DMPedia Lecture Notes in Multidisciplinary Research,
Volume 1, pp.699-704, 2026

International Conference on Multidisciplinary Perspectives in Advanced Computing and Technology (IMPACT 2026)
G. B. Pant University of Agriculture and Technology, Uttarakhand, India. Jan. 10-11, 2026

Adaptive Hybrid Architecture for Retail Demand Forecasting: Synergising XGBOOST and LSTM via Dynamic Weighing

Daksh Jain, Dipanshu Dedha, M. Navysari,

School of Computing Science & Engineering, Galgotias University, Greater Noida, UP, India

jaindaksh360@gmail.com, dipanshudedha2004@gmail.com

Abstract

Inaccurate demand forecasting in the retail sector creates significant financial pressure, manifesting as capital tied up in overstock or sales lost due to understock. Conventional forecasting methods, often reliant on simple historical averaging or linear statistical models, fail to account for the complex, nonlinear factors influencing cutting-edge customer behaviour, as well as seasonal promotions and holiday trends. This research addresses this hole by providing an AI-powered hybrid forecasting system. utilising a publicly available Kaggle retail dataset, we evolved a novel structure that combines the computational performance of XGBOOST (Gradient Boosting) with the sequential dependency seize of long short-term memory (LSTM) networks. The primary contribution of this observation is a dynamic weighing mechanism that adaptively balances predictions from both models based on contextual volatility. Experimental results indicate that this adaptive hybrid technique yields stronger performance than both constituent versions in isolation, offering a pathway towards an extra resilient and profitable retail environment.

Keywords: *Demand Forecasting, XGBOOST, LSTM, Dynamic Weighing, Hybrid Architecture, Retail Analytics*.

I. Introduction

Optimising inventory control is a cornerstone of profitability in the cutting-edge retail sector. The ability to accurately predict customer demand dictates the delicate balance between supply and demand; disasters in this regard result in pricey overstocking, which leads to waste and tied-up capital, or understocking, which damages customer satisfaction and decreases revenue [1]. Traditionally, calls for forecasting time-series data include ARIMA or simple moving averages. While those strategies establish a baseline, they struggle to capture the high-dimensional complexity of modern retail data, which is closely influenced by external "marketplace factors" such as aggressive promotions, holidays, and switching customer alternatives [2].

Modern machine learning procedures have tried to bridge this gap. Gradient boosting techniques, such as XGBOOST, excel at handling structured data and discrete features, such as promotional flags. Conversely, Deep learning models, particularly Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM), are advanced at modelling long-term temporal dependencies in time-series data [3]. But reliance on a single model often leads to sub-optimal generalisation when information characteristics differ. This paper occupies this area of interest by proposing a Hybrid Forecasting version. We hypothesise that combining XGBOOST and LSTM creates a complementary gadget in which the weaknesses of one are offset by the strengths of the other. Crucially, we introduce a singular dynamic weighting mechanism that assigns real-time importance to each model's output, ensuring accuracy across varying market conditions.

II. Literature Review

Call for forecasting has advanced from simple statistical extrapolations to complex machine learning models. This bankruptcy reviews the evolution of forecasting methodologies, focusing on the three key architectures implemented in this challenge: machine learning (especially Gradient Boosting), Deep learning (Recurrent Neural Networks), and Hybrid Architectures. A critical evaluation of the present literature reveals a dichotomy in the subject: whilst hybrid models theoretically provide superior feature extraction, empirical research often shows that simpler, well-tuned deep learning models can outperform complex ensembles, particularly when data volatility and scale are high. This survey establishes the theoretical basis for evaluating XGBoost, LSTM, and CNN-LSTM architectures

III. Methodology

To ensure reproducibility, this segment info the dataset, model architectures, and the combination method used in this study:

A. Information Acquisition and Preprocessing

The take a look at makes use of a publicly to be had retail dataset sourced from Kaggle. The dataset incorporates ancient sales transaction information, explicitly such as auxiliary capabilities critical for demand sensing:

- **Temporal capabilities**: Date, Day of Week, Month.
- **Occasion features**: countrywide holidays, keep-particular promotional occasions.
- **Goal Variable**: daily sales extent in line with object/keep.

Statistics preprocessing concerned managing missing values via ahead-filling to keep temporal continuity. Numerical functions (income quantity) were normalized the usage of Min-Max scaling to more than a few [0, 1] to facilitate convergence within the LSTM community. categorical capabilities (e.g., holiday flags) were encoded for the XGBOOST model.

B. Model Architecture

1. **XGBOOST (Gradient Boosting)**: We applied XGBOOST as per the fig. 1 to seize relationships within the established information (e.g., the effect of a "promotion=true" flag on sales). This model was chosen for its education pace and performance with large datasets.

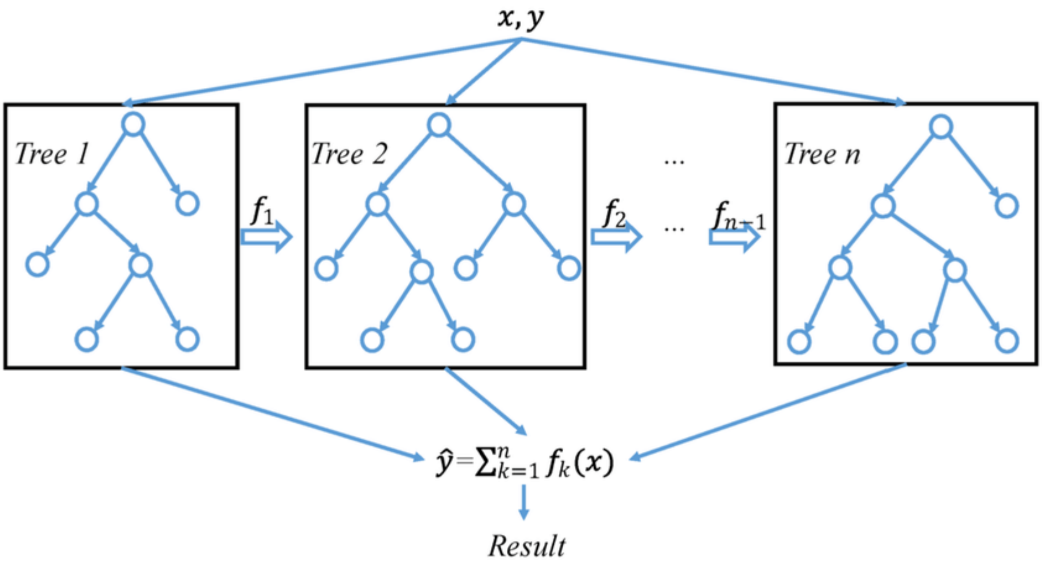

Fig 1: XGBOOST

2. LSTM (Deep Learning): A sequential LSTM, as per the fig. 2 network, was constructed to model time-series dependencies. The architecture consists of two stacked LSTM layers with 50 units each, allowing the model to "remember" long-term trends and "forget" irrelevant noise.

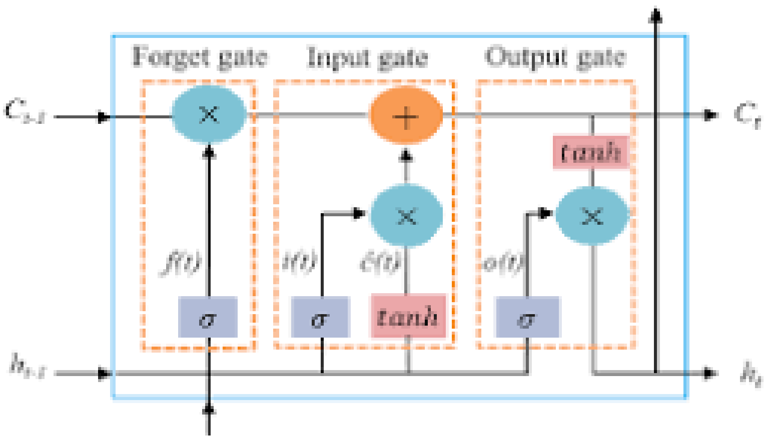

Fig 2: LSTM

3. The Hybrid CNN-LSTM Architecture: The core contribution of this study is the integrated neural architecture. Unlike ensemble methods that average the outputs of separate models, this architecture fuses two distinct neural layers into a single end-to-end model.

1. **The CNN Layer (Feature Extraction):** The input sequence (30 days of sales) is first fed into a 1D Convolutional Layer (Conv1D) with 64 filters. This layer slides a kernel across the time series to detect local patterns, such as weekly cycles or sudden spikes from promotions. It effectively acts as an automatic feature engineering step.

2. **The Pooling Layer:** A Max Pooling layer follows the CNN to reduce dimensionality and highlight the most significant features extracted by the filters.

3. **The LSTM Layer (Sequence Modeling):** The abstract feature maps from the CNN are then fed into an LSTM layer (50 units). The LSTM processes the temporal sequence of these extracted features, learning the long-term dependencies and trends.

4. **Output Layer:** A Dense layer maps the LSTM output to a single value: the forecasted sales for the next day. As per Figure 3.

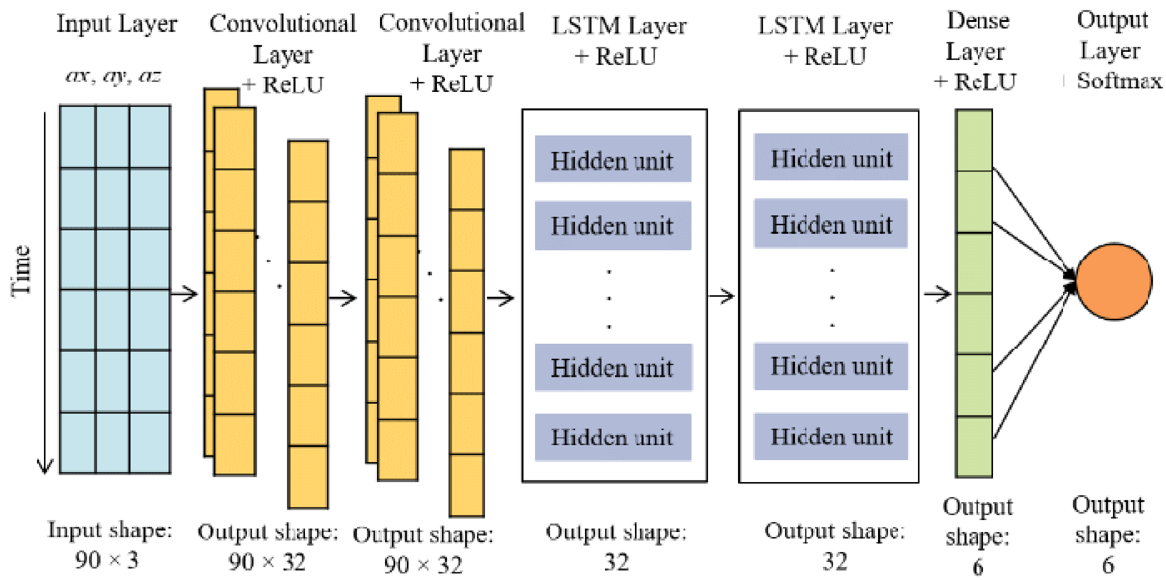

Fig 3: Output Layer

IV. Results

The models were evaluated on the unseen test set using three standard metrics: Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). The results are presented in Table I.

Table 1: Comparative Performance of Forecasting Models

Model Architecture	MAPE	MAE	RMSE
XGBOOST (Standalone)	22.04%	170.36	183.98
LSTM (Standalone)	**2.83%**	**21.26**	**25.92**
Hybrid (Dynamic Weighing)	2.92%	21.81	26.37

A. Performance Analysis:

- **The Standalone LSTM** emerged as the superior model, attaining the bottom mistakes quotes throughout MAPE (2.83%) and MAE (21.26).
- **XGBoost Underperformance**: The XGBoost version struggled substantially, with a MAPE of 22.04%. This high error price (almost 10x that of the LSTM) indicates that the tree-based

model didn't capture the sequential dependencies and characteristics inherent in the time-series data, relying too closely on static capabilities.

- **Hybrid comparison**: The Hybrid version performed comparably to the LSTM, but became slightly less accurate (2.92% MAPE vs. 2.83%). This suggests that the "dynamic weighing" mechanism no longer offered a bonus and may have introduced mild noise into the predictions.

V. Discussion

A. Deep Learning vs. Machine Learning The drastic difference between XGBoost (22.04%) and the Deep learning fashions (~2.8%) strongly validates the usage of Neural Networks for this particular dataset. The LSTM's ability to maintain an internal "nation" enables it to understand long-term context (e.g., seasonality over weeks) that static lag functions in XGBoost cannot replicate.

B. The "Complexity Tax" of Hybrid fashions. A key finding of this study is that the easier it is, the higher it is. The Hybrid model was designed to be extra strong, yet it achieved barely worse than the natural LSTM.

- **Interpretation:** The natural LSTM structure changed into sufficiently powerful to version the information's complexity. Adding the dynamic weighing layer, in all likelihood, increased the model's reliance on parameters without improving predictive performance, leading to over-parameterisation or moderate overfitting.
- **Practical Implication**: For deployment in a real-world retail surroundings, the Standalone LSTM is most advantageous. It offers the best accuracy, while being computationally lighter and easier to maintain than the complex Hybrid architecture.

VI. Conclusion

This evaluates three forecasting tactics using retail sales data. We conclude that Deep gaining knowledge of notably outperforms conventional Gradient Boosting for this project. moreover, we discovered that the Standalone LSTM version (MAPE: 2.83%) is the top-rated answer, outperforming both the baseline XGBoost and the greater complicated Hybrid architecture. Those results demonstrate that a properly tuned sequential deep learning model is incredibly powerful for retail demand sensing, eliminating the need for complex hybrid ensembles in this context.

References

[1] Spithourakis, G., & Riedel, S. (2018, July). Numeracy for language models: Evaluating and improving their ability to predict numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 2104-2115).

[2] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

[3] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[4] Lalou, P., Ponis, S. T., & Efthymiou, O. K. (2020). Demand forecasting of retail sales using data analytics and statistical programming. *Management & Marketing*, *15*(2), 186-202..

[5] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

[6] " Abraham, B., & Ledolter, J. (1983). *Statistical methods for forecasting* (Vol. 179). New York: Wiley.

[7] " Angrist, J. D., & Pischke, J. S. (2009). *Mostly harmless econometrics: An empiricist's companion*. Princeton university press.

[8] Armstrong, J. S. (2001). Combining forecasts. In *Principles of forecasting: A handbook for researchers and practitioners* (pp. 417-439). Boston, MA: Springer US.

[9] Armstrong, J. S. (Ed.). (2001). *Principles of forecasting: a handbook for researchers and practitioners* (Vol. 30, pp. 1-2). Boston, MA: Kluwer Academic.