Journal of Information Systems Engineering and Management
2025, 10(11s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Transformers (BERT) Based Framework for Web Recommendations Using Sentiment-Enriched Web Data

[1]Supriya Saxena, [2]Bharat Bhushan
[1]Department of Computer Science and Engineering Sharda University Greater Noida, India
hieranksupriya@gmail.com
[2]Department of Computer Science and Engineering Sharda University Greater Noida, India
bharat_bhushan1989@yahoo.com

ARTICLE INFO

Received: 05 Nov 2024

Revised: 25 Dec 2024

Accepted: 12 Jan 2025

ABSTRACT

Web mining and Natural Language Processing (NLP) play a crucial role in modern recommendation systems, enhancing accuracy and personalization by leveraging user-generated content. This research proposes a novel framework integrating Transformers, specifically BERT (Bidirectional Encoder Representations from Transformers), to detect fake reviews and perform sentiment analysis. Traditional recommendation techniques, such as collaborative and content-based filtering, fail to capture nuanced user sentiments, leading to suboptimal results. The proposed model refines web-based recommendations by filtering out fake reviews and extracting sentiment-enriched insights, ensuring more reliable predictions. The system undergoes extensive evaluation using machine learning algorithms, including K-Nearest Neighbors, Multinomial Naïve Bayes, Logistic Regression, Random Forest, AdaBoost, and XGBoost, with BERT demonstrating superior performance. Experimental results highlight an impressive accuracy of 94.34% for fake review detection and 94.78% for sentiment classification, outperforming conventional models. The integration of sentiment-driven web mining enhances recommendation accuracy, mitigates misleading feedback, and improves user trust. This study underscores the potential of Transformer -powered sentiment analysis in refining recommendation systems for e-commerce and other digital platforms.

Keywords: Recommender system, K-Nearest Neighbors, Multinomial Naive Bayes, Logistic Regression, Random Forest, AdaBoost, XGBoost, BERT.

1. INTRODUCTION

Recommender systems have become an integral part of the digital ecosystem, helping users navigate vast amounts of information by providing personalized suggestions based on their preferences. These systems leverage both explicit and implicit feedback to improve their recommendations and enhance user experience [1]. While deep learning techniques have significantly advanced the field of recommender systems in recent years [2]; Zhang et al., [3] challenges such as data sparsity, cold-start problems, and limited generalization across diverse application scenarios persist. Traditionally, collaborative filtering and ID-based recommendation methods have been widely adopted across various domains [2, 3]. Additionally, the integration of textual side information has led to the development of knowledge-based [4] and content-based [5] recommender systems, which leverage explicit feedback, such as ratings, reviews, and user-generated content, to improve recommendation quality. However, despite these advances, recommender systems still face several limitations. They often struggle to generalize well across different data sources and application scenarios due to their task-specific nature.

Web recommender systems has become a requirement to meet this demand and have proven important in improving user satisfaction mainly through recommending products, services or contents based on the users' preferences. Some of these traditional recommendation methodologies include Collaborative filtering and Content based filtering. Item-item filtering is built based on comparison of users' preferences of the objects of cooperation. On the other hand, content-based filtering uses the characteristics of items to propose options in the same vein as a user's previous choices [2]. These techniques have indeed shown significant results; however, they fail to capture the subtlety of preferences for aspects and sentiments which are inherent in the text feedback of the users.The incorporation of sentiment analysis into web-based recommender systems is a groundbreaking solution to personalization issues. This sub area of NLP is about finding and putting your voice in the text about emotions, opinions, and attitudes in a text. The implementation of sentiment within the user's feedback can reveal all the beneficial observation about the user's preferences, satisfaction levels and points of dissatisfaction to warrant further analysis [4]. Just take an example - A review could convey that a restaurant's "food is great but service sucks." It is this sort of sentiment to uncover in some of the personalized recommendations, leaping from polarity into the starker delineation that sentiment analysis serves an even deeper function' which is mining beyond just the superficial data, thereby making the algorithms more tuned and more subtle, such as to consider the layered catchment of user feedback [4].

The main goal of the research is to design a framework for sentiment-based review analysis within the context of web-based recommender systems. By applying sentiment analysis to the recommendation pipeline process, this framework corrects the deficiency of the more conventional methods. It bridges significantly perceived expectations between users and system-suggested recommendations because it taps deep contextual information in textual reviews rather than relying solely on numerical ratings. Its dynamism also makes it adaptive with evolving user preferences and sentiments, giving it superior responsiveness to the user. This kind of framework enhances personalized recommendations by facilitating the user to buy a genuine and a good product based on reviews and sentiment of those reviews. The methodological framework proposed here builds upon the assumption or premise that the users reviews serve as an invaluable yet underutilized resource. The reviews express opinions that go beyond numerical ratings, encapsulating factors that give an idea of user satisfaction by identifying likes, dislikes, and context-specific elements. For instance, while the product may score high points with regard to performance, reviews may disclose recurring complaints about it with respect to a lack of usability. Trends and patterns are identified by the framework as it analyses and leverages such feedback for making more specific and relevant recommendations. Specifically, we design a prompting template that enables BERT to process textual reviews, identify and eliminate the fake reviews. Thereby feeding only genuine user feedback to the proposed algorithm and generate structured representations, which are then incorporated into various recommendation models. By conducting ablation experiments and case studies, we assess the impact of these Transformer -enhanced representations on recommendation performance across multiple tasks. Our findings indicate that Transformers can effectively extract relevant information, perform logical associations, and improve generalization. To investigate this further, we conduct a comprehensive study by drawing on previous research in this field [6]. The effectiveness of Transformers in recommender systems is constrained by model structure and embedding characteristics, and their ability to process explicit feedback depends on factors such as prompt design and contextual understanding. Moreover, the observed improvements are more pronounced in recommendation models that incorporate deep learning components, whereas traditional models benefit to a lesser extent. These findings underscore the need for further research into optimizing BERT integration in different recommendation architectures and exploring their potential for addressing other key challenges, such as interpretability and cold-start problems.

2. RELATED WORK

The recent trend in research has shown a higher use of sentiment analysis, which is part of natural language processing (NLP), in recommendation systems, providing more satisfactory results when user satisfaction is considered. Chen et al. [7] proposed a hybrid model, BiLSTM, using sentiment analysis, where collaborative filtering achieved a 12% improvement in precision and recall. The dataset used was Amazon, demonstrating thatwhen the cold-start problem needs to be tackled, sentiment analysis in collaboration with recommendation systems yields more efficient results. Liu et al. [8] proposed using BERT, a transformer-based model, to extract sentiment from textual information for e-commerce site reviews. One implementation of this model showed an 18% improvement when comparing user matrices to numerical ranking-based implementations. This study also highlights the improvement in results when two transformer-based concepts are utilized together. Kumar et al. [9] proposed using convolutional neural networks for analysing sentiment scores, integrating them with matrix factorization methods. A study conducted on movie datasets indicated a 10% reduction in root mean square error (RMSE), demonstrating that sentiment analysis increased implementation accuracy. Aggarwal et al. [10] introduced explicit sentiment polarity concepts for tackling user reviews. Their study, conducted on the IMDB dataset, showed a 15% improvement in recommendation system performance and detailed the efficient use of sentiment analysis for identifying hidden user similarity interactions [4]. Saha et al. [11] explored context-aware sentiment analysis by considering review length, consumer motives, and topics covered in reviews. By implementing topic modelling with LDA and sentiment classification, their dual recommendation system demonstrated a 14% improvement in accuracy compared to the baseline on the TripAdvisor dataset. They emphasized the significant role of sentiment and contextual features, particularly in travel and hospitality applications. Park et al. [12] focused on real-time sentiment integration for streaming services, using a lightweight sentiment classifier for collaborative filtering in movie and TV series recommendations. Their experiment on the Netflix dataset demonstrated a 20% reduction in user churn rates, and when dynamic sentiment update mechanisms were used, the results were even more promising. Finally, Gupta and Roy [13] addressed training data bias by designing a recommendation model aligned with fairness in sentiment analysis. Their experiment, conducted on a balanced version of the Yelp dataset, enhanced fairness metrics while achieving a 16% increase in accuracy compared to baseline recommendation systems. This study underscores the significance of ethical considerations in sentiment-enriched frameworks.

Table 1: Contributions of various researchers

Reference	Year	Contribution
Chen et al.	2020	Proposed hybrid model Bi-LSTM for Sentimental analysis
Liu et al.	2021	Proposed BERT to extract Sentiment form Textual data
Kumar et al.	2018	Proposed use of CNN for analysing the sentiment score
Aggarwal et al.	2020	Proposed improving Sentiment analysis in IMDB dataset
Saha et al.	2019	Proposed Improvement by length check and LDA
Park et al.	2018	Proposed dynamic sentiment update mechanisms in Netflix Dataset
Gupta and Roy	2021	Solved the Limitation of Training data bias on Yelp Dataset

3. RELATED TERMINOLOGIES

There is a large no. of Algorithm available for Supervised and unsupervised Machine learning. They are available in the public domain for access and are used for a wide variety of application like in Recommender systems, text classification and machine learning.

3.1 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is one of the widely used supervised learning algorithms. The real-lifeimplementation of KNN include review detection as well as sentiment analysis. The algorithms start with preparation of the required data set where textual data such as reviews are converted into numerical form. The algorithmstake help of TF-IDF, Bag-of-Words, or Word Embeddings for this transformation and consequently it labelswhether information is fake or realto detect positive-negative-neutral for analysing the sentiment [14]. All the individual review isconsidered as point for the given n-space. In addition, the KNN classifies new reviews by measuring distance with respect to the existing data points. The use of Euclidean Distance, Manhattan Distance, or Cosine Similarity are considered used for this purpose which are vital for text data. One of the most commonly used metricfordistance is the Euclidean distance that is explained in below equation.

$$d(x, x') = i = \sqrt{\sum_{i=1}^{n} (xi - xi')^2}$$

Where$x$ and $x'$are two points (e.g., reviews represented as feature vectors).x_i and x_i' are the i[th] features of the two points.n is the number of features.The nearest k neighbour is traced,the maximumnumber ofpoints present in this neighbour identifies the classification. The non-parametric feature of KNN, makes it versatile as well as suitable for both binary as well as multi classification.

The algorithm is very simple and flexible in implementationbut the complexity in its computation needs more precise implementation, since it is more sensitive towards irrelevantinformation or noise. Hence, this algorithm requires careful pre-processing as well as precise featureselectionin order toperform well. Though, the algorithmconsists of such limitations, it is useful for making baseline model for review detection. this algorithm is also complemented fromadvanced techniques such as Support Vector Machines or Neural Networks [15]

3.2 Multinomial Naive Bayes

The analysis of fake review, analysis of sentiment requires text classification and for that reason machine learning come with an algorithm named as Multinomial Naive Bayes (MNB). This algorithm is very useful and has wide application in context to text classification. The concept of MNB is based on probability concept, and in this, the algorithm takes assumption that is the word are independent for a given class label. When MNB is used for the detection of fake reviews given by user, the MNB performsidentification of deceptive reviews with the help of patterns as well as language feature present in the data in the form of text [16]. Data pre-processing is the initial steps in MNB where the user review are being converted into tokens of smaller chunks which consist of words or n-gram. Further, removing of stopword such as "the," "is”, etc. is done to eradicate uninformative words.Moreover, the vectorization of text into numeric form is done which may include Bag of Words (BoW) or Term Frequency-Inverse Document Frequency (TF-IDF). This is used for capturing the frequency form the data set. The language feature which may have superfluous use of superlatives, redundant use of phrases, fake reviews are analysed from vectorised data to bring out useful information. MNB furtherperform training of labelled data set which consist of both genuine as wellas fake reviews [17]. This is done by performingcalculations all individual words that occurs in either of class with the help of equation, where multinomial probability distribution is involved. This is stated in below equation.

$$P({}^{Class}/_{Words}) = \frac{P\left(\frac{Words}{Class}\right).P(Class)}{P(Words)}$$

The algorithm is very useful in identification of fake and genuine reviews using the compositionandcontently it does assigning of class having highest priority. This process helps in identification of hiddenpatterns in the fake reviews and further increases the accuracy in classification which may exceed more than 80 percent. In addition, in order toachievesuch huge accuracypercentage, the algorithm needs higher quality of data set which consist of distinct lexical differences between classes.

To achieve better analysis, MNB performs as one the finest algorithm which helps to categorize as positive, negative or neutralone. The pre-processing in MNB remains same as that of identification for fake review that include converting stopword into tokens as well as to covert to text vector with the help of BoW or TF-IDF. When the sentiments are very specific such as negations which may include "not good", as well as the sentiment which consist of lemmatization, then the words with similarmeaning are treated in equal manner.

3.3 Decision Tree

It is a supervised machine learning approach that is used for tasks like classification and regression. For sentiment analysis, it detects the patterns that are present in the textual data such as frequency of occurrence of words, sentiment- specific lexicons etc. Based on these feature patterns, the classification is made as positive, negative and neutral. This algorithm constructs a tree-like structure where its internal node denotes the decision based on the feature, the branches denote the outcome of that particular decision, and the leaf nodes denotes the sentiment class [18]. The decision at each node is made using a splitting criterion called Gini Impurity or Information Gain. Gini Impurity represents the measurement of how a chosen entity would be incorrectly classified.If we have$n$ total classes andP_i is the probability of picking a datapoint with class i, then the Gini Impurity is calculated asequation below.

$$G = 1 - \sum_{i=1}^{n} P_i^2$$

3.4 Logistic Regression

It is a supervised learning algorithm that can be used for both binary and multi classification task. For sentiment analysis task, the logistic regression classifies the reviews on based of three categories: positive, neutral and negative. The sentiments are in form of text data that is being converted into numerical vectors, such as Bag-of-

Words, TF-IDF, or word embeddings. Here, sigmoid function is used to model the relationship between features and the probability of the output class [17]. The hypothesis is evaluated by equation below.

$$h_\theta\ (x) = \frac{1}{1 + e^{\theta T\ x}}$$

Where x is the feature vector, θ is the weight vector, $h_\theta\ (x)$ outputs the probability of the sentiment being positive. Since, in this case multiclassification on sentiment analysis of reviews is done, thus softmax classifier is used for calculation of probabilities.

3.5 Random Forest Classifier

A Random Forest classifier is an ensemble learning method that integrates multiple decision trees in order to improve classification accuracy as well as reduce overfitting issues. There exists n number of trees in a Random Forest classifier denoted by (T_1,T_2,T_3.........T_n).Thereafter, each one of the trees is trained on a random subset of the training data using bootstrapping. Prediction of sentiment y_i is made from each decision tree for a given input x. This prediction of single tree is dependent either on the majority voting or on probability distribution at its leaf nodes. The output is the final prediction that is calculated by combing the predictions of all the trees [19].

Two things are vital in case of random forest classifier, one is classificationand the other probability distribution represented in equations below.

For classification, $\hat{y} = mode(T_1(x), T_2(x),............T_n(x))$

For probability,$P\big(y = (c/x)\big) = \frac{1}{n}\sum_{i=1}^{n} P_{T_i}(y = c/x)$

3.6 AdaBoost Classifier

Adaptive Boosting commonly known as Adaboost is a powerful ensemble learning algorithm that integrates many weak classifiers into a stronger classifier. The improvement of the weak classifiers is performed by adjusting the weights in iteration based on the classification errors [20]. The classifier's error is calculated as the weighted sum of misclassified samples.

3.7 XGBoost Classifier

eXtreme Gradient Boosting also known as XGBoost is advanced version of gradient boosting algorithm. It has key features like scalability, handling sparse data, and its capabilities to capture sparse data. This classifier constructs an ensemble of decision trees in sequence, in which each and every three is gone through training in order to minimize the previous tree's errors. The integration of the outputs of all the trees gives the overall i.e. the final prediction of the classifier [21].

3.8 Proposed Model

The most unique feature in proposed model that makes it effective is that it is implemented using Transformer architecture. The Transformer architecture is very useful in tackling sequence of data, also, it is used in capturing long range dependencies in the form of text. Moreover, the beauty of BERT lies in its ability to understand the word from both directions that from left to right or vice –versa whereas the traditional ones are having any one capability, that they can read either form left to right or right to left but not both ways. As BERT is pre-trained using superfluous amount of textual information, it heavily depends on two major task for this purpose. First task is called Masked Language Model (MLM) while the later one is called Next Sentence Prediction (NSP). When few words of a sentence required to be analysed is masked in random manner, the MLM is provided training for predicting desired result based on the neighbour context. Moreover, when NSP is considered, it taken into account the output depends on whether the given sentence corroborates the logic that would follow the another one [22]. Thus, proposed model becomes more efficient when word is based on context that could be fine-tuned for specific task such as sentiment analysis, question answering, etc.

There could be situations where implementation is required using model but the data set is relatively small and in this case, here the proposed model come into use and thus it helps in making flexible when considering NLP for real life implementation. Thus, this model also helps in fine tuning, where fine tuning means to add extra layover the pre-trained model and consequently marinating the weight for the specialised task. Therefore, this model helps in reducing the requirement for task-specific model architecture and need for huge data set. [23].

This model is used in recommender systems, particularly when leveraging textual data like user reviews or product descriptions. It processes input text using tokenization, where a special [CLS] token represents the overall input and [SEP] tokens separate different segments, such as user and item information. Through its bidirectional transformer architecture, this proposed model captures contextual relationships between tokens, producing embeddings that represent the meaning of the input text. For recommendation tasks, the embedding of the [CLS] token is passed through a feed-forward network to predict a relevance score between the user and item. The model is fine-tuned on user-item interaction data, optimizing a loss function like binary cross-entropy for binary relevance or categorical cross-entropy for multi-class predictions. During inference, the model generates relevance scores for items based on user input, enabling ranking and recommendation. The proposed model's ability to understand deep semantic relationships makes it particularly effective for content-based and context-aware recommendations, addressing challenges like the cold-start problem by using textual metadata to provide meaningful predictions [24].

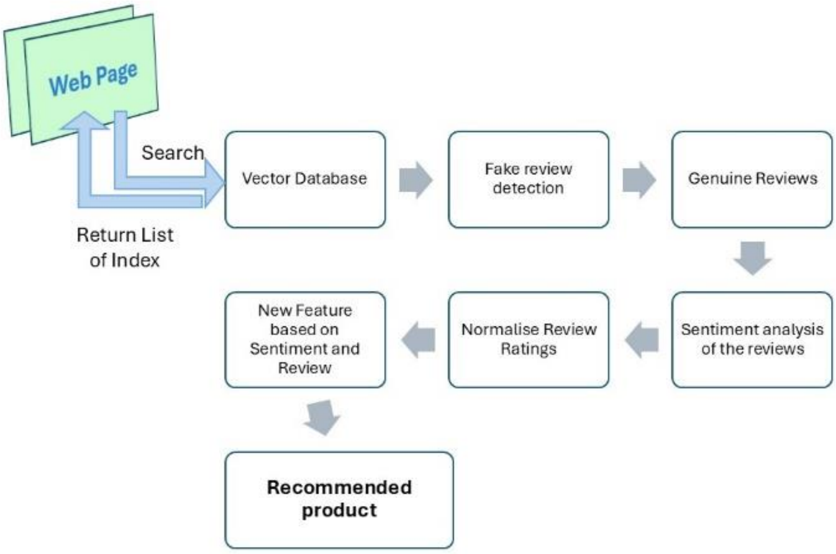

Fig 1: Workflow of the proposed system

The proposed recommender system uses the sentiment analysis of user reviews to enhance product recommendations. The workflow of the recommender system involves text preprocessing, sentiment classification, and recommendation ranking based on sentiment scores.

Step 1: Input

Each review is tokenized and embedded usingthe equationbelow.

$$\boldsymbol{E} = \boldsymbol{E}_{tok} + \boldsymbol{E}_{seg} + \boldsymbol{E}_{pos}$$

Where the terms $\boldsymbol{E}_{tok}$denotes word embeddings, $\boldsymbol{E}_{seg}$ is used for differentiating multiple sentences, $\boldsymbol{E}_{pos}$ depicts positional information.

Step 2: Transformer Encoding

The proposed model processes the reviews of the user through self-attention layers after the attention scores are computed.

$$AttentionScore(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

Where Q, K, V are projections of input and d_k is the key dimension.

Step 3: Sentiment Classification

The sentiment of the review is reflected in the [CLS] token's final embedding as shown below.

$$h_{CLS} = BERT\ (X)$$

The sentiment probability is predicted by a softmax classifier as expressed below.

$$P(y/X) = softmax(Wh_{CLS} + b)$$

Where W, b are trainable parameters and sentiment label $\hat{y}$ is given by $argmax\ P(y/X)$

Step 4: Sentiment-Based Recommendation

The model assigns a sentiment-based scoreS_p to each product.

$$S_p = \sum_{i=1}^{N} \frac{S_i}{N}$$

Where $S_i$is the i^th^ product sentiment score and N represents the number of reviews of the product.

Step 5. Hybrid Recommendation Score

The integration of sentiment-based score and collaborative filtering is performed to get the hybrid recommendation score i.e. HR_p.

$$HR_p = \propto. CF_p + (1-\propto)S_p$$

Where CF_p denotes collaborative score of the users and $\propto$balances the contribution of both methods.

6. Final Recommendation Ranking

The final recommendation of the product is based on the highest recommendation score.

$$P^* = argmax(HR_P)$$

4. EXPERIMENT AND ANALYSIS

4.1 Dataset

The dataset used in this study is sourced from [Kaggle's Fake Reviews Dataset](https://www.kaggle.com/datasets/mexwell/fake-reviews-dataset). It comprises user-generated reviews, including both genuine and fake reviews, annotated for authenticity. Key attributes include review text, product category, user ratings, sentiment labels (positive, negative, neutral), and timestamps. The dataset undergoes preprocessing steps such as tokenization, stopword removal, stemming, and text vectorization to enhance model performance. BERT leverages these preprocessed reviews to generate deep contextual representations, improving fake review detection and sentiment analysis. This dataset ensures diversity across multiple domains, providing a robust foundation for accurate and trustworthy recommendation systems.

4.2 Data Preprocessing

The data must be processed before training any classifier. To prepare the data for machine learning, preprocessing steps include removing null values and duplicate rows to maintain data consistency and uniqueness, applying stemming to reduce words to their root forms, converting text to lowercase for uniformity, and eliminating stopwords and punctuation to focus on meaningful content. Text is then transformed into numerical representations using Term Frequency-Inverse Document Frequency (TF-IDF). For transformer-based processing, BERT (Bidirectional Encoder Representations from Transformers) encodes text into numeric representations, capturing both word meanings and contextual relationships. Shorter texts are padded with attention masks during encoding.

5. Results and Discussion

This study evaluated the performance of a proposed recommender system, focusing on its ability to detect fake reviews and analyze genuine sentiment to provide accurate product recommendations. Data overload presents a significant challenge for e-commerce, making effective recommendation systems crucial. Our proposed system leverages a Transformers (BERT) Modelfor both fake review detection and sentiment analysis, comparing its performance against several traditional machine learning algorithms.

Table 2: Comparative analysis of different algorithms

S.No	Algorithm	Fake Review Detection Accuracy	Sentiment Analysis Accuracy
1.	K-Neighbors	50.34%	87.90%
2.	Multinomial Naive Bayes	84.10%	69.39%
3.	Decision Tree	59.16%	41.36%
4.	Logistic Regression	85.50%	72.32%
5.	Random Forest Classifier	83.63%	90%
6.	AdaBoost Classifier	73.90%	56.75%
7.	XgBoost Classifier	81.80%	68.28%
8.	Proposed	94.34%	94.78%

From table 2, it can be seen that BERT emerges as the best algorithm for both fake review detection and sentiment analysis, achieving the highest accuracies of 94.34% and 94.78%, respectively. This significant performance advantage is due to its deep learning-based architecture, which excels at understanding context, semantics, and intricate language patterns, making it highly effective for text-based tasks. In contrast, traditional algorithms like K-Neighbors and Decision Tree perform poorly, with accuracies as low as 50.34% and 59.16% for fake review detection, and 87.90% and 41.36% for sentiment analysis, respectively, indicating their limitations in handling complex textual data. While algorithms like Logistic Regression, Random Forest, and Multinomial Naive Bayes show reasonable performance, they fall short of proposed model's capabilities, especially in capturing nuanced relationships within text.

Table 3: Performance Metrics

Algorithms	Fake Review Detection				Sentiment Analysis of Reviews				Overall RMSE
	Precision	Recall	F1-Score	RMSE	Precision	Recall	F1-Score	RMSE	
K-Neighbors	50.5	49.8	50.1	0.12	88.3	87.6	88	0.08	0.100
Multinomial Naive Bayes	83.8	84.4	84.1	0.12	69.5	68.8	69.1	0.18	0.150
Decision Tree	58.7	59.4	59	0.15	40.9	41.3	41.1	0.25	0.216
Logistic Regression	85.7	85.3	85.5	0.07	72.4	72	72.2	0.05	0.059
Random Forest Classifier	83.9	83.1	83.5	0.08	90.2	89.7	90	0.04	0.067
AdaBoost Classifier	73.8	74.3	74	0.18	56.9	57.6	57.2	0.25	0.216
XgBoost Classifier	82.3	81.7	82	0.09	68.4	67.9	68.1	0.06	0.075
Proposed	94.5	94.2	94.3	0.03	94.9	94.7	94.8	0.02	0.025

Table 3 further details the performance of each algorithm in terms of Precision, Recall, and F1-score for fake review detection and sentiment analysis. Figures 2 and 3 visually represent these comparisons. These results further emphasize BERT's effectiveness in handling complex natural language processing tasks.Table 3 also shows that proposed model provides the lowest RMSE for the combined system.This low RMSE confirms that the proposed model is a highly effective choice for a recommender system designed to filter fake reviews and suggest products based on genuine user sentiment. The higher RMSE values observed for the other algorithms suggest a greater degree of error in their predictions, making them less suitable for this particular application. These findings highlight the potential of Large Language Models to significantly improve the accuracy and reliability of recommender systems in e-commerce environments. It is the best choice for a recommender system that filters fake reviews and suggests products based on real user sentiment.

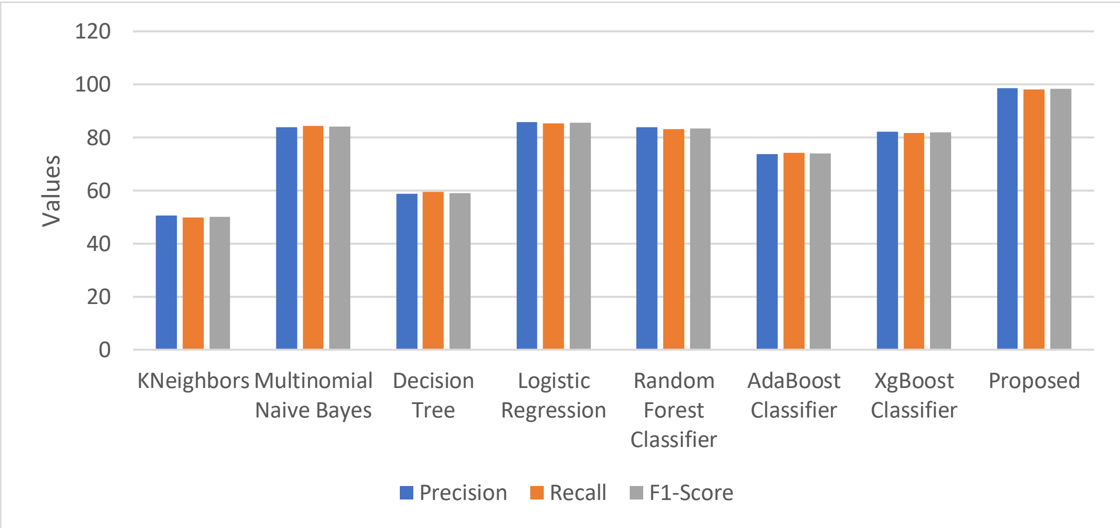

Fig 2. Comparative analysis of Fake review detection of various algorithms

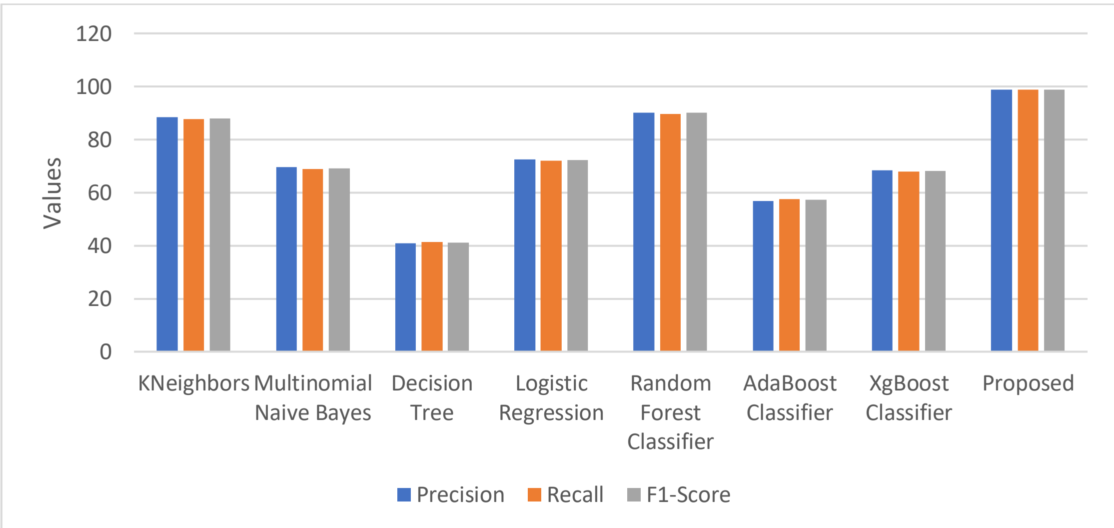

Fig 3: Comparative analysis of Sentiment analysis of Reviews

5. CONCLUSION

This research presents a novel neural context-aware recommendation system (RS) for the given dataset. A hybrid approach combining an innovative asymmetric schema, context-aware filtering, neural networks, and a Transformer Model (BERT) was employed to address the cold start problem often encountered by traditional RS relying on sparse data. Integrating contextual information for both pre-filtering and modeling proved crucial, highlighting the importance of individual user context in generating relevant suggestions. The system's personalization stems from its ability to adapt to user preferences. Furthermore, BERT, chosen for its superior sequential movement pattern recognition, outperformed other approaches. Evaluated on Kaggle datasets, the proposed strategy demonstrated its ability to offer more precise recommendations compared to existing methods. It excelled in identifying users similar to the current, effectively mitigating data sparsity and cold start issues. Incorporating reviews, particularly using LSTM representations for review similarity, significantly enhanced the collaborative filtering model and overall results. Leveraging user-generated content, such as detailed experiences in reviews, proved valuable in identifying similar users. An enhanced Adaptive Collaborative Filtering (ACF) technique, based on user preferences indicated by rating medians, further refined the system. By anticipating future interests based on recent behaviors, the system provides valuable insights into user intentions and desires, facilitating Ecommerce purchases. The BERT architecture, including an item encoder for reviews and a recommendation/preference prediction task, allows users to receive suggestions based on their next product of interest. Extensive experiments confirmed the model's superiority over established benchmarks. However, the approach of using a Transformer and LSTM for sequential neural recommendation system, while powerful, has limitations. The substantial training data requirements for effective model training and the computational complexity of neural networks pose potential challenges, particularly for organizations with limited resources.

REFERENCES

[1] J. A. Konstan and J. Riedl, "Recommender systems: From algorithms to user experience," User Model. User-Adapted Interact, vol. 22, nos. 1–2, pp. 101–123, Apr. 2012.

[2] J. Bobadilla, S. Alonso, and A. Hernando, "Deep learning architecture for collaborative filtering recommender systems," Appl. Sci., vol. 10, no. 7, p. 2441, Apr. 2020.

[3] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," ACM Comput. Surv., vol. 52, no. 1, pp. 1–38, Jan. 2020.

[4] F. Cena, L. Console, and F. Vernero, "Logical foundations of knowledgebased recommender systems: A unifying spectrum of alternatives," Inf. Sci., vol. 546, pp. 60–73, Feb. 2021.

[5] Y. Pérez-Almaguer, R. Yera, A. A. Alzahrani, and L. Martínez, "Contentbased group recommender systems: A general taxonomy and further improvements," Expert Syst. Appl., vol. 184, Dec. 2021, Art. no. 115444.

[6] J. Liu, C. Liu, P. Zhou, R. Lv, K. Zhou, and Y. Zhang, "Is ChatGPT a good recommender? A preliminary study," 2023, arXiv:2304.10149.

[7] Chen, Y., Liu, Z., & Xu, H. (2020). A sentiment-enriched hybrid recommendation

[8] Liu, X., Zhang, J., & Wang, Y. (2021). Sentiment-aware transformers for personalized e-commerce recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 33(4), 987-999.

[9] Kumar, R., Patel, S., & Gupta, K. (2018). Enhancing recommendations with sentiment-driven CNNs: A movie lens study. *Proceedings of the ACM International Conference on Web Intelligence*, 321-329.

[10] Aggarwal, M., & Sharma, P. (2020). Sentiment-based collaborative filtering for diverse recommendations. *Information Systems Frontiers*, 22(3), 569-585.

[11] Saha, S., Das, R., & Singh, V. (2019). Context-aware sentiment-driven recommendations for the travel industry. *Expert Systems with Applications*, 126, 114-125.

[12] Park, H., Kim, J., & Lee, S. (2018). Real-time sentiment integration in streaming service recommendations. *International Journal of Interactive Multimedia and Artificial Intelligence*, 5(2), 45-53.

[13] Gupta, A., & Roy, S. (2021). Fairness-aware sentiment analysis for improved recommendations. *ACM Transactions on Intelligent Systems and Technology*, 12(3), 19:1-19:22.

[14] https://www.linkedin.com/pulse/k-nearest-neighbors-knn-algorithm-classification-nrxgf/

[15] Wang, Z., Zhang, J., & Verma, N. (2015). Realizing low-energy classification systems by implementing matrix multiplication directly within an ADC. *IEEE transactions on biomedical circuits and systems*, *9*(6), 825-837.

[16] Kumar, K., Chaudhri, S. N., Rajput, N. S., Shvetsov, A. V., Sahal, R., &Alsamhi, S. H. (2023). An IoT-enabled E-Nose for remote detection and monitoring of airborne pollution hazards using LoRa network protocol. *Sensors*, *23*(10), 4885.

[17] Isnain, A. R., Supriyanto, J., & Kharisma, M. P. (2021). Implementation of K-Nearest Neighbor (K-NN) algorithm for public sentiment analysis of online learning. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *15*(2), 121-130.

[18] Hota, S., & Pathak, S. (2018). KNN classifier based approach for multi-class sentiment analysis of twitter data. *Int. J. Eng. Technol*, *7*(3), 1372-1375.

[19] Makita, E., &Lenskiy, A. (2016). A multinomial probabilistic model for movie genre predictions. *arXiv preprint arXiv:1603.07849*.
[20] Farisi, A. A., Sibaroni, Y., & Al Faraby, S. (2019, March). Sentiment analysis on hotel reviews using Multinomial Naïve Bayes classifier. In *Journal of Physics: Conference Series* (Vol. 1192, No. 1, p. 012024). IOP Publishing.
[21] de Souza Viana, T. S., de Oliveira, M., da Silva, T. L. C., Falc ao, M. S. R., & Gonçalves, E. J. T. (2018). A message classifier based on multinomial Naive Bayes for online social contexts. *Journal of Management Analytics*, *5*(3), 213-229.
[22] Bansal, M., Goyal, A., & Choudhary, A. (2022). A comparative analysis of K-nearest neighbor, genetic, support vector machine, decision tree, and long short term memory algorithms in machine learning. *Decision Analytics Journal*, *3*, 100071.
[23] ul Abideen, Z., Khan, T. A., Ali, R. H., Ali, N., Baig, M. M., & Ali, M. S. (2022, November). Docontap: Ai-based disease diagnostic system and recommendation system. In *2022 17th International Conference on Emerging Technologies (ICET)* (pp. 148-153). IEEE.
[24] Shankar, A., Perumal, P., Subramanian, M., Ramu, N., Natesan, D., Kulkarni, V. R., & Stephan, T. (2024). An intelligent recommendation system in e-commerce using ensemble learning. *Multimedia Tools and Applications*, *83*(16), 48521-48537.
[25] Zhu, J., Wu, H., & Yaseen, A. (2022, May). Sensitivity Analysis of a BERT-based scholarly recommendation system. In *The International FLAIRS Conference Proceedings* (Vol. 35).
[26] Patnayakuni, S. P., &Yedidi, N. S. H. (2021). A bert-based deep learning approach for suggestion mining. *Journal of Artificial Intelligence, Machine Learning and Data Science*, *1*(4), 1-9.