Journal of Theoretical and Applied Information Technology
28th February 2026. Vol.104. No.4
© Little Lion Scientific

ISSN: **1992-8645** www.jatit.org E-ISSN: **1817-3195**

PERSONALIZED RESEARCH RECOMMENDATION BASED ON TEMPORAL FUSION TRANSFORMER FOR AMAZON FASHION PRODUCTS

GOKILAVANI A[*1,2], DR P AMUDHA[3]

*Corresponding author:gokilavanics005@gmail.com

[1]Research Scholar, Department of Computer Science and Engineering, Avinashilingam Institute for Home Science & Higher Education for Women, TamilNadu, India, 641043
[2]Assistant Professor, Department of Computer Science and Engineering, Jai Shriram Engineering College, TamilNadu India,638660
[3]Professor, Department of Computer Science and Engineering, Avinashilingam Institute for Home Science & Higher Education for Women,TamilNadu India, 641043

ABSTRACT

Personalized product recommendations have become more valuable in recent years as a means of improving the online shopping experience for customers. However, accurate sales forecasting for fashion products remains challenging due to high demand variability, short product life cycles, and complex temporal dependencies influenced by pricing, discounts, and customer behavior. A robust framework for sales prediction is created using a structured pipeline that integrates data collection, preprocessing, feature selection, predictive modeling, and performance analysis. The first step in the inquiry is gathering sales data from Amazon, a dataset known for its size, complexity, and range of features. To ensure data quality and relevance, data is put through preparation processes like data cleaning, transformation, and exploratory data analysis (EDA), which help to identify patterns and correlations in the dataset and prepare it for modeling. During the feature selection stage, the Boruta algorithm is employed, utilizing its ability to identify significant characteristics that improve the model's predictive capabilities. This approach retains only the most relevant information, reducing computing complexity and increasing model efficiency.

For the prediction phase, three different models are examined: a CNN-BiLSTM model, which combines convolutional layers for feature extraction with bidirectional LSTMs for sequential learning; a Temporal Fusion Transformer (TFT), which is well-known for its interpretability and ability to control temporal dynamics; and a hybrid model that incorporates aspects of both CNN and TFT. This combination aims to capture complex temporal patterns and interactions within the data to optimize forecast accuracy.The models' performance is evaluated using a variety of error metrics, including Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Weighted Mean Absolute Percentage Error (WMAPE). These measures provide a comprehensive picture of prediction accuracy by highlighting both absolute and relative errors. By using a range of evaluation criteria, this study ensures that the models' performance is looked at from multiple perspectives, offering insights into the models' practicality and robustness. Experimental results show that the proposed hybrid CNN-TFT model outperforms CNN-BiLSTM and standalone TFT models, achieving an accuracy of 97.25% with lower MAPE (0.383) and RMSE (0.1908). the results demonstrate that the proposed hybrid framework provides reliable and accurate sales forecasts, supporting effective demand planning, inventory optimization, and decision-making in fashion e-commerce platforms.

Keywords: *Temporal Fusion Transformer, CNN, LSTM, Fashion Sales, Prediction.*

1. INTRODUCTION

Over the past ten years, online shopping (web-based shopping) has become immensely popular in India [21]. Easy access to the internet, the proliferation of inexpensive smartphones, mobile shopping apps, expanding awareness in smaller towns and rural regions, and rising income are major reasons driving the growing market share of online retail. Prices and quality are now expected to meet much higher standards by consumers, and the market is becoming more volatile and competitive. Companies now need to focus on efficient supply chain management and gain a better understanding of the needs and behaviors of their customers. They are unable to continue relying only on production cost benefits. Companies in the fiercely competitive retail fashion sector are constantly devising innovative ways to customize their product characteristics to meet the unique needs and preferences of their customers. Fashion products typically have short lifespans, but the vast volumes of historical data that are gathered and kept in corporate systems may help define purchasing and inventory strategies [9].

Predicting client demand is essential to supply chain management since it reduces delivery times and helps avoid over or underproduction. In e-commerce, sales volume is a popular metric for estimating customer demand. Because of this, precisely assessing consumer demand requires a thorough analysis of several variables and how they interact, such as product type, country of purchase, price, discount rate, free delivery option, and mood of online reviews. For online merchants such as Amazon, this type of prediction ability is especially important to ensure customer satisfaction and efficiently manage the supply chain [7]. It is challenging for many e-commerce businesses to manage and evaluate massive volumes of data. The business needs to determine an appropriate plan to boost output and company value to forecast future sales and market demand [4].

While understated sales can result in unmet orders, a loss of profit, and a decline in the company's reputation, overestimated sales can cause excessive inventory, bad cash flow, and even insolvency [11]. Sales prediction is essentially a time series forecasting issue, with the goal of forecasting future sales volume utilizing observed multivariate time series data that includes previous sales volume and pertinent attributes (e.g. brand, season, discount, etc.). To predict sales volume, it is therefore required to accurately model the key elements and previous sales data by creating specific conditions that arise from the interaction of multiple relevant variables, it is feasible to predict the evolution of a trend in sales time series.Recurrent Neural Network (RNN) models are being extensively researched and applied to learn vector illustrations from sequential inputs with regard to context-based learning from raw time series [12].

Compared to earlier time series prediction methods like kernel methods and Gaussian processes, which are constrained by their predetermined nonlinear structures, RNNs are more adaptable and have better modeling skills for complicated, discriminative nonlinear relationships. Additionally, sophisticated RNN variations like the gated recurrent unit (GRU) and long short-term memory (LSTM) [13] have greatly enhanced performance on tasks like image captioning and neural machine translation. The encoder–decoder RNN architecture, which uses two separate RNNs—one to encode the sequential inputs into latent context-dependent vectors and another to decode these vectors into the necessary outputs—is a significant advancement in these applications. Applying encoder-decoder RNNs to sales prediction makes sense because of their efficacy in time series modeling. This is because of their capacity to faithfully depict the nonlinear correlations between sales volume and influential variables [12-14].

Forecasting and inventory management are made more difficult by censored demand, which also sales data. By has an impact on sales in the retail fashion sector. A novel model is developed in order to account for missed sales and mitigate the negative effects of censored demand on sales because there isn't much research on forecasting total sales for fashion products with censored demand. This article aims to give managers useful guidance for decision-making while also increasing the accuracy of overall sales estimates for fashion products [15].

Despite the widespread adoption of DL models for sales forecasting, conventional approaches often struggle to capture the short-term demand fluctuations and long-term temporal dependencies inherent in fashion product sales. Additionally, the high volatility caused by promotions, seasonality, and rapidly changing customer preferences limits the generalization ability of single-model architectures. These challenges highlight the need for a forecasting framework that can simultaneously model local patterns and long-range temporal interactions while maintaining robustness across diverse product categories.

To address these challenges, this work proposes a hybrid forecasting framework that integrates CNN with TFT. The CNN module is employed to extract localized temporal features and short0term demand variations, while the TFT component models long-term dependencies and incorporates interpretable temporal attention mechanisms. Despite significant advances in time-series forecasting and DL models, accurate sales prediction for fashion products remains an unsolved issue. From a theoretical perspective, fashion sales data exhibit strong non-stationarity, sparse temporal patterns, and abrupt regime shifts caused by seasonal trends, promotional campaigns, and rapidly evolving consumer behavior. Conventional statistical models fail to capture the complexities in the data. As a result, forecasting errors propagate into inventory mis management, overstocking, and revenue loss, underscoring the continued relevance and practical implementation of the problem.

From a modeling standpoint, existing approaches typically emphasize on either short-term fluctuations or long-term dependencies but not both simultaneously. CNN-based models are effective in capturing local patterns but lack temporal awareness, whereas transformer-based models excel at long-range dependency modeling but often overlook fine-grained short-term variations. This theoretical mismatch motivates the need for a unified approach capable of learning multi-scale temporal environments. The study hypothesizes that integrating CNN with TFT improves fashion sales forecasting by capturing both short-term demand variation and long-term temporal dependencies.

The key contributions of this study are threefold. First, a hybrid CNN-TFT architecture is proposed to capture the short-term demand fluctuations and long-term temporal dependencies in fashion sales data. Second, an effective and-to-end forecasting pipeline integrating Boruta-based feature selection with deep temporal modeling is developed to handle high-dimensional, non-stationary e-commerce data. Third, extensive experimental evaluation on real-world Amazon fashion data demonstrates that the proposed hybrid model consistently outperforms CNN-BiLSTM and standalone TFT models across multiple performance metrics, while also offering improved interpretability for practical decision-making.

The study adopts a quantitative, experimental research design based on supervised learning and comparative model evaluation, consistent with prior time-series forecasting studies in retail demand prediction [1,2,4,7,9], energy load forecasting [3, 6], and supply chain analytics [10, 15]. By extending these established DL methodologies to the fashion e-commerce domain, the proposed CNN-TFT framework addresses highly volatile and non-stationary sales patterns.

2. SURVEY

The development of Deep Learning (DL) and machine learning models has made it simpler to anticipate sales data. CNN is the most often utilized DL technique [4, 23], with CNN models being used for 3, 6, and 12-month sales forecasts. Huang (2021) and Dai (2021) used an LSTM model with a distinct loss function and hyper-parameter search to maximize accuracy. The performance is displayed using sales data from Kaggle Rossman, an accessible dataset. The experiment results show that, in comparison to other machine learning models using the AutoML (Auto Machine Learning) tool, the proposed technique significantly increased prediction performance on sparse data [5]. The study examines the efficacy of many modeling techniques, such as regression analysis, decision-tree analysis, and Artificial Neural Networks (ANN), for projecting book sales at amazon.in. It does this by using a range of relevant factors and their interactions as predictor variables. Online reviews are used as predictors in these models, and sentiment analysis quantifies their polarity. A comparison of these models yields some important conclusions. First off, the number of reviews is the most significant and relevant predictor of book sales on Amazon.in, according to all three models. Second, average ratings, discount amount, and discount rate have little effect on sales forecasting. Thirdly, the reviews' positive and negative sentiments are not significant in NN models, but they are separately significant predictors in regression and decision-tree models [7].

A preprocessing method based on data mining is recommended for developing a customer profiling system that incorporates customer action prediction and customer equity assessment to improve sales performance. The RFM (Recency, Frequency, Monetary) analysis methodology is used to evaluate client capital, and for forecasting boosted trees is employed. The study highlights how important algorithms and consumer segmentation strategies are to improving forecast accuracy. The main result of this study is the creation of a client profile and sales forecast [8]. In order to predict sales of new distinct items in upcoming seasons, the study applies a DL technique to the fashion business. The physical characteristics of the products as well as the guidance of subject matter experts were taken into consideration when creating the models. This study also compares a set of techniques, such as ANN, Linear Regression, Decision Trees, Random Forests, and Support Vector Regression (SVR), with the sales estimates generated by the DL strategy [9].A NN-based method is suggested, in which the decoder forecasts the sales using the available visual and metadata data, along with the Google Trends encoding, after the encoder has learned a model of the exogenous time series. Being non-autoregressive, the model does not compound significant first-step errors. The second contribution is VISUELLE, an open-source database used to predict sales for newly released fashion items. It contains multimodal information about 5577 real new items that Nunalie, an Italian company, sold from 2016 to 2019 [10].

To predict the present value of a time series based on both the past and present state, a dual-stage attention-based RNN (DA-RNN) is developed. There are two stages to the model. In order to extract the pertinent input features at each step that correlate to the previous encoder hidden state, an input attention approach is first applied. Temporal hidden states are employed in the second phase. The model's data analysis is made simpler by the two-step procedure. The DA-RNN model is demonstrated using the NASDAQ 100 and the SML 2010 stock dataset [14].

Each cluster's parameters are estimated using an evolutionary approach, and product clustering is done using a mixed k-mean strategy. According to the findings from experiments, the model outperforms LR, GBDT, SVR, and ANNs in the majority of scenarios. The model is evaluated using real-world data from a Singaporean company. Additionally, two indicators are developed: "the marginal conversion rate and the average conversion rate", which are used to determine the ideal inventory level and assess the competitiveness of the commodities, respectively. When making decisions, managers in the fashion business can benefit from the guidance these data offer [15].

The process starts with reading sales data from a dataset, which is then passed through a knowledge representation and preprocessing stage. The training and testing datasets are generated from the preprocessed data. The ML algorithm builds a predictive model by identifying patterns in the training data. Following training, the model is used to produce projected outcomes on the test dataset with the goal of predicting future sales patterns. [16]. A unique life cycle and process model with RFM is utilized to forecast sales and market items, and it is combined with user behavior analytics. The company divides its clientele into discrete levels using RFM, making it possible to assess the relative importance of each level. To forecast purchases, XGBoost and Random Forest are employed. The five-fold cross-validation approach is used to assess this model. The apriori algorithm and the theory of association rules are used to complete the basket analysis in the product recommendation model. The trial's F1 score 0.789 shows that XGBoost increased accuracy [17].

Initially, the user was offered the choice to register with a secure login by the authors of [18]. In doing so, the system analyzes and creates a database according to the handler's specifications. The input is compared to the trained model after extraction. Ultimately, the system uses user behavior to forecast sales [18].TADA+ model is improved by an online learning module that makes use of a unique similarity-based reservoir. The similarity-based reservoir, in contrast to typical reservoirs that depend on random sampling, deliberately chooses data samples from which the model finds it more difficult to identify dynamic patterns, hence enhancing model retraining. Tests conducted on two real-world datasets unequivocally demonstrate that “TADA and TADA+” perform better in online and offline sales prediction tasks than other cutting-edge algorithms [20].

Despite advances in DL, fashion e-commerce sales forecasting remains challenging due to high volatility, non-stationarity, and short product life cycles. Existing models typically capture either short-term demand fluctuations or long-term temporal dependencies, but not both simultaneously. CNN-based methods model local patterns but miss long-range dependencies, while transformer-based models emphasize long-term trends and underrepresent short-term variations. These limitations reduce forecasting accuracy and practical applicability, highlighting the need for a unified, interpretable hybrid framework for multi-scale temporal modeling.

3. METHODOLOGY

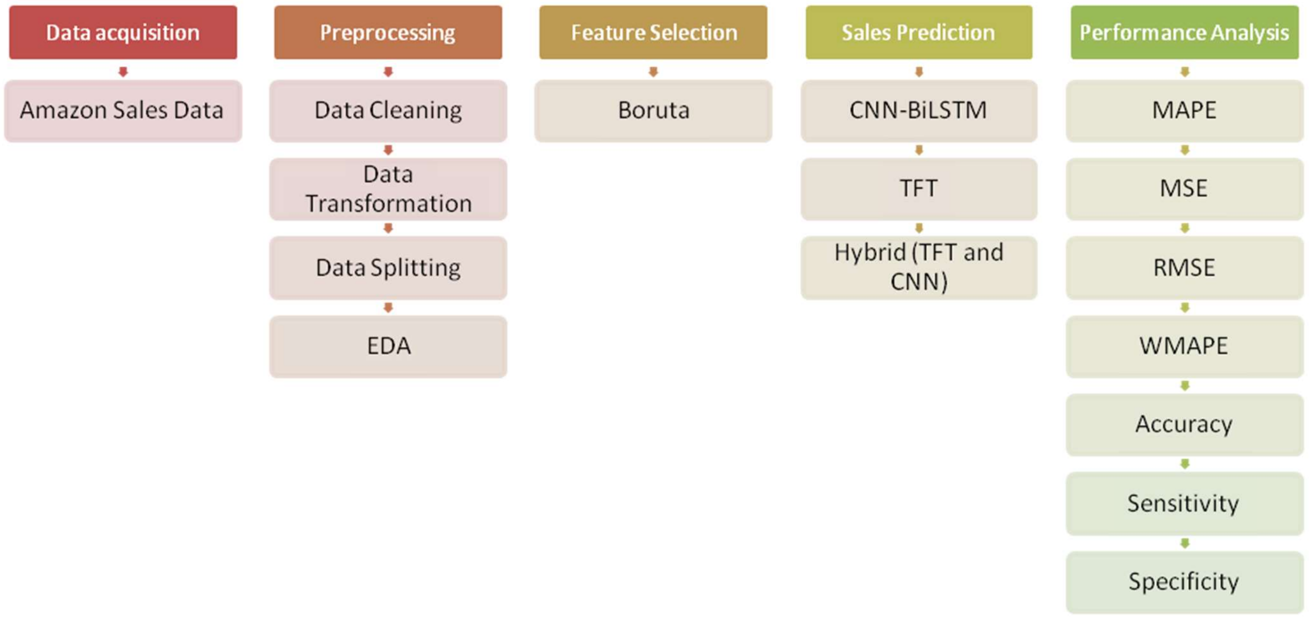


Figure 1: Workflow of the study

Figure. 1 illustrates the pipeline for predicting Amazon sales data, which starts with data collecting and proceeds through several stages. Gathering the Amazon Sales Data is the first phase in the process, after which it goes through preparation. After data cleaning is finished, data transformation is used to fix any inconsistent or missing data by transforming the data into the proper format. To search for trends or patterns, EDA is carried out. In the Feature Selection stage, the Boruta algorithm, a dependable method for selecting relevant featuresis utilized. In the Sales Prediction stage, models such as CNN-BiLSTM and Temporal Fusion Transformer (TFT) are employed. A hybrid method that combines CNN and TFT models is also available for better performance. Finally, a range of evaluation metrics, including MAPE, MSE, RMSE, and WMAPE, are employed in performance analysis to assess the effectiveness and accuracy of the sales forecast models.

3.1 Dataset Description

Figure 2. describes the fashion data considered in this study. The data has been taken from a public repository Kaggle. The data consisted of 142 categories.

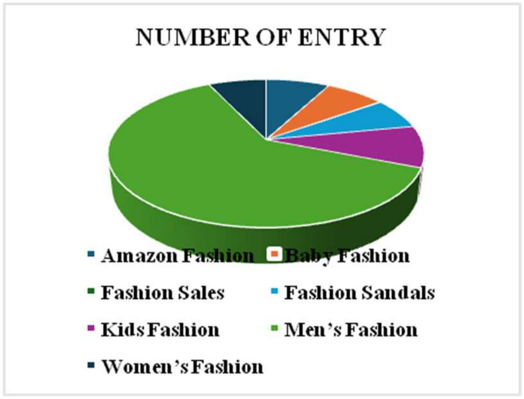


Figure 2. Data in each Category

Of which only the fashion data is selected. A total of 28000 data entries are selected, which contain some redundant and missing data, which is then corrected in the pre-processing step.

3.2 Preprocessing

When it comes to machine learning and data-driven decision-making, pre-processing Amazon sales data is a vital in getting the input ready for analysis and modeling. The organization, transformation, and cleansing of the raw data to enhance its quality for analysis are some of the crucial steps.

Data Cleaning:

Handling Missing Values:Data that has more than 50% missing values is deleted. If the data are continuous, the missing value is replaced with the mean range.

Checking for redundant data: The duplicate entries are removed by matching the product or the order IDs.

Data Transformation:

Normalization eliminates the data's unit limit, which is important for indication comparison and weighing. The following is a representation of the min-max normalization scheme:

$$Y_{i,j} = \frac{x_{i,j}-x_{min,j}}{x_{max,j}-x_{min,j}} \left\} \begin{array}{l} i = 1,\dots,n \\ j = 1,\dots,m \end{array}\right. \quad (1)$$

where $X_{min,j}$ is the min value, and $X_{max,j}$ is the max value.

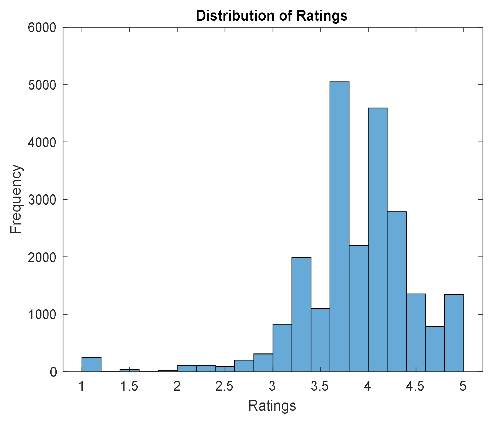
Categorical Encoding: The categorical variables are converted into numerical formats using label-encoding method.

Data Splitting:

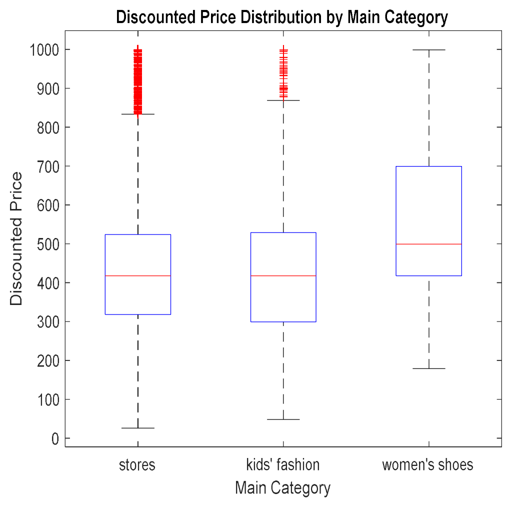
The ratio of the split if the dataset is 70:30.

Exploratory Data Analysis (EDA):

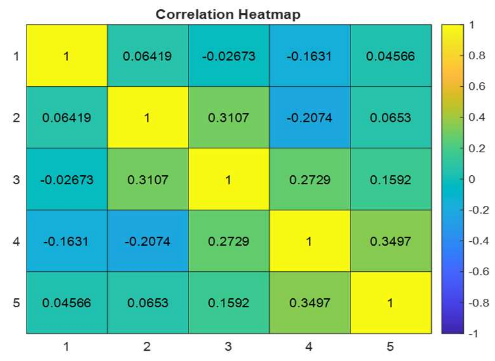
EDA is conducted to identify the patterns and visualize the relationship between the variables.



(a)



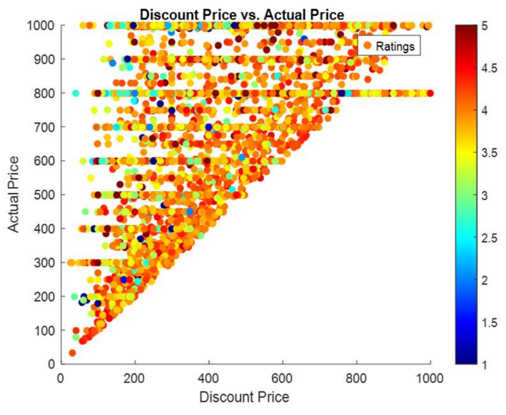
(b)



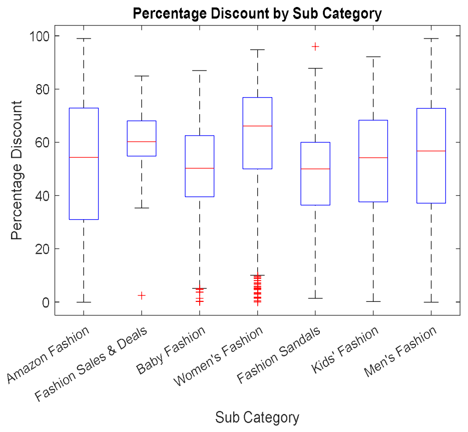
(e)

Figure.3 EDA Analysis

Figure. 3 elaborates the output from EDA. The histogram in panel (a) shows the distribution of product ratings within the dataset. The x-axis displays ratings, which range from 1 to 5, and the y-axis displays the frequency of items at each rating level. The fact that most products appear to have ratings between 4 and 5 indicates a general trend toward positive customer satisfaction. This concentration of higher ratings indicates that most customers are generally satisfied with their purchases because lower ratings are less common. Panel (b) shows a box plot of discounted pricing distributions for the different key product categories. While the x-axis displays the main categories (such as "stores," "men's fashion," and "women's shoes"), the y-axis displays the decreased expenses.



(c)



Each box displays the interquartile range (IQR) for that category, and the line inside each box represents the median discounted price. Individual points outside of the whiskers, which show the range of prices within each category, are used to represent outliers. This analysis displays the price variability and dispersion in each category. Some categories, like "stores," have a greater selection of discounted prices, suggesting that the prices of the products in this category vary significantly.

The scatter plot in panel (c) illustrates the relationship between discounted and real prices. Each point represents a product, with the actual price on the y-axis and the discounted price on the x-axis. The product's rating is represented by the color of each point, creating a heatmap effect that shows how rating and price are related. This plot demonstrates a strong positive correlation between discounted and actual pricing, which is expected given that higher actual prices usually translate into greater discounted prices. The color gradient also helps identify any similarities between price and rating, even when it appears that products with varying ratings are spread out across the price range. The box plot in panel (d) shows the distribution of percentage discounts across different subcategories. The y-axis displays the percentage discount, while the x-axis lists the subcategories. The median, which shows the IQR of the percentage discounts for products in each subcategory, is shown as a line inside each box. Outliers are shown by points outside the whiskers. This statistic enables comparison of discount levels across subcategories, showing, for example, that certain subcategories consistently provide greater discounts than others, suggesting targeted promotions or aggressive pricing strategies in specific geographic areas.

The association between the dataset's several numerical variables, such as ratings, actual prices, discounted prices, and percentage discount, is depicted in the heatmap in Panel (e). Stronger correlations are shown by darker hues, while the direction and strength of the association are indicated by color intensity. Understanding the relationships between variables is made easier with the help of this heatmap. The real and discount prices, for example, have a definite positive correlation, indicating that more costly items usually maintain their higher discount price. Conversely, a lack of direct links between other variables may be indicated by poor or negative correlations between them.

3.3 Feature selection

One of the most important aspects of machine learning is choosing the most pertinent features from a dataset. It significantly affects decreasing overfitting, enhancing interpretability, and enhancing model performance.

a. BORUTA

Boruta is a feature selection technique that has gained popularity due to its ability to identify statistically significant qualities in high-dimensional datasets. The Boruta technique, which is a variant of the Random Forest (RF) algorithm, determines the significance of each feature by combining shadow characteristics with statistical testing. Boruta determines the relative weight of each original characteristic to its matching shadow feature to differentiate between significant and minor qualities.

Step 1: Original features are added to the feature set.

Step 2: Shadow features are created by randomly permuting the original feature as in step

1.

Step 3: The ML model is trained using the output from step 1 and 2.

Step 4: The distinctive features are evaluated by contrasting it with the shadow elements. The relevance used to quantify the features are information gain, Gini impurity and the weights associated with the feature vectors.

Step 5: The statistical importance of a trait is determined by a predefined threshold. If a feature's value is more than the sum of the significance of its shadow traits, it is considered statistically significant.

Step 6: The unwanted features are eliminated from the list of features.

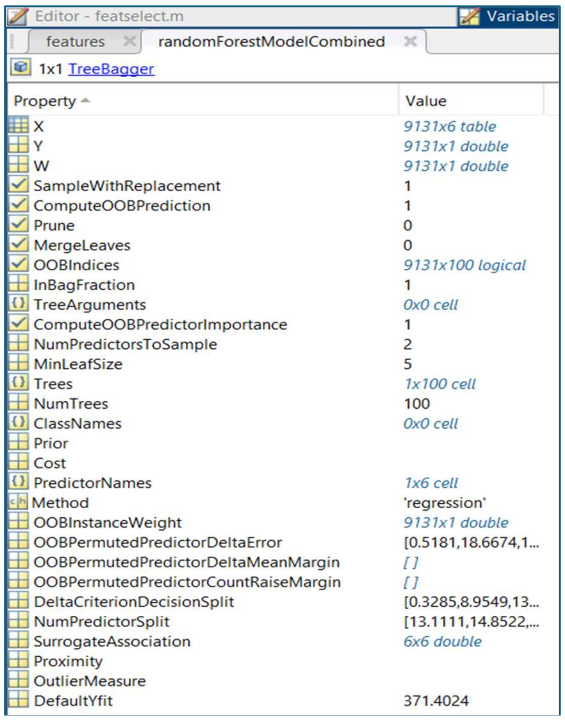
Step 7: Steps 2 through 6 is followed until each feature is approved or rejected.

By randomly permuting the starting feature values while maintaining the target variable constant, shadow features are produced. Boruta compares the value of each initial feature to its shadow feature in order to ascertain whether a feature is statistically significant. The Boruta approach states that the traits that were rejected were considered unimportant, while the features that have been proven are deemed suitable for the prediction objective. Boruta has certain advantages. It can manage a range of ML models, both tree-based and non-tree-based in its methodology, redundancy, and feature interactions.

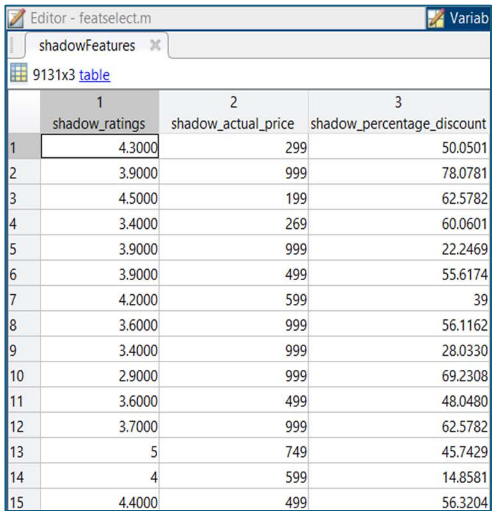
Furthermore, Boruta can produce reliable findings even with noisy datasets due to its degree of noise resistance. It is crucial to remember that Boruta's capacity to select the ideal subset of features might not always be achievable because of things like the significance threshold selection and the caliber of the feature significance measure. Additionally, Boruta could be computationally expensive, especially when working with large feature datasets [27]. Figure. 4. shows the experimented output using Boruta feature selection method.

Boruta is a potent feature selection method that iteratively assesses each feature's significance in a dataset to determine which characteristics are most pertinent for predictive modeling. Boruta assists in identifying the aspects that have the greatest influence on fashion sales forecasting, where a wide range of factors, including pricing, trends, promotions, seasonality, and client demographics, can affect sales.

Boruta carefully evaluates if the importance of a feature surpasses the noise level by generating shadow features (randomized copies) using a RF classifier. By doing this, overfitting is decreased, model interpretability is improved, and sales prediction accuracy is increased in the dynamic and intricate fashion retail sector by ensuring that only attributes with strong predictive potential are kept.



Property	Value
X	9131x6 table
Y	9131x1 double
W	9131x1 double
SampleWithReplacement	1
ComputeOOBPrediction	1
Prune	0
MergeLeaves	0
OOBIndices	9131x100 logical
InBagFraction	1
TreeArguments	0x0 cell
ComputeOOBPredictorImportance	1
NumPredictorsToSample	2
MinLeafSize	5
Trees	1x100 cell
NumTrees	100
ClassNames	0x0 cell
Prior	
Cost	
PredictorNames	1x6 cell
Method	'regression'
OOBInstanceWeight	9131x1 double
OOBPermutedPredictorDeltaError	[0.5181,18.6674,1...
OOBPermutedPredictorDeltaMeanMargin	[]
OOBPermutedPredictorCountRaiseMargin	[]
DeltaCriterionDecisionSplit	[0.3285,8.9549,13...
NumPredictorSplit	[13.1111,14.8522,...
SurrogateAssociation	6x6 double
Proximity	
OutlierMeasure	
DefaultYfit	371.4024



	1 shadow_ratings	2 shadow_actual_price	3 shadow_percentage_discount
1	4.3000	299	50.0501
2	3.9000	999	78.0781
3	4.5000	199	62.5782
4	3.4000	269	60.0601
5	3.9000	999	22.2469
6	3.9000	499	55.6174
7	4.2000	599	39
8	3.6000	999	56.1162
9	3.4000	999	28.0330
10	2.9000	999	69.2308
11	3.6000	499	48.0480
12	3.7000	999	62.5782
13	5	749	45.7429
14	4	599	14.8581
15	4.4000	499	56.3204

Figure 4. Output from the Feature Selection Method.

3.4 Prediction Analysis

a. CNN-Bi-LSTM

CNN is a powerful technique in image processing. CNN has a remarkable track record of achievement in a variety of industries, which makes them a recommended choice for sales forecasting. CNN and other DL algorithms can identify patterns and other features in the input data. CNN was developed as an improvement to traditional statistical models, such as those used in sales forecasting. In general, a CNN consists of the following layers: input, convolution, pooling, complete connection, and output layer. Every layer uses a different mathematical method to reach a particular output at a threshold. Even though three convolution layers can offer the best performance [24], two bi-LSTM layers and one convolution layer are added in for the purpose of sales prediction. The CNN layers are illustrated in Figure. 5.

The Sequence Input Layer is the initial layer that receives sequential data, such as feature vectors with the format [batch_size, sequence_length, num_features]. A Convolutional 2D Layer then uses parameters like stride (e.g., 1), number of filters (e.g., 64), and kernel size (e.g., 3x3) to extract local spatial features. The ReLU Layer introduces non-linearity by setting negative values to zero and maintaining positive values. After that, the Max Pooling 1D Layer minimizes overfitting and lowers computational costs by pooling the maximum values within a specified window.In order to avoid overfitting, the Dropout Layer then employs regularization by periodically deactivating neurons. By processing data both forward and backward and merging past and future contexts with a predetermined number of units (128), a Bidirectional LSTM (BiLSTM) Layer recognizes long-range dependencies in the sequence. Each neuron from the preceding layer is then connected to the current layer by a Fully Connected Layer, which maps outputs to a vector of the required size (64 units).Non-linearity is introduced by a second ReLU Layer, and then more regularization is achieved via a Dropout Layer. The outputs for the regression job are refined by a second Fully Connected Layer. Lastly, the Regression Layer produces continuous output values, like predictions in regression tasks, and is usually optimized using MSE as the loss function.

Table 1. Hyperparameter Tuning of CNN

Training Parameters	Values
Learning Rate	0.0001
Optimizer	Adam
Number of Epochs	20
Number of Filters	3
Stride	1
Padding	0
Dropout	0.5

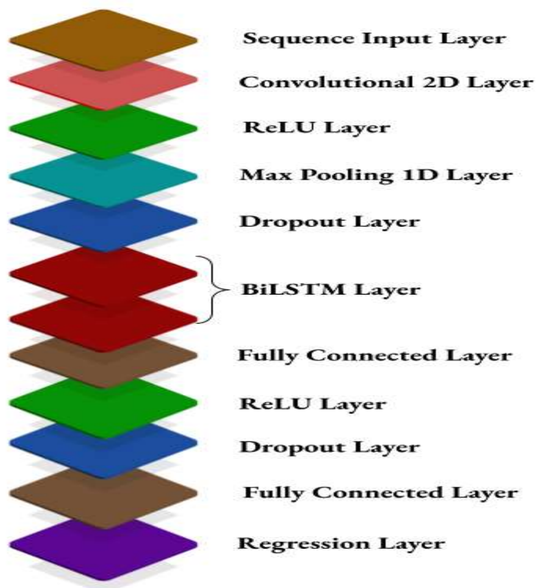


Figure.5 CNN-Bi-LSTM Network

b. Temporal Fusion Transformer (TFT)

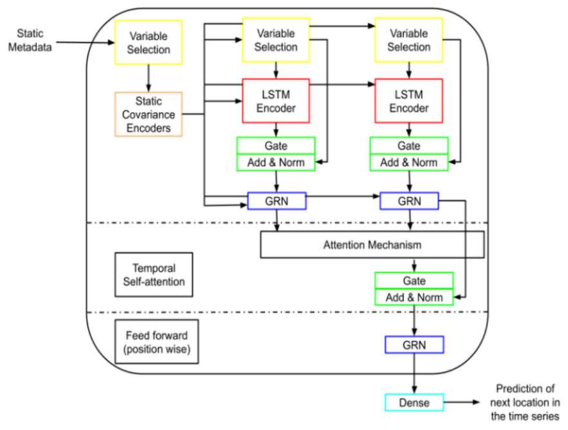


Figure 6. TFT Architecture

The TFT architecture is inspired in [26] as shown in Figure. 6. The approach constructed feature representations of the input type using standardized components.

The major constituent in the architecture

Gating Mechanisms provides adaptive depth in the architecture and the network complexity is increased, thereby increasing the number of datasets that can be analyzed. This part in the architecture uses a Gated Residual Network (GRN) as the primary functional block. In the initial stage, GRN considers an input a and a context vector c. The equation that governs a and c is illustrated in Eqn. 2

$$GRN_w(a, c) = Normalization(a + GLU_w(\mu_1) \quad (2)$$

$$\mu_1 = W_{1,w}\mu_2 + b_{1,w} \quad (3)$$

$$\mu_2 = LinearActivationUnit\left(W_{2,w}.a + W_{3,w}.c + b_{2,w}\right) \quad (4)$$

w is the weight shared. Linear unit is the "Exponential Linear Unit (ELU)" activation function. The gating layers are adopted based on Gated Linear Units (GLUs) that provides flexibility of the network architecture to omit the unwanted information in the dataset. The GLU can be represented as in Eqn. (4). The sigmoid activation function is represented as σ. The TFT can control how much the GRN influences the input thanks to the GLU. It can even get around the GRN layer if needed.The GRN considers that the context input is zero in the absence of a context vector (c = 0 in Eqn. 4). Dropout is applied to μ_1 in Eqn. 3 prior to the gating layer and layer normalization during training.

$$GLU = \sigma(W_{3,w}\gamma + b_{3,w}) \odot (W_{4,w}\gamma + b_{4,w}) \quad (5)$$

The TFT is designed to make instance-wise variable selection easier by extending variable selection methods to both time-dependent and static variables. Variable selection helps the TFT remove any unnecessary noise inputs that can negatively impact performance in addition to indicating which variables are most crucial for the prediction problem.

When working with a big number of variables, it's often unclear how significant each variable is to the result. The TFT uses instance-wise variable selection to address this. This implies that the most important variables are chosen for each individual instance, while irrelevant or noisy variables that can affect performance are left out. TFT uses two techniques: Entity Embeddings with respect to categories. linear conversions applied to continuous variables. TFT employs separate variable selection networks for prospective, historical, and static inputs. Each variable's relevance is determined by the variable selection weights, which are generated via a Softmax layer and a GRN. To extract significant features, a distinct GRN is applied to each input variable at each time step.

Eqn. 6 addresses the selection of variables. It illustrates how the importance of each variable at time t is determined. The context vector c_s (derived from static data) and the input variables Ξ_t, (a set of changed variables at time t) are processed by GRN. The output is normalized using the Softmax function, and then variable selection weights (V_s) are generated to tell the model how important each variable is to the prediction.

$$\tilde{\xi}_t = GRN_{\tilde{\xi}}(\xi_t), \in inputvariablej \quad (7)$$

Eqn. 7 mentions the nonlinear processing of the input variable ${\xi_t}^j$. When the input variable j goes through GRN in time t, it is denoted by GRN_{ξ}, which is used to extract the features and remove unnecessary information or noise. $\tilde{\xi}_t$ is the processed feature vector.

$$\tilde{\xi}_f = \sum_{j=1}^{m_x} v_s{}^j \tilde{\xi}^j{}_t \quad (8)$$

$\tilde{\xi}_f$ is the final processed vector. $v_s{}^j$ is the variable selection weight is combined with the processed feature vector $\tilde{\xi}^j{}_t$ to obtain the result.

To effectively integrate specific static metadata (i.e., data that doesn't change over time, such product category, customer type, etc.) in a model that examines time-based data (i.e., sales trends or stock prices over time).

Included are four GRN encoders that help create different types of context vectors. conceiving of these context vectors as specific, goal-driven summaries or representations of that static metadata. These context vectors are then passed into different parts of the temporal fusion decoder, which is a fancy part of the model that processes data depending on time. This is important because the use of static data at various stages affects how time-based patterns are interpreted.The context vector is,

i. c_s – decides the time-based variables to be selected.
ii. c_c and c_h - used to handle local processing of data, that helps to understand the time-based data in small intervals or chunks.
iii. c_e- enriches the time-based data with static information, influencing the overall understanding of the time-related features. The result of processing the data is collectively represented as ξ, which gets encoded to c_s. This model helps to decide which time-variable to focus on. In the previous research work, LSTM network was implemented in the encoder section. In the proposed approach, the LSTM is replaced by Bi-LSTM network and the efficiency of the prediction is analyzed.

The model learns long-term dependencies over several time steps with the use of a self-attention mechanism. By scaling the values V according to the connections between queries Q and keys K, which identify significant patterns in the data, the attention mechanism operates.The attention mechanism is shown as:

$$Attention(Q, K, V) = A(Q, K)V \quad (9)$$

where $A(Q, K)$ is a regularization function. The method used is scaled dot-product attention:

$$Regularizationfunction = Softmax\left(\frac{QK^T}{\sqrt{d_{attn}}}\right) \quad (10)$$

In Transformer, this is extended to multi-head attention, where different heads attend to various subspaces of the data. For each head h, attention is applied with its own learned weights for queries

$W_Q^{(h)}$, keys $W_K^{(h)}$, and values $W_V^{(h)}$:

$$Multihead(Q, K, V) = [H_1, \dots, H_{m_H}]W_H \quad (11)$$

where each head specific keys computes:

$$H_h = Attention(QW_Q^{(h)}, KW_K^{(h)}, VW_V^{(h)}) \quad (12)$$

Traditional multi-head attention is adjusted in TFT by giving each h

$$InterpretableMultiHead(Q, K, V) = \frac{1}{m_H}\sum_{h=1}^{m_H} Attention(QW_Q^{(h)}, KW_K^{(h)}, VW_V^{(h)}) \quad (13)$$

With the identical set of input data, this modification guarantees that every head concentrates on learning distinct temporal patterns. Furthermore, it improves the model's ability to more correctly represent complicated patterns and facilitates the interpretation of which components are crucial. Compared to typical multi-head attention, TFT's attention mechanism is easier to comprehend and evaluate when focusing on significant time steps and features.

The important time points are often related by the surrounding values, hence instead of using a single convolutional layer, Bi-LSTM sequence modelling is used to handle the past and the future inputs. Due to this uniform temporal features $\varphi(t, n)$ are created. The LSTM states are initiated by the static metadata and gated skip connection and is represented by Eqn. 14.

$$\varphi(t, n) = LayerNormalization(\tilde{\xi}_{t+n} + GLU(\varphi(t, n)) \quad (14)$$

Using GRN, the static metadata is appended to the temporal characteristics in the static enriched layer. The weights of the network are shared equally, represented as in Eqn. 15

$$\theta(t,n) = GRN_{\theta}(\varphi(t,n), c_e) \quad (15)$$

The temporal features are subjected to self-attention following the static enrichment layer. The model can learn correlations over extended time periods while preserving the sequence order by using masking (Eqn. 16):

$$B(t) = InterpretableMultiHead(\Theta(t), \Theta(t), \Theta(t)) \quad (16)$$

$\Theta(t)$is the static enriched temporal features. GRN analyses the self-attention layer's output.The gated residual connection can traverse the entire transformer block, if needed, to streamline the model as in Eqn. 17

$$\tilde{\varphi}(t,n) = LayerNormalization\left(\tilde{\varphi}(t,n) + GLU\left(\varphi(t,n)\right)\right) \quad (17)$$

The sequence-to-sequence LSTM layer handles the local patterns. Static enrichment combines temporal and static data. Self-attention records recurring patterns. A feed-forward layer is used to add further fineness.The model can more easily understand both the short- and long-term connections in time series data because to this combination.

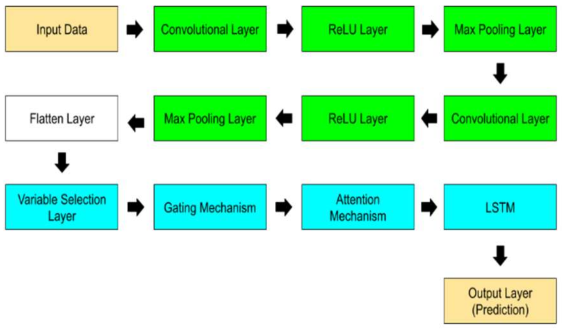


Figure 7. Proposed Architecture of Hybrid Approach

The hybrid approach for sales prediction is depicted in Figure. 7. Layer explanations are included below.

a. Input Layer

Raw sales information, including product features, rating count, and reduced price, is sent to the input layer. The incoming data's dimensionality is important since it affects the layers that follow. A time series of features, such as a collection of daily or weekly sales numbers, is commonly used as the input for sales forecasting, however this might vary. The number of characteristics utilized for prediction and the window size (30 days) determine the input shape. The feature selection technique is used to extract the features.

b. *Convolutional Layer*

The input data features are extracted in the convolutional layer. It's especially helpful for identifying local dependencies, such as patterns or seasonality, in time-series data. In this instance, the crucial hyperparameters are: 64 criteria are used to predict sales. A kernel size of 5, stride 1, and padding 0 are used to represent short-term dependency in time-series data. The output's size is managed using padding.

c. ReLU Layer

The Rectified Linear Unit (ReLU) layer causes the model to become non-linear. The activation function $ReLU(x) = max(0,x)$ is applied to the data to increase complexity and allow the model to learn more intricate associations. Although specific hyperparameter adjustment is not necessary for the ReLU activation, it significantly improves the network's performance and convergence rate.

d. MAX Pooling Layer

To save processing costs and prevent overfitting, max pooling layers shrink the input's spatial dimensions (or temporal in time-series). The pooling layer's primary hyperparameters are:

- Pool Size: The pool size determines the range of values that are taken. The dimensionality is cut in half when the pool size is two.
- Stride: Controls the movement of the pooling window. To prevent overlap in the pooling operation, it is frequently set to the same value as the pool size.

e. Flatten Layer

The 3D feature maps are transformed into a 1D vector by the flatten layer, and this vector is then fed into additional layers such as fully connected or LSTM layers. This stage is essential before the data enters dense or recurrent layers for prediction.

f. Gating Mechanism

The Gating Mechanism of the TFT regulates the flow of data throughout the network. It determines how important each step's input data is in relation to the others and how each feature affects the finished product. This keeps the model from overfitting and allows it to manage noisy or superfluous data. Typical components of the gating mechanism include the following:

- *Input Gating:* This gate determines which parts of the incoming data should be retained for further processing. It decides which features (such sales data, expenditure, etc.) should advance to the next layer or be suppressed.
- *Forget Gate:* This gate decides which information from previous time steps should be disregarded and is found in LSTMs. By preventing the network from "remembering" irrelevant data, it guarantees that only relevant historical data will be utilized in future projections.
- *Update Gate:* The update gate in the LSTM determines how much of the fresh data from the current input should be mixed with the historical data. It aids in the model's determination of the relative importance of historical and contemporary data.
- *Output Gating:* The output gate regulates the final output that is sent to the following layer of the network. It chooses how much of the present hidden state could be used for prediction or passed to the following layer.

g. Attention Mechanism

The model can focus on the most important time steps in the input sequence because of the TFT's Attention Mechanism. This is accomplished by assigning varying "attention weights" or degrees of relevance to different points in time, which aids in the model's identification of the most crucial historical data points for predicting future sales.

- *Context Vector:*A context vector, which is an overview of the most crucial details from the input sequence, is calculated by the attention process. Multiple time step data sets are integrated and given weights based on how important they are.
- *Attention Weights:* The model is trained with attention weights, which represent the relative relevance of specific time steps. For example, if recent sales data are more relevant to the present projection, they can be given greater weight than earlier sales data.
- *Scaled Dot-Product Attention:* One kind of attention that is commonly used in the TFT model is scaled dot-product attention.
  - It works by calculating the dot product of the keys (the previous time steps) and the query (the current time step).
  - Using the Softmax function and scale the output to avoid large gradients to obtain attention weights.
  - Using these weights to sum of weighted values are computed (historical data) to create the context vector.

h. LSTM

Decisions can be made using the network's long-term memory as well as its short-term recall of recent events. The RNN is unrolled. The input layer is at the bottom of the horizontally unfolded recurrent layer diagram, while the output layer is at the top. In a recurrent architecture, a unit layer is a cell that accepts inputs from the environment and from previous time cells, generates outputs, and sends data and outputs to cells that had been assigned beforehand. The cell state is the information content that moves across this network during time (as recurrent connections) and has a value of $c(t)$ at time t. The inputs and outputs from various cells would impact the cell's state as we navigate the network, or more accurately, over the temporal sequences in time. Likewise, the network establishes a recurrent link by sending the output $y(t)$ from one time to the next. In Figure. 8, the architecture is presented.

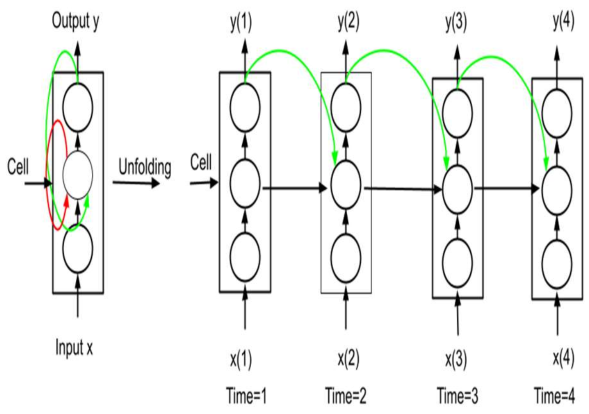


Figure.8. A Timed Sequence is fed into a RNN with a Cyclic Architecture, which Produces a Sequence as Output.

The first modification that needs to be made is the use of a forgetting gate. The weights have an impact on the external input $x(t)$ applied at time step t. A normal weighted addition is carried out in addition to the bias. The output from the recurrent connections is shown as $y(t-1)$ at time step $t-1$. However, sigmoidal activation is employed. Because the sigmoidal function generates outputs that range from 0 *to* 1, it controls the rate at which out-of-date material should be deleted. Forgetting happens when past information increases the amount of forgetting dependent on the new input. The gate is shown in Figure. 9.

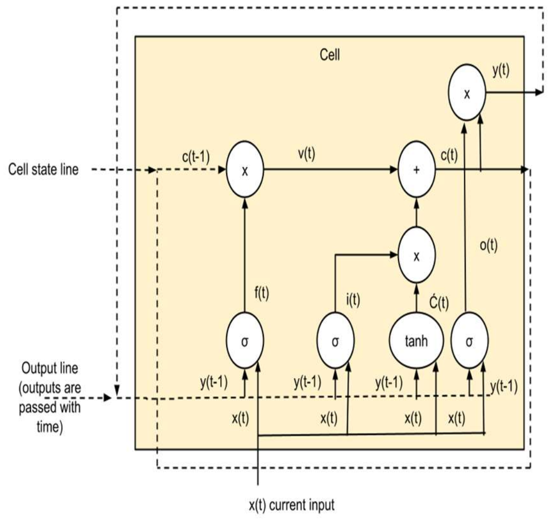


Figure. 9. General LSTM Architecture

The gate equations are given in Eqn. 18-19.

$$f_t = \sigma \sum_a w_f, x_a t + \sum_b w_f, y_b t - 1 + b_f \quad (18)$$

$$v(t) = ftct - 1 \quad (19)$$

The sigmoid activation function is denoted by σ. $v(t)$, which happens after forgetting (but before being impacted by the input), is a representation of the cell state. All recurrent connection inputs $(y(t-1))$ and external inputs $(x(t))$ undergo weighted addition in the first term. The bias is b_f. It is clear that all inputs—aside from the cell state—are sent to the forgetting gate. When a researcher gives a specific cell state access to the forgetting gate, an additional summing is performed across all the cell states. Peephole connections are the name given to these connectors. It should be noted that the cell state $c(t)$, which is a distinct entity, is not the same as the output of one cell, $y(t)$.

The second part of cell processing involves adding data from the new input to the cell state in order to change it as it moves. The input gate takes care of this. There are two ways the gate works. In the first, an activation function that is thought to be tangent hyperbolic is applied to a standard weighted addition. This activation function is useful since it may describe relations in both directions with values between -1 and 1. Not all inputs are helpful, though, and before an input can change the state of a cell, it needs to go through a check that determines how much of it should be used and precisely how much should be ignored immediately. The gate functions are modeled using equations (20)– (22).

$$i_t = \sigma \sum_a w_i, x_a t + \sum_b w_f, y_b t - 1 + b_i$$
(20)

$$ct = tanh\sigma \sum_a w_i, x_a t + \sum_b w_f, y_b t - 1 + b_{\acute{c}}$$
(21)

$$c(t) = v(t) + i_t \acute{c}^t = f_t ct - 1 + i_t \acute{c}t$$
(22)

In this instance, i_t represents the sigmoid function's preservation of the new weight's significance on a scale from 0 to 1. The input in the original state is represented by $\acute{c}t$. The summation begins with the external input, $x(t)$, and goes on to include the bias, $b_{\acute{c}}$, and the recurrent connections, $y(t-1)$. The contribution $\acute{c}t$ is combined to the forget value $v(t)$ to form the new cell state $c(t)$. The freshly converted input, $\acute{c}t$, with weight i_t, and the prior cell state, $c(t-1)$, with weight f_t, are thus the weighted sum of the "new cell state". (18)

Creating the neuron's output to be used as an outcome for the current time step is the final (19) stage. Cell output and state are two different things. Between unfolded stages, cell status and output must be calculated and transmitted. After passing through the activation function, which is tangent hyperbolic provides a range between -1 to 1, the output is a function of the cell state. To choose the pertinent state content for the output and suppress the rest, the sigmoid is applied based on the input. The gate is modeled using Eqn. (23)–(24).

$$o_t = \sigma \sum_a w_i, x_a t + \sum_b w_f, y_b t - 1 + b_o$$
(23)

$$y_t = o_t tanh c_t$$
(24)

The final output, y_t, is created by activation and weighted addition on the external inputs $x(t)$ and recurrent connections $y(t-1)$ after the activation function throughout the cell state, $c(t)$, is multiplied by the intermediate output, o_t. Backpropagation over time is used to train the network, much like an RNN [25].

i. Output (Prediction Layer)

This layer is responsible for creating the final forecast using the evaluated input data. The output represents future sales or demand. The output layer uses a linear activation for regression. It forecast incessant numerical data, like sales prediction. Output is optimized using a loss function that is appropriate for the task. Because they lessen the difference between projected and actual sales, MSE andMAE are widely used loss functions for sales prediction.

4. EVALUATION METRICS

- MAPE: The value of percentage error between the predicted and actual values is represented as MAPE. [28] mentions the clarification of typical MAPE values (table 2).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|y_i-\hat{y}_i|}{y_i}\right) x\ 100$$
(25)

Table 2 MAPE value

MAPE Value for Forecasting
•<10 - Highly accurate •10-20 - Good •20-50 - Reasonable •>50 - Inaccurate

- MSE: Calculates the mean absolute difference between sales values that were expected and those that were achieved.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$
(26)

- RMSE: Calculates the average squared difference between the expected and actual sales values and takes the square root of that difference.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$
(27)

- WMAPE: A variation of MAPE that adjusts for the scale of actual values, commonly used in sales data to avoid distortions caused by low sales values. For outstanding accuracy, a reasonable Weighted Mean Absolute Percentage Error (WMAPE) number is usually less than 10%. Table 3 shows the interpretation.

$$WMAPE = \frac{\sum|y_i-\hat{y}_i|}{\sum y_i}$$
(28)

Table 3 Interpretation of WMAPE Values

WMAPE	**Interpretation**
<5%	Excellent forecasting accuracy.
5-10%	Acceptable accuracy
10-20%	Moderate accuracy
>20%	Poor accuracy, suggesting the need for model improvement.

5. RESULTS AND DISCUSSION

Accuracy is the degree to which the model's predictions match the actual values. The percentage of instances in which the model correctly predicts sales trends, whether they are high or low, is known as accuracy in sales prediction. The model consistently matches real sales data when the accuracy number is higher. The outcomes display that the CNN-BiLSTM approach received 94%, TFT model received 95.6%, and the Hybrid Model received the best score of 97.25%. The higher accuracy of the Hybrid Model indicates its overall reliability in accurately forecasting sales patterns. Table 4 describes the performance metrics of the approaches in study.

Table4. Performance Metrics

Feature Selection	**Algorithm**	**Accuracy**	**Sensitivity**	**Specificity**
Boruta	**CNN [4]**	92.7	97.5	89.14
	CNN-Bi-LSTM	94	100	92.5
	TFT	95.6	98.2	97.5
	Hybrid Model	**97.25**	**100**	**100**

Table 5 Evaluation Metrics

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Feature Selection	Algorithm	MAPE	MSE	RMSE	WMAPE (%)
Boruta	CNN [4]	0.985	0.1025	0.5879	1.99
	CNN-Bi-LSTM	0.9261	0.09847	0.4298	2.71
	TFT	0.805	0.08481	0.2912	3.16
	Hybrid Model	**0.383**	**0.08260**	**0.1908**	**3.61**

Sensitivity, evaluates the model's capability to recognize true positive cases, such as spikes in demand or significant sales occasions. A high sensitivity model is particularly helpful when it's necessary to avoid missing times of high demand, which can impact inventory planning. The CNN-BiLSTM and Hybrid models detected all high-sales events with 100% sensitivity. This performance is crucial for applications that require accurate demand peak forecasting to maximize inventory levels and meet customer demand.

The specificity of the model shows how well it can identify bad circumstances, such as periods of lower-than-normal sales. To prevent overstocking, a model with high specificity accurately discerns between periods of normal and high demand. With 100% specificity, the Hybrid Model once again led, while TFT and CNN-BiLSTM got 97.5% and 92.5%, respectively. Because of its exceptional specificity, which demonstrates that it can reliably distinguish between high and regular sales phases, the Hybrid Model is highly effective at controlling inventory levels without experiencing excessive surpluses.

Table 5 gives the evaluation metrics of the proposed work. A lower MAPE score indicates better prediction accuracy. For these models, the CNN-BiLSTM generated the greatest MAPE of 0.9261, followed by the Hybrid Model at 0.383 and the TFT at 0.805. The low MAPE value of the Hybrid Model suggests that its sales projections are generally more accurate and less likely to differ from actual sales numbers. A lower MSE indicates more accurate projections with fewer severe errors. In this comparison, the CNN-BiLSTM model showed an MSE of 0.018473, the Hybrid Model a slightly lower MSE of 0.08260, while the TFT received a score of 0.08481. Despite having identical MSE values, the Hybrid Model outperforms TFT in terms of minimizing significant prediction errors.

The square root of MSE, or RMSE, has a comparable meaning in the same units as the original data, which facilitates comprehension. Like MSE, a lower RMSE denotes greater model performance in sales prediction. The TFT model scored 0.2912, the Hybrid Model had the lowest RMSE of 0.1908, and the CNN-BiLSTM model had an RMSE of 0.4298. The Hybrid Model is the most accurate forecasting model since it has the fewest average error magnitude.

The different prediction errors are weighted to account for their relative importance in WMAPE, a weighted version of MAPE. Because some products or periods may have higher sales volumes than others, sales forecasting is particularly useful in this regard. WMAPE ratings that are lower indicate better prediction accuracy. With a WMAPE of 3.61%, the Hybrid Model model outperformed the TFT model with a score of 3.16%, and the CNN-BiLSTM with a score of 2.71%, even after adjusting for variations in sales volumes.

The Hybrid Model outperformed the CNN-BiLSTM and TFT models in all metrics, including accuracy, MAPE, MSE, RMSE, and WMAPE, with flawless sensitivity and specificity. This implies that the hybrid model is the most effective model for accurately forecasting sales trends, minimizing errors, and generating forecasts that are valid over a variety of sales periods. Because of its exceptional performance in each of these areas, particularly for applications requiring precise demand forecasting and inventory management, it is the study's top choice for sales forecasting.

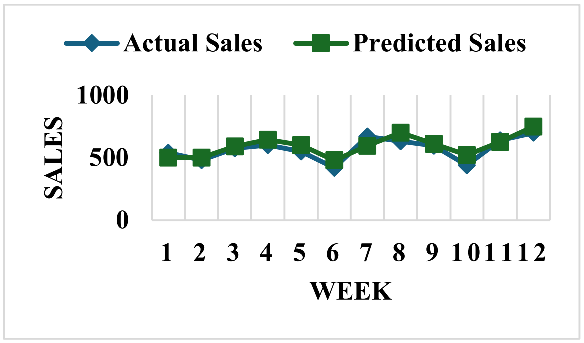


Figure. 10.Weekly Prediction

A comparison of actual and expected sales during a 12-week period is shown in Figure. 10. The weeks are displayed on the x-axis, which gives a clear study period. The labels are corresponding with Weeks 1–12. With a range of 0 to 800, the sales figures on y-axis indicate the figure of sales in a certain unit. Usually, this volume is stated in rupees. The blue line displays the actual weekly sales information. Its fluctuations show periods of higher and lower sales, which could be related to seasonal trends or responses to specific marketing campaigns. However, compared to the definite sales line, the orange line, which signifies foreseen sales based on a forecasting algorithm, usually has a smoother trajectory. Since predictions are meant to reflect broad patterns rather than the week-to-week volatility seen in real sales, they usually smooth the data. The graphic allows for a visual comparison of the two lines, providing insight into the prediction model's effectiveness. If the orange line consistently coincides with the blue line, then a strong predictive performance is demonstrated. However, a significant difference between the two lines suggests potential areas where the forecasting model needs to be improved.

The work focusses on improving fashion sales forecasting using historical transactional and temporal features through a hybrid CNN-TFT model. it does not address real-time streaming prediction, cross-platform or cross-category generalization, external data integration such as social media or macroeconomic indicators, or deployment-level constraints. These aspects are considered outside the scope of the present study and are identified as directions for future research.

A formal ablation study was conducted; the contribution of each architectural component can be inferred through controlled model comparisons already reported in the study. The data is tested on CNN as a standalone system, CNN-BiLSTM, TFT and the hybrid model. The CNN-BiLSTM model serves as a baseline for capturing local temporal relationships and variable selection. The observed performance gap between these two models indicates their complementary strengths and individual limitations.

When these components are integrated in the proposed hybrid CNN-TFT framework, a consistent improvement is observed across all evaluation metrics, including MAPE, RMSE, sensitivity and specificity. This performance gain implicitly demonstrates the additive contribution of CNN-based local feature extraction to the TFT's long-range temporal modeling. Moreover, the improved error stability during demand fluctuation periods in weekly prediction analysis suggests that the hybrid model benefits from multi-scale temporal representation rather than relying on a single modeling paradigm.

The findings are consistent with prior studies on DL-based sales forecasting in fashion retail. CNN-based models effectively capture short-term demand variations but struggle with long-term dependencies, while transformer-based approaches such as TFT model long-range temporal relationships yet under represent localized demand fluctuations in volatile datasets [4, 6, 12, 14]. The proposed hybrid CNN-TFT achieves lower error rates and improved sensitivity and specificity, supporting recent evidence that multi-scale temporal modeling enhances robustness in non-stationary retail data [9, 15]. Additionally, Boruta-based feature selection mitigates noise from high-dimensional e-commerce features, addressing a known limitation in existing studies [8, 27].

6. CONCLUSION

Sales forecasting is essential in retail and e-commerce because it provides businesses with the data, they need to control inventory, gauge demand, and optimize supply chains. Accurate sales forecasting helps prevent issues like overstocking or stockouts, which can have an impact on cash flow and customer satisfaction. Retailers can use advanced predictive models like CNN-BiLSTM and Temporal Fusion Transformers to manage the complexities of changing demand patterns and make data-driven decisions. As a result, firms may respond proactively to market shifts with the help of precise sales forecasting, which fosters long-term growth, resource efficiency, and increased customer satisfaction. The study illustrated the necessity of systematic data preparation and meticulous feature selection. The Boruta technique greatly reduced processing while maintaining the predictive power of the model by retaining only the most pertinent features. This method made sure the model could effectively handle high-dimensional data without overfitting.

The hybrid strategy outperformed the CNN-BiLSTM, Temporal Fusion Transformer (TFT), and CNN-BiLSTM models in terms of all assessment criteria, including MAPE, MSE, RMSE, and WMAPE. This hybrid model was able to better handle both short- and long-term sales trends by utilizing CNN for local feature extraction, BiLSTM for temporal dependencies, and TFT for interpretability and temporal fusion. The hybrid model achieved high sensitivity and accuracy, particularly in identifying demand spikes (sensitivity at 100%), an essential inventory management capability. Furthermore, specificity was perfect, minimizing the chance of overstock and successfully distinguishing times of high demand from normal ones. The low MAPE and RMSE indicated that this hybrid model produced the most accurate estimates, with little deviations from real sales data, even if the WMAPE confirmed the model's excellent performance across sales quantities.The highly accurate predictions of the hybrid model provide useful information for demand planning, reducing the risk of under- or overproduction. With its balanced accuracy throughout periods of high and low demand, retail managers may boost turnover rates, optimize stock levels, and even reduce waste.

Using more complex attention processes and extending the model to account for external market factors (such seasonality and promotions) could further improve predictive accuracy. Additional fine-tuning, including retraining the model on new datasets, helps maintain the model's relevance in changing market conditions. The hybrid model in this study is a good choice for sales forecasting in fashion e-commerce since it supports the decision-making process for inventory management, supply chain optimization, and demand forecasting with improved precision and interpretability. Because of this meticulous approach, the model is positioned as a helpful tool for data-driven business activities.

REFERENCES

[1]Kaunchi, P., Jadhav, T., Dandawate, Y.H., & Marathe, P. (2021). Future Sales Prediction for Indian Products Using Convolutional Neural Network-Long Short-Term Memory. 2021 2nd Global Conference for Advancement in Technology (GCAT), 1-5.

[2]Zhu, H. (2021). A Deep Learning Based Hybrid Model for Sales Prediction of E-commerce with Sentiment Analysis. 2021 2nd International Conference on Computing and Data Science (CDS), 493-497.

[3]Ganhewa, N.B., Abeyratne, S.M., Chathurika, G.D., Lunugalage, D., & De Silva, D.I. (2021). Sales Optimization Solution for Fashion Retail. 2021 3rd International Conference on Advancements in Computing (ICAC), 443-448.

[4]Buyar, V., & Abdel-raouf, A. (2019). A Convolutional Neural Networks-based Model for Sales Prediction. *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control*.

[5]Dai, Y., & Huang, J. (2021). A Sales Prediction Method Based on LSTM with Hyper-Parameter Search. *Journal of Physics: Conference Series, 1756*.

[6]Lim, B., Arik, S.Ö., Loeff, N., & Pfister, T. (2019). Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. *ArXiv, abs/1912.09363*.

[7]Sharma, S.K., Chakraborti, S., & Jha, T. (2019). Analysis of book sales prediction at Amazon marketplace in India: a machine learning approach. *Information Systems and e-Business Management, 17*, 261 - 284.

[8]Kasem, M.S., Hamada, M., & Taj-Eddin, I. (2023). Customer profiling, segmentation, and sales prediction using AI in direct marketing. *Neural Computing and Applications*, 1-11.

[9]Loureiro, A.L., Miguéis, V.L., & Silva, L.F. (2018). Exploring the use of deep neural networks for sales forecasting in fashion retail. *Decis. Support Syst., 114*, 81-93.

[10]Skenderi, G., Joppi, C., Denitto, M., &Cristani, M. (2021). Well Googled is Half Done: Multimodal Forecasting of New Fashion Product Sales with Image-based Google Trends. *ArXiv, abs/2109.09824*.

[11]Chen, T., Yin, H., Chen, H., Wang, H., Zhou, X., & Li, X. (2019). Online sales prediction via trend alignment-based multitask recurrent neural networks. *Knowledge and Information Systems, 62*, 2139 - 2167.

[12]Lai, G., Chang, W., Yang, Y., & Liu, H. (2017). Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval.

[13]Hochreiter, S., &Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9, 1735-1780.

[14]Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., & Cottrell, G. (2017). A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. ArXiv, abs/1704.02971.

[15]Chen, D.S., Liang, W., Zhou, K., & Liu, F. (2022). Sales Forecasting for Fashion Products Considering Lost Sales. Applied Sciences.

[16]Kumar, N., Yogesh, V., & Kavitha (2021). SALES PREDICTION ANALYSIS.

[17]Zhao, X., &Keikhosrokiani, P. (2022). Sales Prediction and Product Recommendation Model Through User Behavior Analytics. Computers, Materials & Continua.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[18]Chordiya, P.S., Shubham, M.A., Gayatri, M.A., Sumeet, M.K., & Harshada, M.M. (2022). Sales Prediction System using Machine Learning.

[19]Punjabi, S.K., Shetty, V., Pranav, S., & Yadav, A. (2020). Sales Prediction using Online Sentiment with Regression Model. *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 209-212.

[20]Chen, T., Yin, H., Chen, H., Wu, L., Wang, H., Zhou, X., & Li, X. (2018). TADA: Trend Alignment with Dual-Attention Multi-task Recurrent Neural Networks for Sales Prediction. *2018 IEEE International Conference on Data Mining (ICDM)*, 49-58.

[21]Khandelwal, D. (2022). Using Data Mining to Prediction Fashion Sales.

[22]Yi, S. (2023). Walmart Sales Prediction Based on Machine Learning. *Highlights in Science, Engineering and Technology*.

[23]Hazim, W.K., Amir, W.K., Bazlla, A., Soom, M., Mat, A., Ismail, J., Asmat, A., & Rahman, R.A. (2023). Sales Forecasting Using Convolution Neural Network. *Journal of Advanced Research in Applied Sciences and Engineering Technology*.

[24]Ghareeb, S., Mahyoub, M., & Mustafina, J. (2023). A comparative Time Series analysis of the different categories of items based on holidays and other events. *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, 131-136.

[25]Rahul Kala, Chapter 12 - An introduction to machine learning and deep learning, Editor(s): Rahul Kala, In Emerging Methodologies and Applications in Modelling, Autonomous Mobile Robots, Academic Press, 2024, Pages 569-625, ISBN 9780443189081, https://doi.org/10.1016/B978-0-443-18908-1.00022-4.

[26]Corrias, R., Gjoreski, M., &Langheinrich, M. (2023). Exploring Transformer and Graph Convolutional Networks for Human Mobility Modeling. Sensors (Basel, Switzerland), 23.

[27]Manikandan, G., Pragadeesh, B., Manojkumar, V., Karthikeyan, A., Manikandan, R., &Gandomi, A.H. (2024). Classification models combined with Boruta feature selection for heart disease prediction. Informatics in Medicine Unlocked.

[28]Montaño Moreno, J.J., Palmer Pol, A., Sesé Abad, A., & Cajal Blasco, B. (2013). Using the R-MAPE index as a resistant measure of forecast accuracy. Psicothema, 25 4, 500-6.