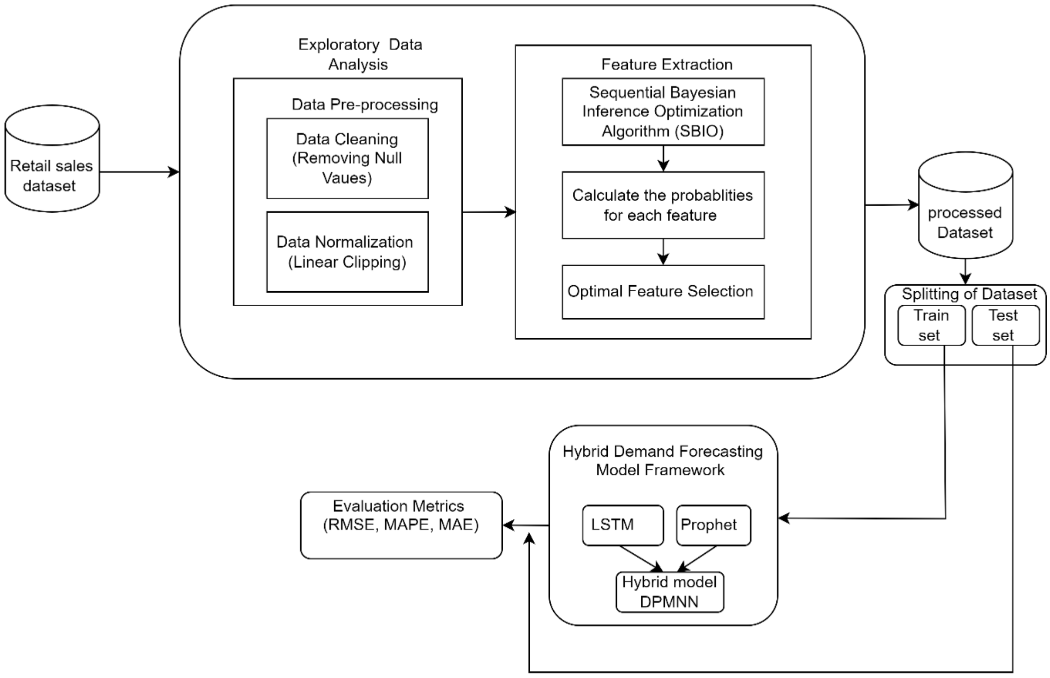

Fig. 1. Schematic representation of the suggested methodology.

A. Data Source

1) M5 forecasting datasets

Walmart's M5 data set compiles historical unit sales data for thousands of products sold in the United States. Included are sales data in the form of units for 3049 products, which are sorted into the three main groups (Hobbies, Foods, and Household) and seven subgroups (Product Departments). The 10 stores are spread among the states of California, Texas, and Wisconsin. The most detailed information, such as product-store unit sales, may be aggregated in two distinct ways: geographically (by shop and state) and thematically (by department and category). For clarity, Walmart picked the states and stores to represent a wide range of selling locations with different characteristics, consumer habits, and competitive dynamics. The selected product categories and sub-divisions included both fast- and slow-moving products, as well as consumables and durables. Explanatory variables may include but are not limited to price, discounts, sales, time of week, and holidays. This massive dataset has the potential to improve forecast accuracy. From the Kaggle competition page at https://www.kaggle.com/competitions/m5- forecasting- accuracy/data, https://github.com/KunalArora/kaggle-m5-forecasting, we were able to get the dataset used in our analysis.

The training data comes in the shape of 3 separate files:

- sales_train.csv: This dataset serves as our primary training data. The dataset consists of a single column representing each of the 1,941 days between January 29, 2011 and May 22, 2016, excluding the validation period of 28 days leading up to June 19, 2016. Additionally, the dataset has the unique identifiers for many entities, such as item, department, category, shop, and state. The total number of rows in the dataset is 30,490, representing all possible combinations of 30,490 goods across 10 different retailers.
- sell_prices.csv: The data includes the shop and item identification numbers, as well as the average weekly sales price of the item.
- calendar.csv: The dataset includes dates, together with associated attributes such as the day of the week, month, year, and three binary indicators representing whether retailers in each state permitted purchases using SNAP food stamps on a given day (1) or did not permit them (0).

2) Predict future sales dataset

One of the major Russian software companies, 1C Company, has generously supplied a series dataset comprising of daily sales data from the following link, we are downloaded the dataset https://www.kaggle.com/competitions/competitive-data-science-predict-future-sales.

B. Exploratary Data Analysis (EDA)

Each product's sales records from the past were chosen. This process of data preparation is essential for removing outliers, seeing patterns, and understanding seasonality. As a result, we experimented with linear clipping data normalization to try to smooth out the noise and substitute missing data for days with missing sale numbers. Eq. (1) uses the missing value, denoted by $\hat{K}_{m_k t}$, and the accessible data, denoted by $Y_{m_k N}$, for the same day in earlier years.

$$\hat{K}_{m_k t} = \frac{1}{N}\left(K_{m_k 1} + K_{m_k 2} + \cdots + K_{m_k N}\right) \quad (1)$$

We have utilized time series decomposition to identify the jumbled data. In the same way that missing data was filled in, the data was smoothed if the recognized noise was more (or lower) than the average plus two standard deviations of sale and did not occur on promotion days or holidays.

Stationarity is a crucial feature of time series data. If a time series' statistical characteristics, such as its mean or standard deviation, remain constant over time, it is said to be stationary. This holds significance because it has the strong likelihood that the time series will repeat its past behavior in the future, making it simpler to forecast. The Dickey Fuller Test determines stationarity in time series data by examining unit roots. This work incorporates effective Exploratory Data Analysis (EDA) technique to make the non-stationarity time series into stationarity by invoking seasonal differencing and Log transformation methods.

Statistical methods may be used to impute data, or estimate missing values in a dataset. The method analyses the correlation between several explanatory factors and a single dependent variable. The goal of Life Cycle Data Network (LCDN) is to discover missing coefficients $\{x_j, K_j\}, x_j \in K^{n-1}, X_j \in K, j = 1, \dots, k$ within clusters in order to minimize the overall fit, given a data set A, can be represented mathematically as,

$$A = \{(a_i, C_i) \in L^{n-1} \times L \mid i = 1, \dots, m\}\text{Clusters } \emptyset \neq C^j \subset C, j = 1, \dots, k \quad (2)$$

where, let $\{x_j, K_j\}$ denote the missing data coefficients computed from the available sample data $A^j, j = 1, \dots, k$. The squared error rate $E_{\boldsymbol{ab}}(x_j, K_j)$ is thus defined for a given point $(a, b) \in A$ and coefficients $\{K_j, X_j\}$.

$$E_{ab}(x_j, K_j) = \left(K_j^T X C + X_j - b\right)^2 \quad (3)$$

$$f_k(x_j, K_j) = \sum_{i=1}^{m} \min_{j=1,\dots,k} E_{ab}(x_j, K_j) \quad (4)$$

The following formula is known as the k^{th} LCDN function (global fit formula). Here $K = (K_1, \dots, K_k) \in K^{(n-1)k}$ and $\boldsymbol{y} = (K_1, \dots, K_k) \in \mathbb{R}^k$. LCDN problem formulation is provided by

$$\begin{cases} \text{minimize} & f_k(x, K) \\ \text{subject to} & K = (K_1, \dots, K_k) \in JK^{(n-1)k}, y \in K^k. \end{cases} \quad (5)$$

LCDN is a method that fits polynomial functions of the independent variables. Subtle relationships between the variables may be captured by these polynomial functions. For an impute $(B_i, A_i) \in \mathbb{R}^n, i = 1, \dots, m$ with a missing value in b_i we compute

$$z_j = x_j^T B_i + K_j,\ j = 1, \dots, k \quad (6)$$

The error weight w_j is computed as

$$w_j = \sum_{h \in \mathcal{C}_j} \frac{Z - rmse_{hi}}{(l-1)\overline{rmse}} \quad (7)$$

where $\overline{rmse} = \sum_{h \in \mathcal{C}_{nn}} rmse_{hi}$, and $rmse_{hi}$ is the RMSE between a_i and a_h. Here, we set $V_j = l_j/l$ if $\overline{rmse} = 0$. If $l = 1$, we simply take the nearest neighbor (the one with the smallest $rmse_{hi}$) and the cluster j^* it belongs to, and we set $V_{j^*} = 1$ and $V_j = 0$ for all $j \neq j^*$.

We can now get the imputed values by

$$b_i^{imp} = \sum_{j=1}^{k} V_j z_j,\ i = 1, \dots, m. \quad (8)$$

To account for the inherent inconsistency in the data, LCDN incorporates missing data precisely and data can be standardized.

C. Feature Extraction

Improving prediction quality through feature input selection is the fundamental goal of feature extraction and selection. To that end, it is necessary to create a new training set before training the model, one that corresponds to the output vector that will be used to make the prediction. The following stages illustrate the recommended technique for the bivariate wrapper forward elimination based on the mutual information.

Step 1: Create the first set of sample pairs for training in the sequence specified by the model. It is possible to display the input-output pairs as

$$S_o = Y, Z. \quad (9)$$

Step 2: Find the MI values between the input and output variables. According to Eq. (9), $MI(Y_i, Z)$ (the number of the original input series). If there are any numbers in vector M that are manifestly fewer than others, take them out and keep the rest. S_{new} is a representation of the input-output pairs. Arrange S_{new} in decreasing order of mutual information values from vector M. The degree of dependency between two random variables is quantified by their mutual information. If we have two independent random variables, X and Y, then MI is a measure of how much information there is between them. To be more precise, MI assesses how much information about one variable may be used to predict another. If we write the joint *probability* density function for X and Y as $\mu_{Y,Z}(Y, Z),\ \mu_Y(Y)$, then the marginal densities for Y and Z are $Z(y) \cdot \mu_X(Y) = \int \mu(Y, Z) dy$ and $\mu_Z(Z) = \int \mu(Y, Z) dZ$. According to Shannon's formulation, the uncertainty on X is given by Eq. (10).

$$H(YZ) = -\int \mu_Y(Y)\ log\ \mu_Z(Z) dZ \quad (10)$$

By calculating the joint entropy of two random variables (X and Y), we may get an idea of how much information is shared between them. Below is a definition of the joint wrapper entropy:

$$H(YZ) = -\iint \mu_{YZ}(x,y)\, log\, \mu_{YZ}(YZ) dydz \quad (11)$$

An expression for the MI between X and Y, where X and Y are random variables, is:

$$I(YZ) = \iint \mu_{zY}(z,y)\, log\, \frac{\mu_{zY}(YZ)}{\mu_z(z)\mu_Y(y)} dzdy \quad (12)$$

This is the same as the Kullback-Leibler separation between the product of the marginal distributions of z and Y and the joint probability density of the two variables. The degree of dependence between z and Y may be quantified using the MI. A high value of mutual information indicates that z and Y are correlated. This is significant since it implies that z has data relevant to Y. This finding will indicate that z might be an important factor. Therefore, the joint Probability Density Function (PDF) between z and Y must be estimated in order to estimate the MI between them.

$$I(z,Y) = H(z) + H(Y) - H(z,Y) \quad (13)$$

The MI may be expanded to include m-dimensional variables:

$$\begin{aligned} I(z_1, z_2, \cdots, z_m) = \quad & H(z_1) + H(z_2) + \cdots + H(z_m) \\ & + H(z_1, z_2, \cdots, z_m) \end{aligned} \quad (14)$$

Therefore, MI may be used to evaluate the degree of dependence among a set of random variables $(z_1, z_2, \dots, z_m)$.

Step 3: Identify n, the total number of characteristics that were extracted. Pick the first n input-output pairs from T_{new} to use as features in your represented set T_n.

Step 4: Take the variable with the highest MI value and put it first in set T. The $MI(T, X_j)$ values are calculated by picking the j th variable ($1 \le j \le n$) from T_n. Consider the value of α as a criterion. Z_j is a superfluous variable that may be omitted if $MI(T, X_j) \le \alpha, T = T \cup X_j$.

Step 5: Iterate Steps 3 and 4 until all of the desired attributes have been accounted for, including those relating to revenue, pricing, and lag extraction and their relationships. We calculated the difference between the aforementioned lags of numerical characteristics derived above and included it as an additional feature.

D. Feature Selection

SBIO refers to a class of optimization techniques that use machine learning. Finding the minimum of a well-specialized feature function specified over an imputed data set is a general optimization issue that this method addresses.

$$f(z): Z \to \mathbb{R}, Z \subseteq \mathbb{R}^n, z_m = \underset{z \in Z}{argmin} f(z) \quad (15)$$

A Gaussian Process is used as a surrogate function to mimic the goal function, and an acquisition function is used to choose the next location to sample from. The following are some examples of SBIO:

Step 1: To construct the demand model, one must first do research on the nature of a certain retail business, analyzing the indices in Eqs. (16) and (17).

Step 2: Produce a sample data set of sales figures using the demand model.

Step 3: Using the SBIO feature selection approach, you may collect the optimal feature subset.

Step 3.1: Randomize the Bayesian variable's starting value and encipher the issue.

Step 3.2: Use the lookup table to rearrange the samples before calculating the RMSE. specified a sample of $z_{1:k}$and a function $f(z_{1:k})$, it estimates a set of functions across a specified domain, whose mean is μ(z) and whose variance is σ(z).

$$\begin{gathered} f(z) \mid f(z_{1:k}) \sim Normal(\mu(z), \sigma(z)) \\ \mu(z) = k(z, z_{1:k})^T k(z_{1:k}, z_{1:k})^{-1} f(z_{1:k}) \\ RMSE\sigma(z) = k(z,x) - k(z, z_{1:k})^T k(z_{1:k}, z_{1:k})^{-1} k(z, z_{1:k}) \end{gathered} \quad (16)$$

where $k(z_{1:k}, z_{1:k})$ kernel $k(z_i, z_j)$ is a semi-positive function that defines each element of the covariance matrix. The Gaussian kernel is a widely used example of the kernel function's many useful features. $k(z_i, z_j) = exp\left((-z_i - z_j^2)/a\right)$.

Step 3.3: The RMSE index is used to determine the fitness function.

$$\begin{gathered} z_n = \underset{z \in Z}{argmin} a(z) \\ a(z) Fitness = \mu(z) + \kappa\sigma(z) \end{gathered} \quad (17)$$

Step 3.4: Swap out the outmoded variables with the new ones.

Step 3.5: If the maximum generation has been achieved, the operation terminates and the feature subset represented by the best chromosome is output.

Step 4: Create a forecasting model after training a using the optimal feature subset.

E. Demand Prediction

The demand value for this product at the retailer level is derived by the demand function described in Eq. (1), assuming that the total planning horizon is nT (the review periods for both the general dealer and the retailer are T days). It presupposes that there is unpredictability and seasonality in the demand for this product at the consumer level (across all brands). Additionally, the spring festival and the summer season represent two annual demand peaks. DPMNN was used to forecast consumer interest. There are three components to a forecast: the dependent variables y_a, the independent variables (X), and the time horizon h_b. Distribute to X^L the variables in X where the optimal time lag is less than or equal to h_b. In order to make projections, the proposed model separates the time series into its trend (O_t), seasonality (s_t), and temporal and past (h_t), as shown in Eq. (18).

$$y_t = O_t + s_t + h_t + \epsilon_t \quad (18)$$

where ϵ_t is the residual at time t after decomposition. O_t and s_t include log transformations for multiplicative seasonality.

The proposed network model is built by combining the layers. They are made up of a memory cell and one of three different kinds of gates used to control the flow of data. There are three primary gates: the input, the output, and the forget. Input activation vector i_t, output activation vector o_t, and forget activation vector f_t are all associated with these gates. Information about what time t is kept in the memory state vector c_t is determined by the first gate and a second gate c_t^*. The data stored in a memory cell in time period t–1 is either retained or removed based on the decisions made by the forget gate. The memory cell's final output is determined by the information selected by the output gate.

$$f_t = \sigma_{LR}\left(V_f Y_t + U_f h_{t-1} + b_f\right) \tag{19}$$

$$i_t = \sigma_{LR}(V_i Y_t + U_i h_{t-1} + b_i) \tag{20}$$

$$o_t = \sigma_L(V_0 Y_t + U_0 h_{t-1} + b_0) \tag{21}$$

$$c_t^* = V_c Y_t + U_c h_{t-1} + b_c \tag{22}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \sigma_h c_t^* \tag{23}$$

$$h_t = o_t \odot \sigma_h(c_t) \tag{24}$$

where Y_t is a vector that is fed into the LSTM unit, h_t is the vector that is output from the LSTM unit, $\sigma_{LR}$is the Leaky Rectified Linear Unit (LeakyReLU) activation function.

The trend, O_t, may be thought of as a nonlinear function with logistic growth that has reached its maximum. Thus, it's a piecewise-linear function.

$$O_t = (k + a_t^T \delta)t + m + a_t^T \gamma \tag{25}$$

where k is the standard expansion rate and m is the offset. A changepoint is established at time s_j, where j is an integer from 1 to S, for each of the S shifts in the trend. Therefore, we create a vector of growth rate corrections, $\delta \in \mathbb{R}^S$, where δ_j represents the rate of change at instant s_j. The interest rate at any given moment t is equal to k plus the sum of all adjustments up to that point in time t: $k + \sum_{j:t>s_j} \delta_j$. To clarify, let's say we want to create a vector $a_t \in \{0,1\}^S$such that:

$$a_{j,t} = \begin{cases} 1, \text{if } t \geq s_j \\ 0, \text{otherwise} \end{cases} \tag{26}$$

So, at instant t, the rate is $k + a_t^T \delta$. When k is modified, m must likewise be modified to maintain continuity between the segment's terminal positions. A vector γ is also used to modify m, using the components$\gamma_j = -s_j \delta_j$.

By imposing a sparse prior on, the transition points s_j are chosen automatically. DPMNN employs the prior $\delta_j \sim$ Laplace $(0, \tau)$ to choose the best candidates for the transitions, where the range of allowed changes in the model's rate is specified by the parameter. In tabular supervised contexts, additional regressors like Cases and Vax are considered as supporting terms, as in Eq. (25). Predicting the target variables at a future date requires knowing the additional regressors at that date.

The Fourier series representation of the smoothed timely seasonality is provided by:

$$s_t = \sum_{e=1}^{E} \left(a_e cos \left(\frac{2\pi et}{P}\right) + b_e sin \left(\frac{2\pi et}{P}\right)\right) \tag{27}$$

where P = 7 is the typical period for the reported seasonal time series and E is a smoothing parameter that controls how fast seasonal patterns are fitted. As a result, there is a higher chance of overfitting when E is increased. Estimating the 2E parameters is also required for seasonality fitting hence, $\beta = [a_1, b_1, \ldots, a_E, b_E]^T$.

The demand for this product may be expressed as a Eq. (27) that takes into account the price, the amount of shelf space available at a certain store, and the promotional influencing factor.

$$M_{it} = \left[\lambda \cdot a_i \cdot P_{it}^{-b_i} + (1-\lambda) \cdot \alpha_i \cdot X_i^{\beta_i}\right] + S_{it} + \gamma_{it} \cdot G_{it} - \theta_i \cdot \sum_{j(j\neq i)} \gamma_{jt} G_{jt} + n_i \times rand\ (\cdot)$$

$$(i = 1, 2, \ldots, m; t = 1, 2, \ldots, n) \tag{28}$$

where R_t is the retail period and D_{it} is the seasonal demand for product it. The i^{th} product's seasonal component during the i^{th} period is denoted by S_{it}. The cost of the i^{th} item in the time period is denoted by P_{it}. Scale parameter a_i and price elasticity factor $b_i(a_i > 0, b_i > 1)$are represented as follows. Shelf location Xi indicates the retail location of the i^{th} product. The scale parameter i and elastic factor of shelf-space $\beta_i(\alpha_i > 0{,}0 < \beta_i < 1)$ are denoted by α_i and γ_{it}, respectively; G_{it} indicates whether the retailer engages in sales promotion for the ith product in the t period, with G_{it} equal to 1 if the retailer does so, and 0 otherwise. The i^{th} product's i^{th} period promotion-caused demand increase is denoted by. Clearly, the values of $\gamma_{it^*}, \gamma_{i(t^*+1)}, \ldots, \gamma_{i(t^*+\tau)}$ are different when the starting time of promotion is different. If the merchant starts the promotion at t = t* period, $G_{it^*} = G_{i(t^*+1)} = \cdots = G_{i(t^*+\tau)} = 1$, others are 0. Where denotes the number of weeks the promotion lasts. $\sum_{j(j\neq i)} \gamma_{jt} G_{jt}$. The promotion-induced uptick in sales of competing brands, denoted by G_{jt}, has a negative influence on the sales of the i^{th} product, denoted by θ_i. The constant n_i is the demand deviation factor, and rand $(\cdot)$ is a random number generator that produces values with a known distribution.

Let

$$\rho_i = \begin{cases} 1, \text{ if } \mathrm{i^{th}} \text{ product is selected} \\ \quad 0, \text{otherwise} \end{cases} \tag{29}$$

$\pi_{ij}\left(0 \leq \pi_{ij} \leq 1\right)$ if brand j is not available, the percentage of brand j consumers who would have purchased brand i instead shows the brand-specific rate. This allows us to characterize the demand for the i^{th} commodity during the t^{th} time as:

$$\hat{M}_{it} = \rho_i \cdot \left[M_{it} + \sum_{j\neq i} \pi_{ij} \cdot M_{jt} \cdot \left(1 - \rho_j\right)\right] \tag{30}$$

Finally, the product demand can be predicted.

Using the Algorithm 1, we can calculate the optimal set of features from retail sales dataset. Algorithmic steps for the Calculation of probability for The Sequential Bayesian Inference Optimization techniques has been illustrated in Algorithm 2. Finally optimal time series features have been extracted using SBIO and building of the hybrid model with LSTM and Prophet using Algorithm 3 for training and futures sales predicted efficiently using the DPMNN model is evaluated using Mean Squared Error (MSE), MAPE, and RMSE.

Algorithm 1: Time Series Feature selection using Sequential Bayesian Inference Optimization	
	Input: Time series Input features from Retail Sales datasets (M5 Forecasting Accuracy and predict Future data set
	Output: Optimal Time Series Features using SBIO
1	Data initialization {Features *f*}
2	**for each** feature *f*
3	Define modality, outmoded variables and feature probability
4	**while** stopping criteria is not satisfied
5	**for each** kernel do
6	Estimate $\boldsymbol{z_n}$ by employing equation $\boldsymbol{z_n} = \underset{z\in Z}{\operatorname{argmin}} a(\boldsymbol{z})$ where $a(\boldsymbol{z})\text{Fitness} = \mu(\boldsymbol{z}) + \kappa\sigma(\boldsymbol{z})$
7	**end for**
8	Identify the best features
9	***for each*** *features f*
10	Create a random number **rnd** from (0,1) probablity_value← probablity_value$\rho_i, \hat{M}_{it}$)
11	**if** rnd < probablity_value **then**
12	Move towards best solution
13	**else**
14	Move randomly
15	**End if**
16	**End for**
17	Upgrade the value for feature exponent
18	**End while**

Algorithm 2: Calculation of probability Distribution for SBIO	
function probability	
1	**for each** feature *f* Compute ρ_i
2	$\rho_i = \begin{cases} 1, & \text{if i}^{\text{th}} \text{ product is selected} \\ 0, & \text{otherwise} \end{cases}$
3	Arrange the values depend on the rank
4	**End for**
5	**while** (M> max ($\mathcal{E}$ *of* $\hat{M}_{it}$})
6	***If*** *(M=min (*$\mathcal{E}$ *of* $n = \hat{M}_{it}$))
	Eliminate the low Rank values
7	$\hat{M}_{it} = \rho_i \cdot \left[M_{it} + \sum_{j\neq i} \pi_{ij} \cdot M_{jt} \cdot (1-\rho_j)\right]$
8	***end if***
return probability ($\rho_i, \hat{M}_{it}$)	

Algorithm 3: DPMNN (Deep Prophet Memory Neural Network)	
Input: optimal time series features {yi}from SBIO	
Output: Future prediction of retail sales	
1	Split data: 70% training and 30% testing data Size←length(series)*70 train←series{0..size} test ← series {size…length(size)}
2	x←len(optimal_timeseries)* 0.70
3	y← series-x
	#Fit LSTM model
4	model.add(LSTM(neurons), stateful=true)
	# one-step ahead forecast
5	forecast_lstm(model,x)
6	yhat← model.predict(x)
	# validation on the test data
7	**for each** I in range (length(test)) **do**
8	x← test(i)
9	yhat← forecast_lstm(lstm_model,x)
10	predict.append(yhat)
11	expected← test(i)
12	**End for**

13	# fit prophet model
	Model1=prophet ()
	Model1.fit(train)
	pred1=model1.predict(future)

14	total_train=reconstruct(yhat,pred1)
15	total_test= DPMNN(test)
16	Evaluate: MSE= MSE(total_test); MAPE= MAPE(total_test); RMSE= RMSE(total_test)

IV. Performance Analysis

In this research, product demands are predicted by using deep learning based DPMNN categorization strategies that was analyzed in this section. The whole study was coded in Python and ran on a two different type of data collection. Initially the work was carried out in a M5 forecasting dataset. The data is daily and spans January 29, 2011, through June 19, 2016, for a total of 1,969 days (or about 5.4 years). Sales data are broken down by category (Level 4), state (Level 3), and overall (Level 1) in Fig. 2. As can be observed, both daily and monthly levels reveal robust seasonal trends across all series. Sales are around 20% higher on Saturdays and Sundays than on other weekdays, with Fridays and Mondays coming in a close second and third. In general, sales of WI and Hobbies are lower than the rest of the series on Sundays, although the average departures from that seasonal trend are rather tiny. With the exception of the Hobbies subcategory, the series exhibits robust and regular monthly seasonality. Compared to the rest of the year, sales are 6% higher in July and August, 3% higher in March, and 1% lower in February and November. As a result, capturing seasonality accurately at various aggregation levels is seen as the most important component for enhancing the overall efficacy of forecasting.

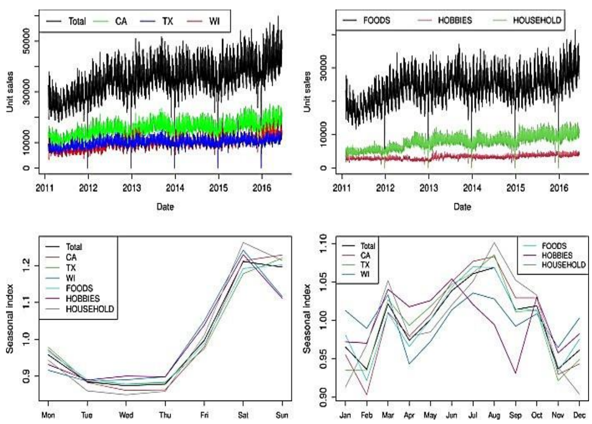

Fig. 2. Analysis of the M5 forecasting dataset.

Demand is higher for stores in California and Texas than in Wisconsin, as seen in Fig. 3. There will be a guaranteed rise in demand in the case of any incident. The demand for cultural and national events tends to be higher. The food section is larger while the hobbies section is smaller. Demand is highest on Saturdays and Sundays. Every year, Christmas has either no sales or a very low total. In order to prove the efficiency of the suggested methodology it can be contrast with the existing Prophet and LSTM over certain parameters

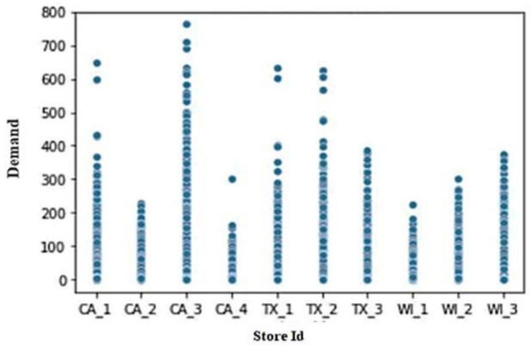

Fig. 3. Demand prediction.

A. Mean Square Error (MSE)

If we need to find deviations, MSE is a tool you can utilize. Evidently, the squaring portion of the function amplifies the mistake if the model ultimately returns a single really incorrect forecast.

$$MSE = \frac{1}{m}\sum_{i=1}^{m} (X_i - Y_i)^2 \tag{31}$$

(best value $= 0$; worst value $= +\infty$)

Since $\mathrm{R}^2 = 1 - \frac{\mathrm{MSE}}{\mathrm{MST}}$ and since MST is fixed for the data at hand, R^2 is negatively (monotonically) linked to MSE; so, a ranking of regression models based on R^2 will be equal (in reverse order) to a ranking of models based on MSE or RMSE. Models may be ranked using either MSE or RMSE, and the results will be the same for both.

B. Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{m}\sum_{i=1}^{m} \left|\frac{Y_i - X_i}{Y_i}\right| \tag{32}$$

C. Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m} (X_i - Y_i)^2} \tag{33}$$

(best value $= 0$; worst value $= +\infty$)

There is a monotonic (through the square root) relationship between MSE and RMSE.

Due to its definition, MAPE is suggested for usage in jobs where it is more critical to be sensitive to relative differences than to absolute variations, making it a useful performance indicator for regression models. However, it also has a number of limitations, the most significant of which are its definitional constraint to only positive data and its bias towards low projections, rendering it unfit for predictive models when big mistakes are predicted. The overall simulated output was illustrated in Fig. 4 by using the DPMNN.

We examine the correlations between demand intensity between the existing Prophet and LSTM with the proposed DPMNN at a particular time interval (year of 2023) which was given in Fig. 4. As of from the result obtained the suggested methodology shows slight difference of actual prediction ratio shows its efficiency over other existing LSTM and prophet

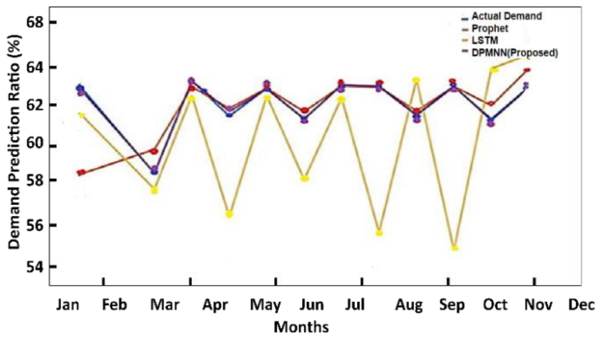

Fig. 4. Actual vs. Predicted ratio.

The more accurate a model's predictions, the smaller the RMSE number should be. The lowest RMSE value is found with our proposed model, as expected identified from Fig. 5. Our Network model outperforms the Linear Regression method, the SVM algorithm, and the NN in terms of root-mean-squared error has been identified from Fig. 6. The results of these experiments demonstrate that our approach, which employs the DPMNN model, is superior at predicting demand from the M5 demand prediction dataset. The DPMNN model was evaluated and the plot results, represents the suggested methodology appear to work well with the input data by obtaining the approximate peak range with the actual values.

After completing analysis with M5 forecasting dataset the analysis was done on Predict Future Sales datasets.

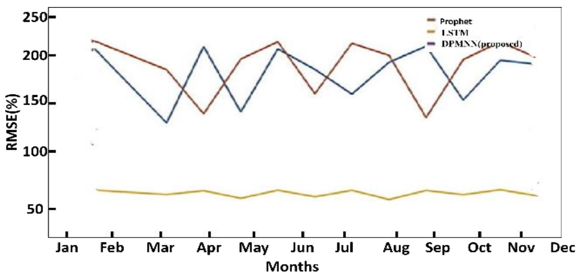

Fig. 5. RMSE calculation.

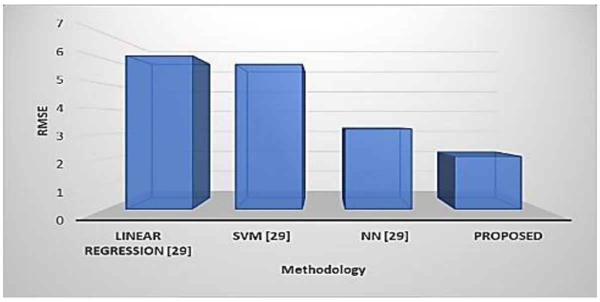

Fig. 6. RMSE comparative analysis.

The features of the Predict Future Sales dataset were depicted in Figs. 7 and 8. The variables can be depicted below,

- Shop_id: The distinct identifier of a shop item; ID is an Id that represents a (Shop, Item) tuple inside the test set._id: A product's unique identification.
- item_category_id: An item category's unique identification.
- item_cnt_day: the quantity of goods sold. You're projecting this measure's monthly amount.
- item_price: The item's current price.
- date: Date in the dd/mm/yyyy format.
- date_block_num: a convenient number representing the next month. October 2015 is 33; January 2013 is 0; February 2013 is 1; and so on.
- item_name: The item's name
- shop_name: The shop's name.
- item_category_name: The category name for the item

The overall prediction output was illustrated in Fig.9 by using the rank obtained over simulation. From the result Models are tested using Predict Future Sales datasets, with DPMNN showing considerable improvement over the benchmark methods in terms of root-mean-squared error (RMSE), suggesting the suggested method is more effective at capturing demand features in product demand forecasting. The proposed DPMNN outperform the Prophet and LSTM in terms of RMSE, indicating that the exogenous variables are highly valued. Exogenous variables refer to predictors that are unrelated to the forecasting model being used, and their forthcoming values must be ascertainable to include them into the prediction procedure. The use of external factors such as promotions, discounts, weather have been considered as important features and it has the potential to improve the precision of predictions.

date_block_num	shop_id	item_category_id	item_id	avg_item_price	avg_item_cnt	item_price	item_cnt	transactions
0	18	35	5288	994,5833	2,352941	16907,92	40	17
0	28	41	19636	2249	1	6747	3	3
0	18	23	4248	1999	3	1999	3	1
0	42	35	5823	2462,857	1,285714	17240	9	7
0	42	75	3146	2485	1	4970	2	2
0	4	55	5324	299	1	2093	7	7
0	6	40	14390	149	1	298	2	2
0	6	55	6007	299	1	299	1	1
0	35	55	4579	291,5	1	583	2	2
0	53	23	3328	2199	1	6579	3	3
0	21	55	2626	299	1	299	1	1
0	44	23	4248	1999	1.5	3998	3	2
0	25	43	8509	249	1	498	2	2
0	31	73	831	4600	1	4600	1	1
0	59	67	10691	850	1	850	1	1
0	28	19	6727	499,5	1	499,5	1	1
0	7	75	3140	1090	1	1090	1	1
0	18	69	14931	699	1	3495	5	5
0	22	49	19346	400	1,625	3200	13	8
0	52	58	7998	899	1	899	1	1

Fig. 7. Sample features of "Predict Future Sales" dataset.

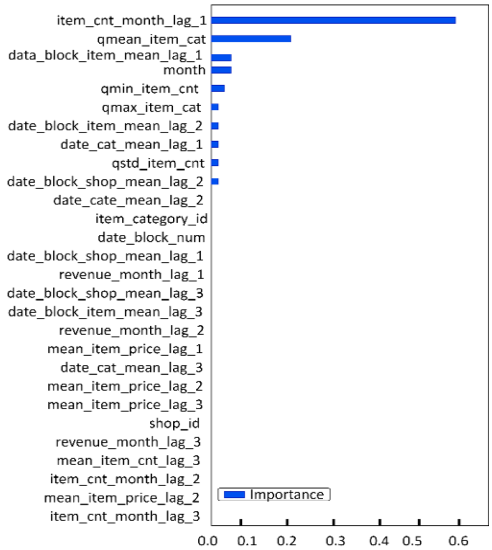

Fig. 8. Feature analysis.

In order to assure the appropriate consideration of their impacts, it is essential to include these exogenous variables like promotions, product discount, weather in both the training and prediction stages. This approach will contribute to enhancing the precision of forecasts and delivering more dependable predictions. This suggests that precise feature selection might reduce computational complexity without significantly diminishing forecast accuracy obtained the item of id 4964 have higher demand than other products. From Fig. 8, the suggested DPMNN in which the prediction ratio shows slight variance comparable to actual prediction output for the year of 2023 shows the efficiency of the suggested methodology. As of from Fig. 9, Feature extraction using sequential Bayesian optimization has been done and optimal features were identified based on the ranking of probabilities. The future sales were predicted depend upon the optimal features analyzed that was depicted elaborately in the Fig. 9.

item_id	shop_id	item_cnt	transactions	year	item_cnt_mean	item_cnt_std	item_cnt_Shiffte1	features	prediction	Rank
5037	5	0	0	2015	0	0	0	(7,[0,1,3], [5.05...	0.109438	0
5037	5	0	0	2015	0	0	0	(7,[0,1,3], [5.05...	0.109438	0
5320	5	0	0	0	0	0	0	(7,[0,1,3], [5.0,5...	0.056177	1
5233	5	1	1	2015	1.666667	1.154701	3	5.0,50.0,2...	2.462165	2
5232	5	0	0	2015	0	0	0	(7,[0,1,3], [5.0,5..	0.109438	3
5268	5	0	0	0	0	0	0	(7,[0,1,3], [5.0,5..	0.056177	4
5039	5	1	1	2015	1	0	1	5.0,50.0,2...	0.338105	5
5041	4	2	2	2015	2.5	0.707107	3	5.0,5041,...	2.34036	6
5046	5	0	0	2015	0	0	0	(7,[0,1,3], [5.05...	0.109438	7
5319	5	0	0	2015	0	0	0	(7,[0,1,3], [5.05...	0.109438	8
5003	5	0	0	2015	0	0	0	(7,[0,1,3], [5.05...	0.109438	9
4806	5	3	3	2015	3.33333	1.527525	2	5.0,472.0,2...	2.004846	10
4843	5	0	0	2015	0	0	0	(7,[0,1,3], [5.05...	0.109438	11
4607	5	0	0	2015	0	0	0	(7,[0,1,3], [5.05...	0.109438	12
4869	5	0	0	2015	0	0	0	(7,[0,1,3], [5.05...	0.109438	13
4870	5	2	2	2015	2.33333	0.57735	2	5.0,4..0,2...	1.178071	14
4872	5	6	6	2015	3	2.645741	1	5.0.0,6.0,2...	1.278332	15
4874	5	0	0	2015	0	0	0	(7,[0,1,3], [5.05...	0.109438	16
4878	5	0	0	2015	0	0	0	(7,[0,1,3], [5.05...	0.109438	17
4892	5	2	2	2015	2	0	2	5.0,0.92.0,2...	0.849117	18
4964	5	0	0	2015	0	0	0	(7,[0,1,3], [5.05...	0.109438	19

Fig. 9. Prediction output of "Predict Future Sales" dataset.

Models are tested using Predict Future Sales datasets, with DPMNN showing considerable improvement over the benchmark methods in terms of Root-Mean-Squared Error (RMSE), suggesting the suggested method is more effective at capturing demand features in product demand forecasting. The proposed DPMNN outperform the Prophet and LSTM in terms of RMSE, indicating that the exogenous variables are highly valued. This suggests that precise feature selection might reduce computational complexity without significantly diminishing forecast accuracy.

From the result obtained the suggested methodology obtained lower ratio of RMSE over predict future sales dataset than other existing mechanisms in use. These factors are used to evaluate our models' efficacy, and their values are summarized in Tables I and II. When compared to the other models, DPMNN performs best on all three metrics (MAPE = 15, RMSE = 98). The findings demonstrate that the DPMNN model outperforms competing neural network models. The DPMNN has surpassed other conventional approaches in terms of predicting accuracy across a variety of performance indicators, making it a standout among traditional models. With the addition of the vacation element, the forecast accuracy has increased to 99.2% (Fig. 10), which is much higher than the already used technique.

TABLE I. COMPARATIVE PERFORMANCE ANALYSIS

Model	Training RMSE	Validation RMSE	Test RMSE
XGBoost [28]	0.764467	0.807264	0.87815
LSTM [28]	0.804657	0.889786	0.92417
ARIMA [28]	0.963426	0.982234	1.09266
Proposed (DPMNN)	0.72	0.72	0.72

TABLE II. PREDICTION RESULTS OF DIFFERENT MODELS [10]

Model	MSE	RMSE	MAPE
ARMA	87,237.010	295.360	33.880
ARIMA	79,804.310	282.500	35.070
SARIMA 1	42,305.370	205.680	28.890
SARIMA 2	55,497.860	235.580	33.500
DES	40,4596.360	636.080	98.790
TES	49,846.4800	223.260	30.810
Prophet1	37,992.52	194.920	26.670
Prophet2	27,986.560	167.290	22.620
Vanilla LSTM	18,829.660	137.220	18.390
Stacked LSTM	16,515.490	128.510	17.3400
Bidirectional LSTM	53,981.320	232.340	31.400
LSTM 1	68,671.500	262.050	29.400
CNN	39,938.470	199.850	22.260
DPMNN (M5 dataset)	10,000.010	99.50	15
DPMNN (Future sales prediction dataset)	10,000	98	15

These factors are used to evaluate our models' efficacy, and their values are summarized in Table II. When compared to the other models, DPMNN performs best on all three metrics (MAPE = 15, RMSE = 98). The findings demonstrate that the DPMNN model outperforms competing neural network models. The DPMNN has surpassed other conventional approaches in terms of predicting accuracy across a variety of performance indicators, making it a standout among traditional models. With the addition of the vacation element, the forecast accuracy has increased to 99.2% (Fig. 11), which is much higher than the already used technique.

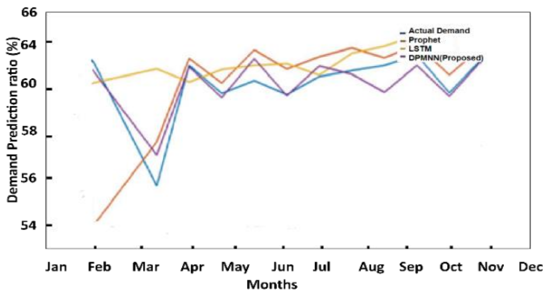

Fig. 10. Actual vs. prediction rate analysis.

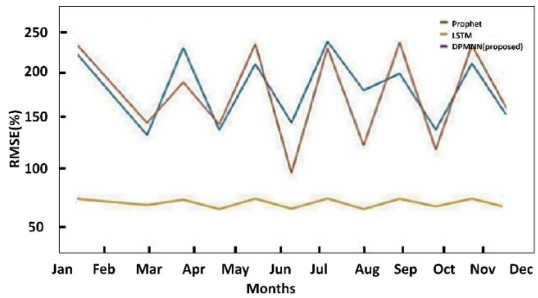

Fig. 11. RMSE analysis.

Tables III and IV exhibits the comparison of the actual and forecasted results for future sales prediction dataset and M5 Accuracy forecasting datasets with its accuracy. The proposed methodology outperforms with the monthly horizon settings for both datasets which gives us good accuracy.

TABLE III. COMPARISON OF ACTUAL AND FORECASTED FOR M5 FORECASTING DATASET

Month	Actual	Predicted	Accuracy (%)
Jan	5175	5521	99.3
Feb	3621	3221	99.2
Mar	6430	6420	99.2
April	2345	2344	99.2
May	2444	2400	99.2
Jun	5000	5000	99.2
Jul	4789	4700	99.2
Aug	2313	2300	99.2
Sep	4567	4500	99.2
Oct	2344	2200	99
Nov	4666	4600	99.2
Dec	2567	2399	98.9

TABLE IV. COMPARISON OF ACTUAL AND FORECASTED FOR FUTURE SALES PREDICTION DATASET

Month	Actual	Predicted	Accuracy (%)
Jan	6600	6521	99.2
Feb	3621	2821	99.2
Mar	5430	5120	99.2
April	2230	2229	99.2
May	5000	4900	99.2
Jun	6797	5723	99.2
Jul	5673	5600	99.2
Aug	3456	3300	99.2
Sep	2543	2541	99.2
Oct	3654	3564	99.2
Nov	3566	3456	99.1
Dec	3454	3462	99.1

The proposed methodology (DPMNN) is compared to existing work [27] on machine learning and deep learning algorithms, including LSTM, BACK PROPAGATION NEURAL NETWORKS, LINEAR REGRESSION, AND MLP REGRESSOR, among others, with which DPMNN predicts future sales more accurately and with higher prediction accuracy is clearly illustrated in the Fig. 12.

This study aims to compare the cross-validation accuracy and the percentage of false negatives (overestimation) across five different categorization models which depicts in the Fig. 13. The magnitude of the bubbles corresponds to the standard deviation of cross-validation accuracy.

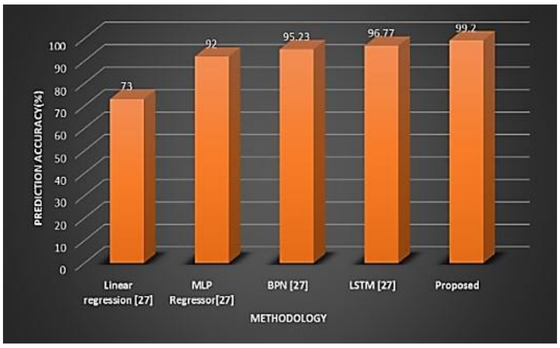

Fig. 12. Prediction accuracy analysis.

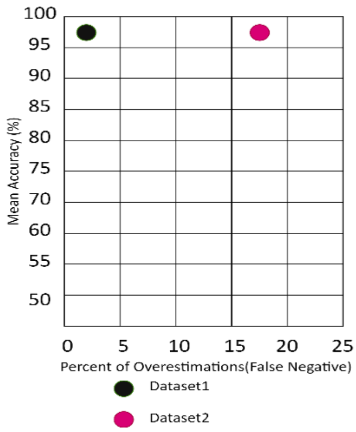

Fig. 13. Cross validation of accuracy.

Fig. 14 exhibits the trade-off between bias and variance is a fundamental concept in statistical modelling and machine learning. Models characterized by low complexity tend to exhibit a higher degree of bias, while models characterized by high complexity tend to exhibit a higher degree of variation. The optimal model will strive to achieve an equilibrium that minimizes the magnitude of prediction errors. From the result obtained the suggested methodology outperforms well than other existing mechanism in use. The suggested model is an essential and pivotal procedure that enables firms to enhance decision-making capabilities via the anticipation of forthcoming trends and consequences. Through the examination of historical data, market trends, and other pertinent elements, the practice of forecasting allows organizations to proactively anticipate shifts in demand or supply and make appropriate adjustments to their strategy.

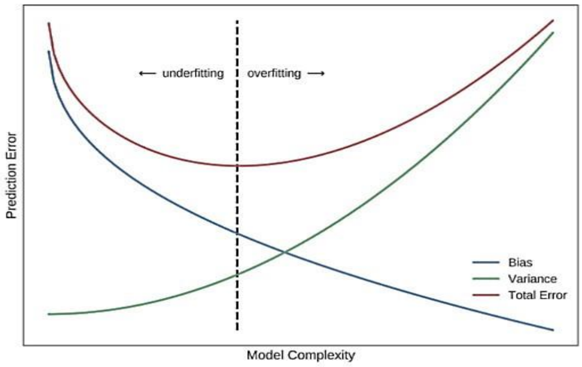

Fig. 14. Trade-off between model complexity and forecasting accuracy error.

V. DPMNN MODEL INTERPRETABILITY USING SHAP (SHAPLEY ADDITIVE EXPLANATION) APPROACH

Time series data forecasting is a significant field for machine learning. However, the lack of explanation provided by researchers for their predictions prohibits machine learning from becoming widely used. The SHAP evaluates the influence of independent variables, or explanatory factors, while accounting for their interactions with different variables as stated in [29]. By contrasting what a model predicts with and without a feature, it determines how important a feature is. Feature importance quantifies the impact of explanatory variables as well; however it experiences a negative impact when variables interact. When the sell_price variable is removed, for example, Percentage of Revenue might receive a greater feature priority. If the sell_price variable is added, the feature relevance of Percentage of Revenue decreases.

According to Fig. 15, the variable Percentage of Sales has the greatest SHAP value, although its importance is very modest. Based on the end result, the selling price is surprisingly unimportant. On the contrary Stores, states, departments, and weekdays variables, have a greater significance despite their low SHAP value. These variables are based on demography and time in general. As a result, the predictive power in this situation may outperform on demographic and time-specific variables.

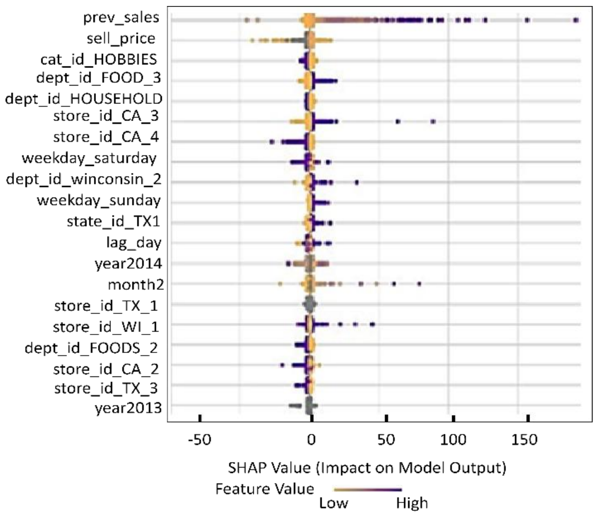

Fig. 15. SHAP for DPMNN interpretability (M5 Accuracy Dataset).

From Fig. 16, in predict future sales dataset, prev_days_on_sale_item has the high SHAP value which helps to identify the future sales easily. "Black Boxes" of the DPMNN Model can be revealed by using this explainable AI approach which delves to reveal which features have the most influence on prediction. This data can be used to influence corporate decisions, marketing initiatives, and product development efforts.

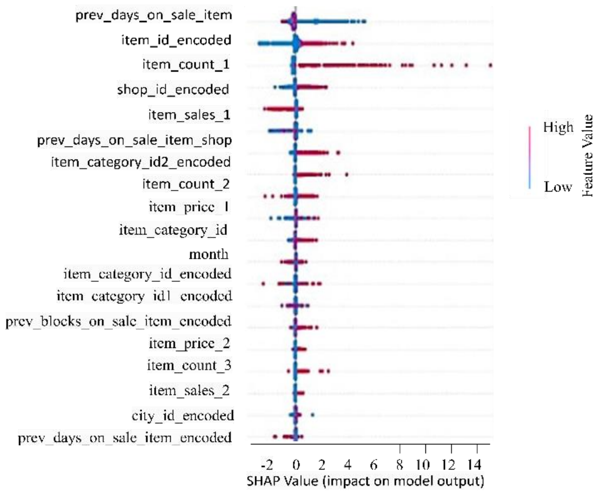

Fig. 16. SHAP for DPMNN interpretability (Predict Future Sales Dataset).

VI. Conclusion

The purpose of this work is to propose a DPMNN model for demand prediction. According to the findings, DPMNN is the best model available. The results have also shown the potential benefits of a hybrid Prophet/LSTM model. This model has performed remarkably well, and better than the other conventional approaches, by obtaining high range of demand prediction accuracy (99.2%).

The findings of this study can be used to anticipate future sales of any item with an identical seasonality. Furthermore, the given models can be applied to different datasets by completing the appropriate data pre-processing and modifying the hyperparameters. This has significant impacts for planning inventories and effective managerial decisions.

This article's primary drawback was the proposed models, which are usually created using historical values. They can make forecasts for a certain time period, are simple to model, and can have their parameters adjusted based on the discrepancy between the actual and expected values in the recent past. Nevertheless, some of them fail to account for the impact of other variables, such as demand at other network nodes, that may have an impact on demand. The recommended Networks are made to use a statistical method to discover how these parameters and demand relate to one another. Although the hybrid mathematical model used in the recommended techniques is expensive. The recommended network produces a result with the highest number of mistakes if there are any patterns contained in the data. However, when the time horizon is increased, the trends are not linear, or there are exogenous influences present, these models' forecast accuracy dramatically decreases. We want to expand our strategy in the future to address the shortcomings in the current paper, concentrate on other kinds of sales challenges, and improve our model to enable it to provide forecasts that are more precise.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

The authors confirm contribution to the paper as follows: Study conception and design: Praveena S, Prasanna Devi S; Data collection: Praveena S; Analysis and interpretation of results: Praveena S, Prasanna Devi S; draft manuscript preparation: Praveena S. All authors reviewed the results and approved the final version of the manuscript.

References

[1] G. P. Zhang and D. M. Kline, "Quarterly time-series forecasting with neural networks," *IEEE Transactions on Neural Networks*, vol. 18, pp. 1800–1814, 2007.

[2] S. Chopra and P. Meindl, "Supply chain management," *Strategy, Planning & Operation*, 2007.

[3] P. Ramos, N. Santos, and R. Rebelo, "Performance of state space and ARIMA models for consumer retail sales forecasting," *Robotics and Computer-Integrated Manufacturing*, vol. 34, pp. 151–163, 2015.

[4] H. Yang, X. Li, W. Qiang, Y. Zhao, W. Zhang, and C. Tang, "A network traffic forecasting method based on SA optimized ARIMA–BP neural network," *Computer Networks*, vol. 193, 108102, 2021.

[5] C. Tandon, S. Revankar, and S. S. Parihar, "How can we predict the impact of the social media messages on the value of cryptocurrency? Insights from big data analytics," *International Journal of Information Management Data Insights*, vol. 1, 100035, 2021.

[6] P.-F. Pai, K.-P. Lin, C.-S. Lin, and P.-T. Chang, "Time series forecasting by a seasonal support vector regression model," *Expert Systems with Applications*, vol. 37, pp. 4261–4265, 2010.

[7] L. F. Tratar, B. Mojskerc, and A. Toman, "Demand forecasting with four-parameter exponential smoothing," *International Journal of Production Economics*, vol. 181, pp. 162–173, 2016.

[8] I. Svetunkov, H. Chen, and J. E. Boylan, "A new taxonomy for vector exponential smoothing and its application to seasonal time series," *European Journal of Operational Research*, vol. 304, pp. 964–980, 2023.

[9] C. Li and A. Lim, "A greedy aggregation–decomposition method for intermittent demand forecasting in fashion retailing," *European Journal of Operational Research*, vol. 269, pp. 860–869, 2018.

[10] Y. Ensafi, S. H. Amin, G. Zhang, and B. Shah, "Time-series forecasting of seasonal items sales using machine learning—A comparative analysis," *International Journal of Information Management Data Insights*, vol. 2, 100058, 2022.

[11] S. J. Taylor and B. Letham, "Forecasting at scale," *The American Statistician*, vol. 72, pp. 37–45, 2018.

[12] G. Papacharalampous, H. Tyralis, and D. Koutsoyiannis, "Predictability of monthly temperature and precipitation using automatic time series forecasting methods," *Acta Geophysica*, vol. 66, pp. 807–831, 2018.

[13] T. H. McCoy, A. M. Pellegrini, and R. H. Perlis, "Assessment of time-series machine learning methods for forecasting hospital discharge volume," *JAMA Network Open*, vol. 1, pp. 184087–184087, 2018.

[14] A. Subashini, K. Sandhiya, S. Saranya, and U. Harsha, "Forecasting website traffic using prophet time series model," *Int Res J Multidiscip Technovation*, vol. 1, pp. 56–63, 2019.

[15] C. B. A. Satrio, W. Darmawan, B. U. Nadia, and N. Hanafiah, "Time series analysis and forecasting of coronavirus disease in

Indonesia using ARIMA model and prophet," *Procedia Computer Science*, vol. 179, pp. 524–532, 2021.
[16] I. Alon, M. Qi, and R. J. Sadowski, "Forecasting aggregate retail sales: A comparison of artificial neural networks and traditional methods," *Journal of Retailing and Consumer Services*, vol. 8, pp. 147–156, 2001.
[17] G. Verstraete, E.-H. Aghezzaf, and B. Desmet, "A data-driven framework for predicting weather impact on high-volume low-margin retail products," *Journal of Retailing and Consumer Services*, vol. 48, pp. 169–177, 2019.
[18] M. Zekic-Susac, A. Has, and M. Knezevic, "Predicting energy cost of public buildings by artificial neural networks, CART, and random forest," *Neurocomputing*, vol. 439, pp. 223–233, 2021.
[19] A. K. Kar, "Bio inspired computing—A review of algorithms and scope of applications," *Expert Systems with Applications*, vol. 59, pp. 20–32, 2016.
[20] A. Chakraborty and A. K. Kar, "Swarm intelligence: A review of algorithms," *Nature-Inspired Computing and Optimization: Theory and Applications*, pp. 475–494, 2017.
[21] A. K. Kar and Y. K. Dwivedi, "Theory building with big data-driven research–Moving away from the 'What' towards the 'Why'," *International Journal of Information Management*, vol. 54, 102205, 2020.
[22] Y. Kaneko and K. Yada, "A deep learning approach for the prediction of retail store sales," in *Proc. 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 2016, pp. 531–537.
[23] H.-C. Lee, H.-C. Rim, and D.-G. Lee, "Learning to rank products based on online product reviews using a hierarchical deep neural network," *Electronic Commerce Research and Applications*, vol. 36, 100874, 2019.
[24] J. C. B. Gamboa, "Deep learning for time-series analysis," arXiv preprint, arXiv:1701.01887, 2017.
[25] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," arXiv preprint, arXiv:1607.00148, 2016.
[26] R. S. Raval, "State-of-the-art deep learning for multi-product intermittent time series forecasting," Master thesis, Pennsylvania State University, 2021.
[27] B. Lakshmanan, P. S. N. Vivek Raja, and V. Kalathiappan, "Sales demand forecasting using LSTM network," *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pp. 125–132, 2020.
[28] D. Swami, A. D. Shah, and S. K. Ray, "Predicting future sales of retail products using machine learning," arXiv preprint, arXiv:2008.07779, 2020.
[29] J. Chen, W. Koju, S. Xu, and Z. Liu, "Sales forecasting using deep neural network and SHAP techniques," in *Proc. 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, 2021, pp. 135–138.

Copyright © 2024 by the authors. This is an open access article distributed under the Creative Commons Attribution License (CC BY-NC-ND 4.0), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.