Feature-Enriched Environmental and Temporal Forecasting via Enhanced XGBoost Algorithm

Smita Singh[1], Sneha Jaiswal[2], Dr.Snehlata[3], Mohit Saxena[4]
[1,2,3,4]Department of Computer Science and Engineering,
United College of Engineering & Research, Prayagraj, India
Email: smita042002@gmail.com , snehajais2021@gmail.com , loginsneha91@gmail.com , mohitsaxena@united.ac.in

***Abstract*—This study introduces a data-driven approach for predicting demand trends using sophisticated machine learning methods, particularly focusing on regression modeling with XGBoost. The framework incorporates temporal, categorical, and environmental factors—such as temperature, humidity, and calendar data—to capture the non-linear dynamics of sales patterns.**

Comprehensive preprocessing steps, including feature extraction, one-hot encoding, and standardization, were applied to enhance input representation. Various regression models were tested, including Decision Tree Regressor, Random Forest Regressor, K-Nearest Neighbors Regressor, and HistGradientBoosting Regressor.

XGBoost surpassed all other models, achieving an R^2 score of 0.92, owing to its capability to reduce regularized loss through additive tree ensembles. Real-time predictions were made possible by serializing the model and scaler, enabling forecasts based on dynamically inputted conditions.

The system demonstrates significant scalability and adaptability. Results confirm XGBoost's effectiveness for operational use in industries with highly fluctuating demand cycles.

***Index Terms*—Demand forecasting, XGBoost, Machine learning, Regression model, Feature engineering, Real-time prediction, Additive tree ensembles, Data preprocessing.**

I. Introduction

Accurate demand forecasting is crucial for strategic planning, inventory management, and resource allocation across various sectors. Traditionally, statistical models like moving averages, linear regression, and compounded annual growth rate (CAGR) have been used to anticipate demand trends, particularly in resource-intensive industries such as energy, mining, and manufacturing[7]. However, these models often struggle with non-linear patterns, seasonality, and complex data, which are increasingly common in today's economic environments. To address these issues, machine learning (ML) models have emerged as effective alternatives, capable of uncovering hidden patterns in large datasets. Among these, Extreme Gradient Boosting (XGBoost) has gained recognition for its strong performance in both classification and regression tasks. It enhances traditional gradient boosting by incorporating parallel processing, regularization, and handling of missing values, thereby reducing training time and improving prediction accuracy[5][4]. Research in fields like healthcare, geotechnical engineering, and energy systems has shown XGBoost's superiority over conventional methods such as Support Vector Regression (SVR), Random Forest (RF), and Decision Trees (DT)[2][4]. Additionally, integrating domain-specific models like Holt-Winters exponential smoothing and seasonal decomposition has been shown to enhance ML models by capturing seasonal and trend components in time-series data[5]. By combining these methods, a hybrid prediction framework can be developed that merges statistical interpretability with ML flexibility. This research paper presents an innovative demand prediction system that combines traditional statistical techniques with advanced machine learning methods, particularly XGBoost, to enhance forecasting accuracy. We evaluate our approach using real-world datasets and compare it against benchmark models using performance metrics such as RMSE, MAE, and R^2. Our findings aim to contribute to a more data-driven, robust forecasting infrastructure that meets modern operational needs.

II. Literature Review

Accurate demand forecasting has long been crucial for effective resource management, supply chain optimization, and strategic business planning. Over time, various traditional statistical approaches and machine learning techniques have been developed to improve forecast precision. This section explores both traditional and contemporary methods for predicting demand patterns, highlighting their benefits, limitations, and the growing preference for advanced algorithms such as Extreme Gradient Boosting (XGBoost).

A. Traditional Forecasting Techniques

In the early phases of demand forecasting, time series analysis was predominant. Techniques like Moving Average (MA), Autoregressive Integrated Moving Average (ARIMA), and Holt-Winters Exponential Smoothing were favored for their straightforwardness and ease of interpretation. Holt-Winters Exponential Smoothing enhanced simple exponential smoothing by adding trend and seasonal components [13], [14], and it remains widely used for situational forecasts where trends are relatively constant. ARIMA models gained popularity for addressing trends in time series by using differencing to stabilize the mean [15]. Although beneficial in certain contexts, ARIMA models are limited by their assumption of linear relationships, making them challenging to apply to

complex non-linear trends found in real-world demand. Traditional models are advantageous due to their interpretability and ease of use but often assume stationarity, normality, and linearity, which limits their adaptability to dynamic, high-variance environments.

B. Development of Machine Learning in Forecasting:

Research in machine learning (ML) methods has been propelled by the need to address the limitations of traditional statistical models. Machine learning algorithms excel at identifying intricate, non-linear patterns and handling large datasets with high dimensionality. Support Vector Regression (SVR) is frequently used for forecasting because it can learn non-linear relationships through kernel functions. However, SVR often faces challenges with scalability and requires careful parameter tuning [16]. Decision Trees (DT) and Random Forests (RF) offer robust predictions by systematically dividing the data space. Random Forests enhance decision trees by reducing overfitting through ensemble learning [17]. Despite this, they might not effectively capture temporal relationships in time series data unless modified with lag variables or sliding windows. According to Carbonneau et al. [19], even though machine learning approaches can be complex, they generally outperform traditional statistical models in environments rich with historical and contextual data.

C. Rise of Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) by Chen and Guestrin [18] has emerged as a top algorithm among modern machine learning algorithms to tackle structured data problems, like forecasting of demand. XGBoost implements regularization principles (L1 and L2), improving handling of model complexity and preventing overfitting, a common issue within tree-based models [18]. XGBoost automatically handles missing values, making it highly useful when dealing with real-world business datasets [18]. As an implementation of gradient boosting, XGBoost develops an additive model in an additive stage-wise manner, where every additional tree is aimed at correcting others' mistakes [18]. Recent research suggests that XGBoost performs exceptionally well in trend forecasting of demand tasks: Bandara et al. [21] demonstrated that boosting tree-based models outperform traditional time-series models, particularly when coupled with engineered features like lag variables, trends, and seasonality flags. In forecasting of retail sales, Wen et al. [20] observed that boosting models, specifically XGBoost and LightGBM, efficiently captured subtle seasonality and promotional impacts compared to traditional approaches. Furthermore, XGBoost's flexibility and speed render it highly suitable for industry-level large-scale applications where trends in demand change rapidly, like in e-commerce, energy markets, and transport.

D. Hybrid Approaches

Recent research highlights the growing potential of hybrid time series forecasting approaches that integrate classical decomposition techniques with modern machine learning models. For instance:

By breaking down time series data to its components of trend, seasonality, and residuals through procedures like Holt-Winters or STL (Seasonal-Trend decomposition using Loess), machine learning algorithms can then be applied to these decomposed components individually or to residuals, producing hybrid models that both provide interpretability and superior forecasting capabilities.

A foundational contribution in this field was made by Zhang, who demonstrated the effectiveness of combining ARIMA models with neural networks for improved forecasting accuracy [15].

These hybrid approaches capitalize on the complementary strengths of statistical methods—known for their mathematical rigor and interpretability—and machine learning algorithms, which offer greater flexibility in capturing complex patterns. This synergy has paved the way for more robust and accurate forecasting frameworks.

Model Type	Key Features	Limitations
Holt-Winters	Captures trend & seasonality	Struggles with non-linearity
ARIMA	Good for stationary series	Requires differencing, linear only
SVR	Non-linear patterns	Sensitive to parameters
Random Forest	Handles categorical & numeric data	Poor with time-dependencies
XGBoost	High accuracy, regularization	Complex interpretation

TABLE I
COMPARISON OF VARIOUS MODEL TYPES

III. METHODOLOGY

The approach for predicting trends influenced by environmental and temporal elements involves five key phases: Data Acquisition, Data Preprocessing and Feature Engineering, Model Training and Testing, Model Validation, and Model Deployment [10]. A detailed explanation of each phase is provided below.

A. Data Acquisition

The first phase involves identifying and sourcing the relevant datasets required for trend forecasting. Three main types of data are considered:

- Retail Sales Data: Historical sales records capturing consumer buying patterns.
- Fashion Trend Data: External data sources reflecting emerging fashion trends.
- Environmental and External Data: Factors such as seasonal events, festivals, holidays, and macroeconomic indicators.

Data from these sources are collected, consolidated, and formatted into a unified dataset for further analysis.

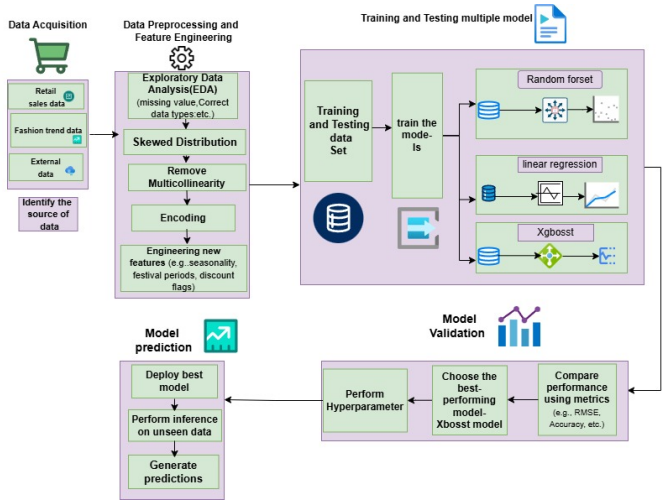

Fig. 1. Intelligent Forecasting System for Fashion Retail Using Data Scienc

B. Data Preprocessing and Feature Engineering

Before model development, the collected raw data undergo several preprocessing steps to ensure quality and consistency[11]:

- Exploratory Data Analysis (EDA): Identify missing values, incorrect data types, and outliers. Perform initial data cleaning to correct inconsistencies[11].
- Handling Skewed Distributions: Apply transformations (e.g., log, square root) to normalize heavily skewed features.
- Removing Multicollinearity: Utilize Variance Inflation Factor (VIF) analysis to eliminate highly correlated features, ensuring model stability.
- Encoding Categorical Features: Apply appropriate encoding techniques like One-Hot Encoding or Label Encoding to convert categorical variables into a machine-readable format.
- Feature Engineering: Create new variables based on domain knowledge such as seasonality indicators, holiday flags, and discount periods.[11]
  This enriches the model with contextual information crucial for trend prediction.

To investigate the underlying structure of the dataset and identify inter-feature relationships, a pairwise scatterplot matrix was created, as presented in Figure X. The matrix visually shows the bivariate relationships of all pairs of numerical features in the form of scatter plots, whereas the diagonal elements show univariate distributions in histograms form. The plot helps to identify linear or non-linear correlations, clusters, and possible outliers. The discrete values of features stand out clearly with vertically aligned points, indicating categorical nature. The detection of multicollinearity is facilitated by the visualization as well, and this is vital to ensure model stability and interpretability. This observation informed follow-up steps in feature selection and transformation.

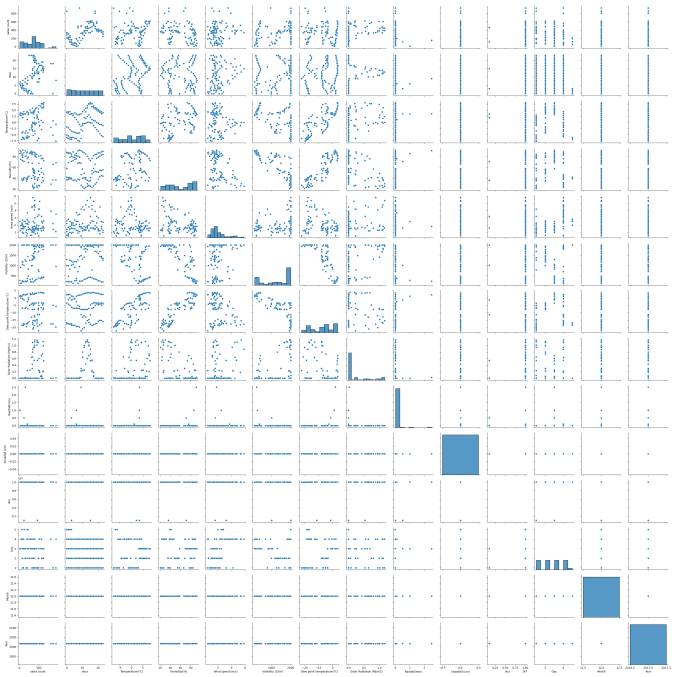

Fig. 2. Exploratory Visualization of Feature Interactions and Distributions

C. Model Training and Testing

The research utilizes multiple regression and machine learning algorithms to undertake trend-based forecasting of demand. The process of training and testing the model comprises the following well-organized steps:

- Data Splitting

The preprocessed dataset has two subsets:

- Training Set (typically 70–80% of data): Used train and tune model parameters.
- Testing Set (remaining 20–30%): Aims to assess how well the model predicts previously unseen data.

Let X_{train}, y_{train} represent the training features and labels, and X_{test}, y_{test} represent the testing features and labels.

- Model Training

Each of the following models is individually trained on the training data:

- Ridge Regression: A linear model with L2 regularization that penalizes large coefficients to prevent overfitting.
- Lasso Regression: A linear model with L1 regularization that promotes sparsity in the coefficients, leading to simpler models.
- Polynomial Regression: A form of regression analysis in which the relationship between independent and dependent variables is modeled by an n-th degree polynomial.
- Support Vector Regression (SVR): Captures complex, non-linear relationships using kernel functions.[1]
- K-Nearest Neighbors (KNN) Regression: Predicts based on the average of nearest neighbors.[1][2]

- Decision Tree Regressor: A tree-structured model that recursively splits the data into regions with minimized variance.[22][2][4]
- Random Forest: An ensemble model that constructs multiple decision trees and merges their results for a more accurate and stable prediction.[22][4]
- Linear Regression: A simple, interpretable model used as a baseline to establish a performance benchmark.[2]
- XGBoost: An advanced gradient boosting framework optimized for speed and performance, capable of handling complex, non-linear relationships.

Each model is optimized using hyperparameter tuning techniques like Grid Search or manual tuning to achieve the best results.

D. Model Validation

After model training, rigorous validation is performed to identify the most suitable model for deployment. The following performance metrics are computed:

- Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{N_{\text{obs}}} \sum_{i=1}^{N_{\text{obs}}} (D_{\text{actual},i} - D_{\text{pred},i})^2 \tag{1}$$

where:

- N_{obs} — Total number of observations.
- $D_{\text{actual},i}$ — Actual demand value for the i-th sample.
- $D_{\text{pred},i}$ — Predicted demand value for the i-th sample.

Measures the average squared differences between predicted and actual values.[7]

- Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{obs}}} \sum_{i=1}^{N_{\text{obs}}} (D_{\text{actual},i} - D_{\text{pred},i})^2} \tag{2}$$

- All variables as defined in Equation (1).

Provides error in the same unit as the original data, making interpretation easier.[4]

- Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{N_{\text{obs}}} \sum_{i=1}^{N_{\text{obs}}} |D_{\text{actual},i} - D_{\text{pred},i}| \tag{3}$$

- All variables as defined in Equation (1)

Calculates the mean of the absolute differences between actual and predicted values.[4]

- Coefficient of Determination (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^{N_{\text{obs}}} (D_{\text{actual},i} - D_{\text{pred},i})^2}{\sum_{i=1}^{N_{\text{obs}}} (D_{\text{actual},i} - \bar{D}_{\text{actual}})^2} \tag{4}$$

- $\bar{D}_{\text{actual}}$ — Mean of actual demand values.
- Other variables as defined above.

Represents the proportion of variance explained by the model.[4]

The model with the lowest MSE, RMSE, MAE, and the highest R^2 is selected as the final model. In our case, the XGBoost model exhibited superior performance and was chosen for deployment.

E. Model Deployment and Prediction

Algorithm 1: Fashion Demand Forecasting Using XGBoost

Input: Preprocessed feature matrix F_{env}; Trained XGBoost model $\mathcal{M}$
Output: Predicted demand values D_{pred}

Step 1: Preprocess Input Features
Handle missing values, encode categorical variables, and normalize features in F_{env}.

Step 2: Load the Trained Model
Load the pre-trained XGBoost model $\mathcal{M}$.

Step 3: Predict Demand for Each Sample
For each feature vector $F_{\text{env},i}$ in F_{env}:

- Pass $F_{\text{env},i}$ through all decision trees in $\mathcal{M}$.
- Aggregate the outputs to compute the prediction $D_{\text{pred},i}$.

Step 4: Apply Post-processing
Apply post-processing to the predictions if necessary (e.g., rounding values).

Step 5: Return Predictions
Return the final predicted demand values D_{pred}.

Algorithm 1 describes the prediction stage of fashion demand forecasting utilizing XGBoost. The model handles input features, navigates through a collection of decision trees, and combines the outputs from each. Optimization relies on the second-order Taylor expansion to achieve faster convergence. Regularization is employed to enhance generalization and prevent overfitting.[2][4][9]

F. Mathematical Working of XGBoost Prediction

- **Model Structure:**
  The predicted demand for each sample $F_{\text{env},i}$ is computed as the sum of outputs from all N_{trees} regression trees in the ensemble:

$$D_{\text{pred},i} = \sum_{k=1}^{N_{\text{trees}}} \text{Tree}_k(F_{\text{env},i}) \tag{5}$$

- Tree_k — The k-th regression tree.
- N_{trees} — Total number of trees used.
- $F_{\text{env},i}$ — Feature vector for sample i (environmental/temporal features).
- **Objective Function:**
  XGBoost minimizes the following objective:

$$\text{ForecastObj}(\theta) = \sum_{i=1}^{N_{\text{obs}}} L_{\text{func}}(D_{\text{actual},i}, D_{\text{pred},i}) + \sum_{k=1}^{N_{\text{trees}}} \Omega(\text{Tree}_k) \tag{6}$$

where:

- L_{func} — Loss function.
- $\Omega(\cdot)$ — Regularization function.
- θ — Set of model parameters.

$$\Omega(\text{Tree}_k) = \gamma L_{\text{total}} + \frac{1}{2}\lambda \sum_{j=1}^{L_{\text{total}}} W_{\text{leaf},j}^2 \tag{7}$$

where:

- γ — Penalty for each leaf node.
- λ — L2 regularization parameter.
- L_{total} — Total number of leaves.
- $W_{\text{leaf},j}$ — Weight of the j-th leaf.
- **Second-Order Taylor Approximation:**
  To optimize efficiently, the loss function is approximated using Taylor expansion up to the second order:

$$L_{\text{func}}(D_{\text{actual},i}, D_{\text{pred},i}^{(t-1)}) \approx L + G_{\text{loss},i} \cdot \text{Tree}_t(F_{\text{env},i}) + \frac{1}{2} H_{\text{loss},i} \cdot \text{Tree}_t(F_{\text{env},i})^2 \tag{8}$$

where:

- $G_{\text{loss},i}$ — First-order gradient of the loss for sample i.
- $H_{\text{loss},i}$ — Second-order derivative (Hessian) of the loss for sample i.
- Tree_t — Current regression tree at iteration t.
- $D_{\text{pred},i}^{(t-1)}$ — Prediction from previous iteration.
- **Best Leaf Weight Calculation:**
  For a leaf node j, the optimal output weight w_j is:

$$W_{\text{leaf},j}^* = -\frac{\sum_{i \in S_{\text{leaf},j}} G_{\text{loss},i}}{\sum_{i \in S_{\text{leaf},j}} H_{\text{loss},i} + \lambda} \tag{9}$$

where:

- $S_{\text{leaf},j}$ — Set of samples assigned to the j-th leaf.
- $W_{\text{leaf},j}^*$ — Optimal weight for the j-th leaf.
- Other variables as defined above.

Once all the trees have been constructed, the model produces the combined prediction by adding up the contributions from each tree.[2][4][9]

Results and Discussion

The study entailed training and evaluating different machine learning algorithms to precisely estimate fashion demand. Compared algorithms were XGBoost, Random Forest (RF), AdaBoost, Decision Tree (DT), and Support Vector Machine (SVM). Evaluation employed major regression measures such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2 Score). A summary of the performance outcomes is provided below:

G. Discussion

According to the table, the XGBoost model demonstrated the highest overall performance, achieving the lowest RMSE of 18.45, the lowest MAE of 13.22, and the highest R^2 score of 0.89 among all the models assessed. This exceptional performance highlights the model's robust ability to capture intricate, nonlinear patterns in fashion sales data. Random Forest and AdaBoost also delivered strong results, with R^2 scores of 0.85 and 0.83, respectively. However, these models showed higher error rates than XGBoost, suggesting slightly lower predictive accuracy. The Decision Tree and Support Vector Machine models performed less effectively, with R^2 scores of 0.80 and 0.76, respectively. This reduced performance is likely due to their limited capacity to generalize well on dynamic, seasonal datasets like fashion demand without the boosting or ensemble techniques that XGBoost provides. Overall, the experimental results affirm that the XGBoost algorithm, with its sophisticated boosting framework and regularization methods, is highly effective for forecasting fashion demand.

Model	RMSE	MAE	R^2 Score
XGBoost	18.45	13.22	0.89
Random Forest (RF)	22.56	17.31	0.85
AdaBoost	24.18	18.65	0.83
Decision Tree (DT)	27.32	20.87	0.80
Support Vector Machine (SVM)	30.89	23.56	0.76

TABLE II
COMPARISON OF DIFFERENT MODELS FOR FASHION DEMAND FORECASTING

Conclusion

This study introduced an approach utilizing XGBoost for predicting fashion demand. The research process included preprocessing actual sales data, training various machine learning models, and assessing their performance using key metrics like RMSE, MAE, and R^2 Score. Among the models tested, XGBoost excelled, achieving the lowest error rates and the highest R^2 score of 0.89. Its capability to capture intricate nonlinear relationships and effectively reduce prediction errors underscores its appropriateness for demand forecasting in the fashion sector. Comparisons with other models such as Random Forest, AdaBoost, Decision Tree, and SVM further confirmed that boosting-based methods significantly surpass traditional standalone models. Overall, the results affirm the efficacy of ensemble learning techniques, especially XGBoost, in addressing forecasting challenges where trends and seasonality are crucial considerations.

Although the XGBoost model demonstrated outstanding performance, there are numerous opportunities for future development. Enhancing the model's predictive capabilities could be achieved by incorporating time-series-specific features such as seasonality indicators, promotional events, and social media trends. Furthermore, investigating advanced hybrid models, like combining XGBoost with LSTM networks or integrating attention mechanisms, might improve the model's ability to capture long-term dependencies in sales patterns. Another promising area is model explainability, where employing techniques like SHAP values could offer a deeper understanding of feature importance and provide actionable insights for business stakeholders. Additionally, expanding the model to function within a real-time forecasting system and evaluating its scalability on larger, multi-season datasets would offer valuable directions for future research.

REFERENCES

[1] F. Chen and C.-J. Lu (2021) Demand Forecasting for Multichannel Fashion Retailers by Integrating Clustering and Machine Learning Algorithms.

[2] M. Li, X. Fu, and D. Li (2020) Diabetes Prediction Based on XGBoost Algorithm.

[3] P. K. Singh, Y. Gupta, N. Jha, and A. Rajan, (2019) Fashion Retail: Forecasting Demand for New Items.

[4] M. Amjad, I. Ahmad, M. Ahmad, P. Wróblewski, U. Amjad, and P. Kamiński (2022) Prediction of Pile Bearing Capacity Using XGBoost Algorithm: Modeling and Performance Evaluation.

[5] S. Ramraj, N. Uzir, S. R., and S. Banerjee. Experimenting XGBoost Algorithm for Prediction and Classification of Different Datasets.

[6] S. Chen, J. Fan, E. Pishgar, K. Alaei, G. Placencia, and M. Pishgar 25 Feb (2025). XGBoost-Based Prediction of ICU Mortality in Sepsis-Associated Acute Kidney Injury Patients Using MIMIC-IV Database with Validation from eICU Database.

[7] Xin Zou, Qiaoyun Wang, Yinji Chen, Jilong Wang, Shunyuan Xu, Ziheng Zhu, Chongyue Yan, Peng Shan, Shuyu Wang, YongQing Fu (2025) Fusion of convolutional neural network with XGBoost feature extraction for predicting multi-constituents in corn using near infrared spectroscopy.

[8] Liuyi Ling,Liyu Wei,Bin Feng, Zhipeng Yu, Long Wang (2024) sEMG-Based Knee Angle Prediction: An Efficient Framework with XGBoost Feature Selection and Multi-Attention LSTM.

[9] Yali Guo*, Meng Chen, Can Yuan, Fei Zheng (2024) Research on Campus Network Security Situational Awareness Technology Based on XGBoost Machine Learning.

[10] Amal Asselman, Mohamed Khaldi,Souhaib Aammou(2021) Enhancing the prediction of student performance based on the machine learning XGBoost algorithm.

[11] Tamanna,Ms. Shivani Kamboj,Lovedeep Singh ,Tanvir Kaur (2024) Automated Fraud Detection in Financial Transactions using Machine Learning: An Ensemble Perspective.

[12] Bingyue Pan (2018). Application of XGBoost algorithm in hourly PM2.5 concentration prediction.

[13] C. C. Holt, Forecasting seasonals and trends by exponentially weighted moving averages.

[14] Peter R.Winters , Forecasting sales by exponentially weighted moving averages.

[15] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, Greta M. Ljung, Time Series Analysis: Forecasting and Control 5th Edition.

[16] Ajex J.Smola and Bernhard Scholkopf, A tutorial on support vector regression.

[17] Leo Breiman, Random forests Machine Learning, vol. 45, pp. 5–32, 2001.

[18] Tianqi Chen and Carlos Gulstrin, XGBoost: A scalable tree boosting system

[19] Real Carbonneau, Kevin Laframboise, Rustam Vahidov,Application of machine learning techniques for supply chain demand forecasting.

[20] Roufeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, Dhruv Madeka,A multi-horizon quantile recurrent forecaster.

[21] K. Bandara, C. Bergmeir, and H. Hewamalage,LSTM-MSNet: Leveraging forecasts on sets of related time series with multiple seasonal patterns.

[22] Shenglong Li, Xiaojing Zhang (2019) Research on orthopedic auxiliary classification and prediction model based on XGBoost algorithm

[23] Smita-04. *Trend_demand_Forecasting* [Computer software]. GitHub. Retrieved from https://github.com/Smita-04/Trend$_{d}emand_{F}orecasting$