International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 11-No.3 (2025) pp. 4268-4276

http://www.ijcesen.com

ISSN: 2149-9144

Research Article

Enhancing Accuracy in Recommender Systems with a Hybrid Deep Learning Approach for Web Usage Mining

Supriya Saxena[1*], Bharat Bhushan[2]

[1]Research Scholar, Department of Computer Science and Engineering, Sharda University, Greater Noida, India
*** Corresponding Author Email:** supriya08.saxena@gmail.com - **ORCID**: 0009-0005-5967-2447

[2]Associate Professor, Department of Computer Science and Engineering, Sharda University, Greater Noida, India
Email: bharat_bhushan1989@yahoo.com - **ORCID:** 0000-0002-9345-4786

Article Info:

DOI: 10.22399/ijcesen.2845
Received : 22 March 2025
Accepted : 12 June 2025

Keywords

Recommender System
Deep Feedforward Neural Network
Long Short-Term Memory
Sequential Learning
Web Usage Mining
Algorithm Efficiency

Abstract:

The rapid expansion of e-commerce platforms has created an urgent demand for intelligent recommender systems capable of providing personalized and context-aware product recommendations. While traditional recommendation models offer some effectiveness, they often face challenges such as the cold start problem, data sparsity, and the inability to capture sequential user behavior. This paper presents a hybrid deep learning-based recommendation architecture that combines a Deep Feedforward Neural Network (DFNN) with a Long Short-Term Memory (LSTM) network to overcome these limitations. The proposed hybrid model was evaluated against standalone DFNN and LSTM models using a real-world e-commerce clickstream dataset. The results demonstrate the hybrid model's superior performance, achieving a training accuracy of 99.06% and validation accuracy of 98.98%. Additionally, the model excels in classification performance, with precision, recall, and F1-scores approaching 1.00 across critical user actions such as add_to_cart, purchase, search, and view.

1. Introduction

The rise of digital platforms for shopping, whether through websites or mobile applications, has increased the demand for customer-oriented systems. In this context, recommendation systems have become essential in meeting these needs, as they are designed to cater to user preferences and enhance the overall customer experience. E-commerce platforms generate vast amounts of data related to diverse product choices, and the introduction of new businesses requires valuable customer feedback to improve service quality. This feedback can be effectively gathered by analyzing user behavior while interacting with these platforms. Recommendation systems offer an ideal solution by personalizing product suggestions and improving customer engagement. Additionally, they can be used strategically to appeal to sustainability-conscious consumers, thus aiding businesses in making more informed sales decisions and fostering long-term customer relationships [1]. A Recommendation System plays a vital role in identifying potential and well-suited choices for users by gaining insights into their preferences and behaviors. These insights can be gathered through various means, including user ratings, reviews, purchase history, and interaction timestamps. The application of recommendation systems in e-commerce is essential for business sustainability, particularly in maximizing profitability. By providing personalized recommendations, such systems help enhance customer loyalty and satisfaction, which are crucial for business growth and retention [2]. In-depth knowledge of customer preferences and behaviors enables businesses to engage more effectively with their target audience. Recommendation systems take various forms, each contributing uniquely to improving user experience and business outcomes, with the most important approaches discussed in the following subsections.

1.1 Collaborative Recommender system

The concept of a Collaborative Recommendation System is grounded in the principle of identifying users with similar shopping behaviors and leveraging their past interactions to suggest relevant products or webpages to new users. This approach primarily relies on historical user-item interaction data—such as ratings or purchase history—to identify patterns and predict user preferences. Collaborative filtering (CF) algorithms utilize these historical scores to recommend items liked by similar users, based on the assumption that users who agreed in the past will likely agree in the future. This is often represented as a function of shared preferences or behavioral trends. Two primary variations of CF algorithms are widely adopted: user-based and item-based collaborative filtering. These techniques are particularly effective for developing personalized, user-centric recommendation systems in e-commerce. The core principles underlying CF involve recognizing similar user preferences, maintaining consistency in user choices, and analyzing historical interaction data, making collaborative filtering a foundational approach in recommender system design [3].

1.2 Content-based recommender system

Content-based recommender systems play a crucial role in providing personalized suggestions by analyzing items that a user has previously engaged with or expressed interest in. These systems rely heavily on the attributes of items the user has liked in the past, and recommend similar products based on those features. The recommendation process is inherently content-driven, focusing on matching the user's preferences with comparable product characteristics [4].

1.3 Hybrid recommender system

A Hybrid Recommender System combines multiple recommendation techniques, such as Collaborative Filtering (CF), Content-Based Filtering, and Context-Aware Systems, to enhance recommendation accuracy and personalization. By integrating various methods, hybrid systems leverage the strengths of each while mitigating their individual weaknesses, leading to more reliable, diverse, and relevant recommendations. Common hybrid approaches include weighted hybrid, where methods are combined based on assigned weights, and switching hybrid, where different techniques are applied depending on specific user conditions. Mixing hybrid applies different algorithms to distinct user or item subsets, while feature augmentation hybrid enhances predictions by using one method's output as input for another. These systems offer significant advantages in accuracy, diversity, and personalization, particularly in e-commerce, media streaming and social networking platforms, where tailored recommendations are crucial. However, they also come with challenges such as increased complexity and scalability issues, requiring careful design and computational resources to maintain efficiency.

The primary objective of this study is to develop a hybrid deep neural network architecture that integrates a Deep Feedforward Neural Network (DFNN) with a Long Short-Term Memory (LSTM) network to generate accurate, context-aware, and personalized recommendations. This hybrid design captures both the static characteristics and the dynamic behavioral patterns of users, offering a comprehensive understanding of user preferences. Specifically, the DFNN is employed to abstract high-level representations from static user features, while the LSTM component models the sequential and temporal dependencies inherent in user browsing behavior. The architecture effectively leverages both static and time-dependent attributes, enabling the system to adapt to user interactions in real-time. Through this design, the model enhances the accuracy, adaptability, and relevance of the recommendation process by learning intricate patterns formed through various stages of user engagement across the e-commerce platform.

2. Literature Review

The rapid evolution of recommendation systems has led to the development of diverse approaches incorporating deep learning, content-based filtering, context-awareness, and social network analysis. This literature survey reviews key contributions that have significantly advanced the field, focusing on hybrid neural architectures, sequential modeling, attention mechanisms, and regularization techniques. Additionally, the survey explores community detection in social recommenders and the integration of contextual information to enhance personalization, accuracy, and user engagement across various e-commerce and digital platforms.

Dropout, a regularization method designed to lessen overfitting in neural networks, was first presented by Srivastava et al. [6]. Randomly "dropping out" units (together with their connections) during training are the main notion, which keeps units from co-adapting too strongly. As a result, the network is forced to acquire more resilient and broadly applicable features. Dropout averages the predictions made during inference and efficiently mimics the training of an ensemble of many networks. Numerous empirical tests on a range of tasks, such as speech processing and picture recognition, showed notable gains in model performance. Since then, the method has been widely used to train deep neural networks.

Javed et al. [7] provide an extensive review of content-based and context-based recommendation systems, highlighting their underlying principles, methodologies, and real-world applications. The paper distinguishes between traditional content-based filtering, which relies on user-item profiles, and context-aware systems, which incorporate additional situational factors such as time, location, and user mood. It further examines the strengths and limitations of each approach, emphasizing the growing importance of contextual information in enhancing personalization and relevance. The study also outlines recent advancements and challenges, offering valuable insights into future research directions for building more adaptive and intelligent recommender systems.

Gasparetti et al. [8] present a comprehensive survey on community detection techniques within the context of social recommender systems. The paper explores how identifying user communities can enhance recommendation performance by leveraging social relationships and behavioral similarities. It categorizes community detection approaches into topology-based, interaction-based, and hybrid models, analyzing their applicability and impact on recommendation accuracy and scalability. The study also discusses key challenges such as data sparsity, scalability, and dynamic user behavior. Overall, the survey highlights the pivotal role of community structures in improving personalization, trust, and diversity in social recommendation systems.

Saini and Singh [9] propose a content-based recommender system that leverages a stacked LSTM network combined with an attention-based autoencoder to enhance recommendation quality. The model captures temporal patterns in user behavior using stacked LSTM layers, while the attention-based autoencoder focuses on identifying the most relevant features from item content data. This hybrid approach enables the system to better understand user preferences and provide more accurate, personalized suggestions. Experimental results demonstrate improved performance compared to traditional content-based methods, highlighting the model's effectiveness in capturing both sequential and contextual relationships within user-item interactions.

DLERSE, a deep learning-enhanced recommendation system intended to promote e-commerce user interactions, is introduced in this work [10]. In order to improve recommendation accuracy, it investigates cutting-edge architectures such as CNNs, RNNs, and transformers. The study uses hybrid models and data augmentation to overcome issues including cold starts and data sparsity. In order to gain user trust, it integrates explain ability techniques and stresses model scalability through edge computing and compression. Additionally, DLERSE incorporates privacy-preserving methods such as federated learning and differential privacy. The system offers a thorough framework for next-generation e-commerce customization and is built to integrate seamlessly into current platforms, as confirmed by real-world case studies.

Sethi et al. [11] suggests using the LCNA-LSTM-CNN attention-based model to improve e-commerce platforms' recommendation algorithms. The model successfully captures users' sequential behavior and local feature patterns by combining Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN) with an attention mechanism. By digging deeper into customer preferences, this hybrid design seeks to offer more precise and tailored product recommendations. The study shows how recommendation accuracy has increased and how the model might help e-commerce businesses implement more successful marketing campaigns. The outcomes demonstrate significant performance improvements over conventional models.

Shi J, 2022 [12], proposed the BERT-BiLSTM model as a cutting-edge framework for recommender system development. BERT's pre-training task is its key component; it makes the model bidirectional and helps it identify more complex contextual relationships in the data. Alternatively, the BiLSTM (Bidirectional Long Short-Term Memory) network improves the model by processing data both forward and backward, enabling it to capture context-based information efficiently. By merging these two models, the BERT-BiLSTM framework enhances text data comprehension and produces recommendations that are more precise. A lower RMSE value of 0.82 is obtained by the suggested model, outperforming other models like BERT-SVM, BERT-RNN, and BERT-LSTM.

The paper BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer by Sun et al. [13] introduces BERT4Rec, a model that leverages BERT for sequential recommendation. Unlike traditional models that process sequences in a unidirectional manner, BERT4Rec utilizes a bidirectional approach to capture both past and future context of user interactions. The model is designed to better understand the sequential dependencies in user behavior, resulting in improved recommendations. Experimental results show that BERT4Rec outperforms state-of-the-art sequential recommendation methods in terms of accuracy and efficiency.

3. Methodology

The methodology involves integrating a Deep Feedforward Neural Network (DFNN) with a Long Short-Term Memory (LSTM) network to create a hybrid recommendation model. DFNN processes static user-item features, while LSTM captures sequential user behaviors. The combined model is trained and evaluated using an e-commerce clickstream dataset for personalized recommendations. For the proposed recommendation system, the methodology followed is shown in Fig. 1.

The Hybrid DFNN+LSTM model excels in personalization, context-awareness, and the ability to handle sequential data. Its ability to generalize better than standalone DFNN or LSTM models, along with high precision, recall, and scalability, makes it highly suitable for real-world recommendation systems—especially in e-commerce platforms where user behavior is dynamic and complex. A comparison table of the advantages of Hybrid model over individual models is illustrated in Table 1.

Table 1. *Summary of Key Advantages of Hybrid model*

Parameter	**DFNN**	**LSTM**	Hybrid DFNN+LSTM
Personalization	Limited to static features	Limited to sequential behavior	High personalization using both static and sequential data
Context Awareness	None	Limited to sequential data	High context-awareness with sequential and static data
Sequential Data Handling	No	Excellent	Excellent, combines both static & sequential
Generalization	Good for known patterns	Good for sequential patterns	Excellent, combines strengths of both DFNN and LSTM
Flexibility and Adaptability	Moderate	Limited to time-based behavior	High, adaptable to different tasks
Precision & Recall	Moderate	Moderate	High, improves overall recommendation quality
Scalability	Moderate	Low (sequential nature)	High, scalable for large datasets
Feature Integration	Static features only	Sequential features only	Flexible, integrates both static and dynamic features

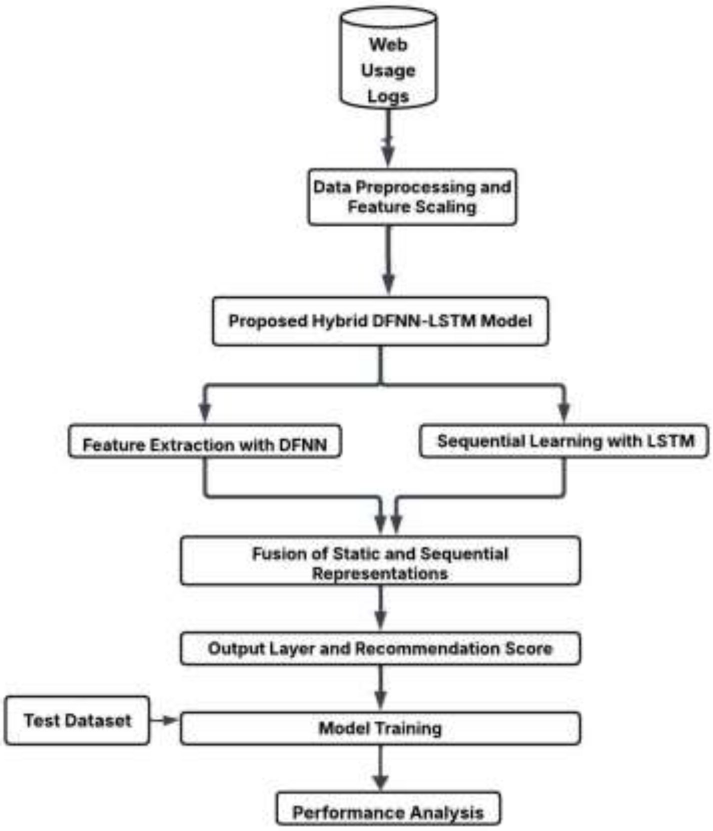

Figure 1. *Proposed Methodology for Recommendation system*

3.1 Dataset Description

This research makes use of a dataset made up of clickstream logs gathered from an online store where customers engage with a range of goods. Every dataset entry records the user's browsing habits and offers crucial information about their interests and preferences. Finding important elements that affect user engagement and determining the level of popularity of various products are made possible by examining these trends. In order to predict user interests and provide useful information for developing an efficient recommender system, these browsing patterns are crucial. Table 2 provides a thorough explanation of the dataset.

Table 2. *Features and their description*

Features	Description
User_id	This acts as a unique identity for the user and it helps in traversing the browser behavior of individual users.
Product_id	This acts as unique identification for every individual product
Event_type	This relates to the nature of interaction done by the user that could be viewing of cart, padding product on cart as well as purchasing of cart. This is very helpful when modeling of user is done.
Catogery_id	This acts as an identification element for categorizing the product and this is useful when grouping products on the basis of logical selection.
Catogery_code	This represents the hierarchical category where they have code which may represent mobile, books, etc. This type of hierarchical category is useful in taxonomy-based model creation.
price	Price of the product at the time of interaction. Key for revenue and price sensitivity analysis. This relates to the amount that is mentioned when user is interacting over web. This is very useful when the vital elements of revenue or the sensitive process dynamics need to be calculated.
Timestamp	This is usually in milliseconds when considering Unix based timestamp. This is very useful when tracking temporal patterns as well as understanding the session of the user browser.

3.2 Data Preprocessing and Feature Scaling

Typically, these kinds of e-commerce logs include categorical features, such as the user's ID, the product's ID, and user-performed operations. All of these classifications are textual in nature. The LabelEncoder is used to encode these textual features because the DL is unable to read text information directly. Additionally, the encoder needed for each column is maintained and stored using the directory label_encoder. This directory comes in very handy when decoding is required.
Utilizing this transformation technique also has the advantage of recognizing the user's unique characteristics and comprehending how their product traverses in the form of discrete values. When taking into account all of the input features, it is crucial that they be treated equally. To this end, MinMaxScaler is used, which aids in data normalization by transforming all of the input values into a specific range of [0, 1].

3.3 Proposed Hybrid DFNN-LSTM Model

3.3.1 Input Feature

Feature	Type	Description
User_id	Categorical	Static user identity
Product_id	Categorical	Dynamic, sequential product interaction
Event_type	Categorical	Dynamic behavior (click, view, cart, etc.)
Category_id	Categorical	Static item hierarchy
Category_code	Categorical	Semantic content category
Price	Numeric	Item price
Timestamp	Temporal	Sequential interaction timing

3.3.2 DFNN Component — Static Context Encoder

Feature Selection:

- Categorical: User_id, Category_id, Category_code
- Numerical: Aggregated Price, Session duration (if derived)

Encoding:

- Label Encoding + Embedding → Project categorical data into dense continuous space
- Min-Max / Z-score Normalization for numerical values

Feedforward Network:
Let $x\_static \in R^n$ be the static input vector after preprocessing and embedding.

$$\mathbf{h_1} = \mathbf{ReLU}(\mathbf{W_1} \cdot \mathbf{x}_{static} + \mathbf{b_1})$$
$$\boldsymbol{h_2} = \boldsymbol{ReLU}(\boldsymbol{W_2} \cdot \boldsymbol{h_1} + \boldsymbol{b_2})$$
$$\boldsymbol{z_{DFNN}} = \boldsymbol{ReLU}(\boldsymbol{W_3} \cdot \boldsymbol{h_2} + \boldsymbol{b_3})$$

Where:

- $\boldsymbol{W_1}, \boldsymbol{W_2}, \boldsymbol{W_3}$: Learnable weight matrix of the Feedforward Layers
- $\boldsymbol{b_1}, \boldsymbol{b_2}, \boldsymbol{b_3}$: corresponding bias vectors
- $\boldsymbol{h_1}, \boldsymbol{h_2}$: Hidden representation capturing abstract feature interaction
- $\mathbf{z_{DFNN}}$: Dense Static user-item preference representation

3.3.3 LSTM Component — Sequential Behavior Encoder

Feature Encoding per timestep t:

$$\mathbf{x_t} = \mathbf{Concat}\begin{pmatrix} \mathbf{Embed}(\boldsymbol{Product_{id_t}}), \mathbf{Embed}\left(\mathbf{Event_{type_t}}\right), \\ \mathbf{Scaled(Price_t)}, \mathbf{TimeEncoding(Timestamp_t)} \end{pmatrix}$$

Sequence Modeling:

$$(\mathbf{h_t}, \mathbf{c_t}) = \mathbf{LSTM}(\mathbf{x_t}, \mathbf{h_{t-1}}, \mathbf{c_{t-1}})$$
$$\mathbf{z_{LSTM}} = \mathbf{h_T}$$

Where:

- x_t : encoded input at Timestep t
- h_t, c_t : Hidden and cell states of the LSTM
- z_{LSTM}: Final hidden state, capturing dynamic user intent

3.3.4 Hybrid Fusion Layer

Combine both semantic + behavioral embeddings:

$$\mathbf{z_{hybrid}} = \mathbf{Concat}(\mathbf{z_{DFNN}}, \mathbf{z_{LSTM}})$$

Pass through fusion MLP:

$$\mathbf{h_{fusion}} = \mathbf{ReLU}(\mathbf{W_f} \cdot \mathbf{z_{hybrid}} + \mathbf{b_f})$$

Final Output Layer:

$$\hat{\mathbf{y}} = \boldsymbol{\sigma}(\mathbf{W_o} \cdot \mathbf{h_{fusion}} + \mathbf{b_o})$$

Where:

- z_{DFNN} : Dense Static user-item preference representation
- z_{LSTM} : Dynamic user-item preference representation
- ŷ: Recommendation score
- σ : (binary sigmoid or multi-class Softmax)

3.3.5 Output Layer

- Use sigmoid for binary classification (e.g., click vs. no click)
- Use Softmax for next-item prediction or ranking over all products

3.3.6 Loss Function

- Binary Cross-Entropy Loss for click prediction
- Categorical Cross-Entropy for next-product prediction

3.4 Model Training

In order to understand the efficiency of the proposed work, the experiment were done using three different models which are (1) Deep Feedforward Neural Network (DFNN), (2) Long Short-Term Memory (LSTM), and (3) a Hybrid DFNN+LSTM model. The models have been trained independently considering the same dataset.

3.5 Model Evaluation

The proportion of true positive predictions among all positive predictions is indicated by precision, whereas accuracy gauges a model's overall correctness. Recall evaluates how well the model can recognize every real positive instance. The F1 score is particularly helpful in datasets that are unbalanced because it balances precision and recall into a single metric. The purpose of each metric varies based on the situation and the expenses related to false positives and false negatives. The four aforementioned measures are as follows:

$$\text{Precision} = \frac{|\text{ Recommended Items} \cap \text{Relevant Items }|}{|\text{ Recommended Items }|}$$

Where,TP (True Positives) is the recommended items that are relevant and FP (False Positives) denotes the recommended items that are not relevant.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{|\text{ Recommended Items} \cap \text{Relevant Items }|}{|\text{ Relevant Items }|}$$

Where, TP (True Positives) represents recommended and relevant and FN (False Negatives) are the relevant items that were not recommended.

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where TN (true negatives) means items that have not been recommended and are not relevant.

$$\text{F-1 Score} = 2 \text{ x} \frac{\text{Precision x Recall}}{\text{Precision} + \text{ Recall}}$$

Only when Recall and Precision are both high does the F1 score rise. Better than Accuracy, the F1 score is the harmonic mean of precision and recall.

4. Results and Discussion

The dataset is trained and tested using three models: DFNN, LSTM and hybrid architecture of DFNN and LSTM. The performance metrics of these three models are observed. The following

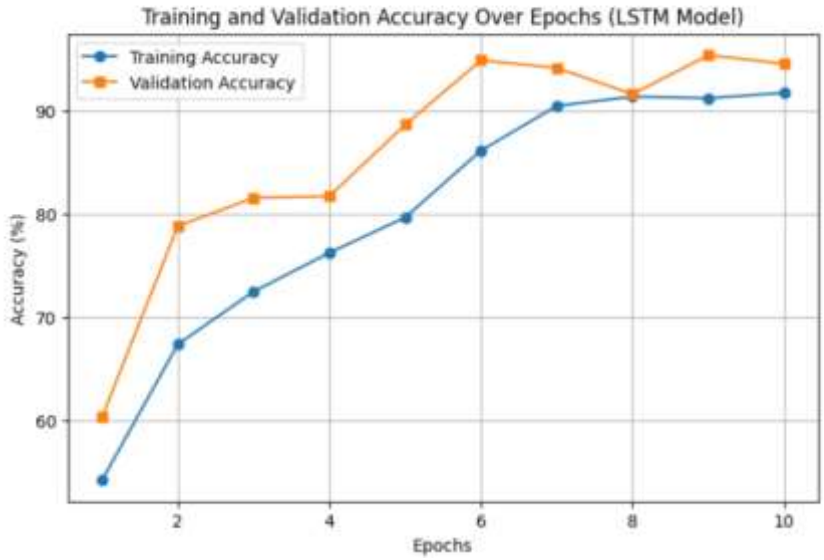

Figure 2. *Training and Validation Accuracy of LSTM*

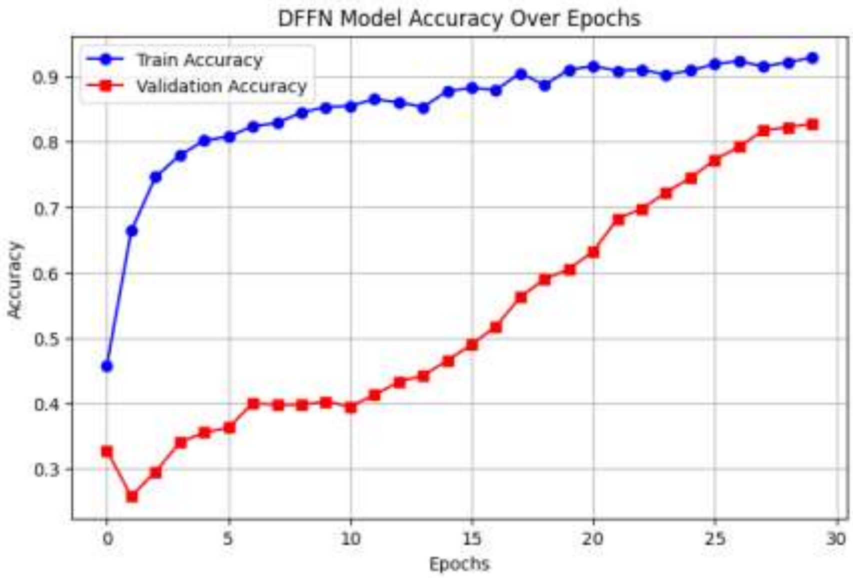

Figure 3. *Training and Validation Accuracy of DFNN*

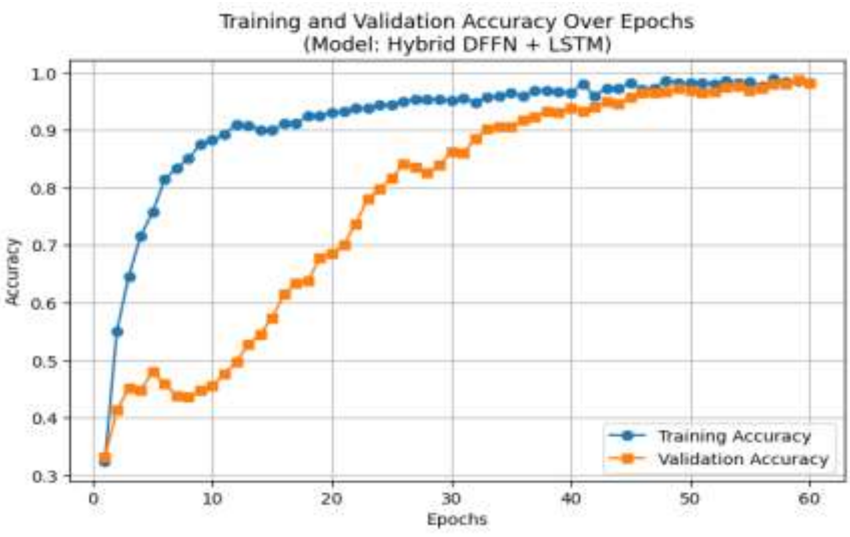

Figure 4. *Training and Validation Accuracy of Proposed Hybrid model (DFNN+LSTM)*

Table 3 shows comparative analysis of overall Accuracy of these models during the testing and training phase.

Table 3. *Comparative analysis based on accuracy*

Models	Training Accuracy	Validation Accuracy
DFNN	91.23%	86.50%
LSTM	93.85%	94.53%
Proposed Model	99.06%	98.98%

Fig. 2, Fig. 3 and Fig. 4 denotes the training and validation graphs of all these three models. The evaluation results show that the Proposed Model significantly outperforms both the DFNN and LSTM models in terms of training and validation accuracy, achieving 99.06% and 98.98% respectively. While the DFNN (91.23% train, 86.50% validation) lacks temporal awareness, the LSTM (93.85% train, 94.53% validation) better captures sequential user behavior with improved generalization. The Proposed Model integrates advanced deep learning techniques such as temporal modeling, attention mechanisms, and feature embeddings, enabling it to deliver highly personalized and context-aware recommendations with minimal overfitting and superior generalization, making it the most robust and effective recommender among the three.

Furthermore, other performance metrics like F1, Recall and Precision is calculated for the parameters “add_to_cart” , “purchase”, “search”, “view”. Table 4, Table 5 and Table 6 represent the various observations on these metrics for each individual model.

Table 4. *Precision, Recall and F1-score of DFNN*

Action_Class	Precision	Recall	F1-Score
add_to_cart	0.94	0.91	0.93
purchase	0.98	0.97	0.97
search	0.96	0.95	0.95
view	0.99	0.98	0.98

Table 5. *Precision, Recall and F1-score of LSTM*

Action Class	Precision	Recall	F1-Score
add_to_cart	1.00	0.91	0.95
purchase	0.92	0.94	0.93
search	0.73	1.00	0.85
view	0.90	0.62	0.72

Table 6. *Precision, Recall and F1-score of Proposed model*

Action Class	Precision	Recall	F1-Score
add_to_cart	1.00	0.99	0.99
purchase	0.97	0.99	0.98
search	1.00	1.00	1.00
view	0.99	0.98	0.99

The Hybrid Model outperforms both DFNN and LSTM in all evaluation metrics — Precision, Recall, and F1-Score — across all classes (add_to_cart, purchase, search, view). Its high consistency and near-perfect scores make it the most accurate, robust, and reliable model for building a user behavior-based recommender system in this scenario as shown in Table 7.

Table 7. *Comparative analysis of the three models*

Metric	DFNN	LSTM	Hybrid Model
Precision	Moderate (~0.70–0.90)	High (~0.75–1.00)	Very High (~0.97–1.00)
Recall	Moderate (~0.60–0.92)	High (~0.85–1.00)	Very High (~0.98–1.00)
F1-Score	Moderate (~0.65–0.93)	High (~0.84–0.98)	Very High (~0.98–1.00)

5. Conclusion and Future Work

This study presents a hybrid deep learning framework that integrates DFNN with LSTM networks to deliver personalized and context-aware recommendations for e-commerce platforms. The proposed model effectively captures both static and dynamic (e.g., clickstream or time-series) user behavior to provide highly accurate and timely suggestions. Experimental results using e-commerce web log data confirm the model’s superiority over standalone DFNN and LSTM architectures, achieving a training accuracy of 99.06% and validation accuracy of 98.98%. Furthermore, the hybrid model consistently outperforms in terms of Precision, Recall, and F1-score across critical action classes such as "add_to_cart", "purchase", "search", and "view", with scores nearing or reaching 1.00. These findings validate the model's robustness, generalization capability, and real-world applicability in understanding and predicting user’s intent.

However, despite the strong performance, the study acknowledges certain limitations. Firstly, the model requires a large volume of well-labeled sequential data, which may not be available in all e-commerce setups. Additionally, the cold-start problem for new users or items remains partially unresolved, and model performance may degrade if the user behavior changes significantly over time. For future work, the integration of attention mechanisms or transformer-based architectures can further enhance context modeling. Incorporating reinforcement learning could help adapt recommendations in real-time based on user feedback. Moreover, expanding the model to support web content data (e.g., text, image, and video) along with web log data in recommendations will be key directions for evolving intelligent recommender systems.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Singh, P. K., Pramanik, P. K. D., Dey, A. K., & Choudhury, P. (2021). Recommender systems: An overview, research trends, and future directions. *International Journal of Business and Systems Research, 15*(1), 14–52.
[2] Ko, H., Lee, S., Park, Y., & Choi, A. (2022). A survey of recommendation systems: Recommendation models, techniques, and application fields. *Electronics, 11*(1), 141.
[3] Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E., & Fragopoulou, P. (2022). Collaborative filtering recommender systems taxonomy. *Knowledge and Information Systems, 64*(1), 35–74.
[4] Jozani, M., Liu, C. Z., & Choo, K. K. R. (2023). An empirical study of content-based recommendation systems in mobile app markets. *Decision Support Systems, 169*, 113954.
[5] Lin, D., & Jingtao, S. (2015, October). A recommender system based on contextual information of click and purchase data to items for e-commerce. In *Third International Conference on Cyberspace Technology (CCT 2015)* (pp. 1–6). IET.
[6] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research, 15*(1), 1929–1958.
[7] Javed, U., Shaukat, K., Hameed, I. A., Iqbal, F., Alam, T. M., & Luo, S. (2021). A review of content-based and context-based recommendation systems. *International Journal of Emerging Technologies in Learning (iJET), 16*(3), 274–306.
[8] Gasparetti, F., Sansonetti, G., & Micarelli, A. (2021). Community detection in social recommender systems: A survey. *Applied Intelligence, 51*(6), 3975–3995.
[9] Saini, K., & Singh, A. (2024). A content-based recommender system using stacked LSTM and an attention-based autoencoder. *Measurement: Sensors, 31*, 100975.
[10] Bhavani, B., & Haritha, D. (2025). Dlerse: Deep learning-enhanced recommendation systems for e-commerce user interaction. *Journal of Theoretical and Applied Information Technology, 103*(4).
[11] Sethi, V., Kumar, R., Mehla, S., Gandhi, A. B., Nagpal, S., & Rana, S. (2024). LCNA-LSTM CNN-based attention model for recommendation system to improve marketing strategies on e-commerce. *Journal of Autonomous Intelligence, 7*(1).
[12] Shi, J. (2022, December). E-commerce products personalized recommendation based on deep learning. In *2022 6th Asian Conference on Artificial Intelligence Technology (ACAIT)* (pp. 1–5). IEEE.
[13] Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019, November). BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 1441–1450).