Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057071

ARTICLE

# A Fusion Model for Personalized Adaptive Multi-Product Recommendation System Using Transfer Learning and Bi-GRU

**Buchi Reddy Ramakantha Reddy and Ramasamy Lokesh Kumar***

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, 632014, Tamilnadu, India

*Corresponding Author: Ramasamy Lokesh Kumar. Email: lokeshkumar.r@vit.ac.in

Received: 07 August 2024 Accepted: 29 October 2024 Published: 19 December 2024

## ABSTRACT

Traditional e-commerce recommendation systems often struggle with dynamic user preferences and a vast array of products, leading to suboptimal user experiences. To address this, our study presents a Personalized Adaptive Multi-Product Recommendation System (PAMR) leveraging transfer learning and Bi-GRU (Bidirectional Gated Recurrent Units). Using a large dataset of user reviews from Amazon and Flipkart, we employ transfer learning with pre-trained models (AlexNet, GoogleNet, ResNet-50) to extract high-level attributes from product data, ensuring effective feature representation even with limited data. Bi-GRU captures both spatial and sequential dependencies in user-item interactions. The innovation of this study lies in the innovative feature fusion technique that combines the strengths of multiple transfer learning models, and the integration of an attention mechanism within the Bi-GRU framework to prioritize relevant features. Our approach addresses the classic recommendation systems that often face challenges such as cold start along with data sparsity difficulties, by utilizing robust user and item representations. The model demonstrated an accuracy of up to 96.9%, with precision and an F1-score of 96.2% and 96.97%, respectively, on the Amazon dataset, significantly outperforming the baselines and marking a considerable advancement over traditional configurations. This study highlights the effectiveness of combining transfer learning with Bi-GRU for scalable and adaptive recommendation systems, providing a versatile solution for real-world applications.

## KEYWORDS

Personalized recommendation systems; transfer learning; bidirectional gated recurrent units (Bi-GRU); performance metrics; adaptive systems; product reviews

## 1 Introduction

With the explosion of big data [1], particularly in the financial sector, recommendation systems have become an essential part of contemporary life [2]. The rise of online shopping and related platforms has driven academics to develop efficient, tailored e-commerce recommendation systems [3]. Conventional recommendation systems typically employ methods such as collaborative filtering, content-based recommendation, and hybrid recommendation. However, classical collaborative filtering systems face challenges such as cold start along with data sparsity difficulties [4]. Additionally, these methods often fail to capture comprehensive item and user representations [5]. Content-based recommendation approaches require vectorization methods to improve their outcomes [6]. The massive volumes of user data collected via the Internet make it impractical to use handcrafted and lower-level characteristics in practical applications [7]. Analyzing heterogeneous data (labels, text, and pictures) in content-based recommendation algorithms is computationally costly [8,9]. Hybrid recommendation approaches attempt to address data sparsity by integrating auxiliary information. However, due to the complexity of auxiliary information, hybrid approaches often perform poorly [10,11].

Deep learning models provide exceptional performance in the fields of signal processing, computer vision, and language processing, hence opening up novel possibilities for recommendation systems. Personalized recommendation systems can reduce data overload by mining customer preferences, aiding service providers in better decision-making [12]. Key objectives for network service platforms include personalized user experiences and decision support [13,14], as they aggregate vast amounts of news content [15]. Researchers are optimizing personalized recommendation algorithms to enhance accuracy and address information overload [16]. Traditionally, recommender systems use a single data source, like a user's history, for suggestions. However, leveraging multiple information sources is now popular for better precision and applicability. While collaborative filtering models (neighborhood approaches and matrix decomposition) perform well with limited data [17], modern data complexity challenges these methods. Reliable recommendations require dynamic knowledge extraction from user data and deep models that handle large samples [18]. Current personalized recommendation models can improve the capture of data characteristics for more accurate information extraction.

In this research, we propose the PAMR (Personalized Adaptive Multi-Product Recommendation System) model using Transfer Learning and Bi-GRU. The methodology begins with data pre-processing, including denoising and information encoding. Feature extraction involves feature vectorization to reduce dimensionality and user vector embedding to capture preferences and behaviors. The core of the PAMR model employs transfer learning models (AlexNet, GoogleNet, ResNet-50) for attribute-specific feature extraction, with Bidirectional GRU (Bi-GRU) layers capturing sequential dependencies in user-item interactions. An Attention Update Gate-GRU (AUG-GRU) with an attention mechanism refines the model's focus on relevant features, enhancing its learning from complex data patterns.

This study aims to enhance e-commerce recommendations by developing the PAMR model, integrating transfer learning and Bidirectional Gated Recurrent Units (Bi-GRU). It seeks to overcome the limitations of existing systems by adapting to user preferences and diverse product catalogs. Specific goals include creating a scalable model for large datasets from platforms like Amazon and Flipkart, improving recommendation precision through user-generated content, and evaluating the model's performance against traditional Long Short-Term Memory (LSTM)-based models across various metrics. The study also demonstrates the model's adaptability across different product categories and platforms, establishing its robustness in real-world applications.

In the proposed work, we use three datasets: Amazon, Health Product, and Flipkart Health. The PAMR model variants show distinct strengths: AlexNet-Bi-GRU excels on the Amazon dataset with nearly 97% accuracy and F1-score, indicating robust predictive power. GoogleNet-Bi-GRU performs best on the Health Product dataset, especially in recall and F1-score, highlighting its precision for health products. ResNet-50-Bi-GRU leads on the Flipkart Health dataset with superior accuracy and balanced performance, making it the most reliable. Each model's top performance on a specific dataset highlights the importance of matching model architecture to dataset characteristics for optimal performance.

The significant contributions of the illustrated PAMR model for multi-product recommendation can be encapsulated as follows:

- Combines data denoising and information encoding for high-quality data preparation, along with feature vectorization and embedding for capturing detailed user preferences and product attributes.
- Employs transfer learning (AlexNet, GoogleNet, and ResNet-50) and Bi-GRU for dynamic, context-aware attribute-specific feature extraction, crucial for personalized recommendations across varied product categories.
- Integrates AUG-GRU with an attention mechanism to prioritize significant data points, thereby refining the recommendation process based on user-item interactions.
- Validates the model's effectiveness with comprehensive performance metrics and demonstrates its real-world applicability in providing accurate multi-product recommendations.

Below is the outline for the remainder of the article. Section 2 examines research on the best deep and transfer learning recommender systems; Section 3 lays out the approach to be used; Section 4 shows the results of the experimental evaluations of the system's performance on various recommendation tasks; and Section 5 concludes the study.

## 2 Related Works

A literature review of existing articles on sentiment analysis is presented in this section. An information filtering system may filter scene characteristics for personalized advice. In several sectors, personalized suggestions help consumers make decisions and maximize advantages [19]. Personalized recommendations rely on pinpointing user interests [20]. Therefore, whether a recommendation system can effectively derive users' unseen predilections from imperfect user information. It is important to measure the pros and cons of the research topic.

With the advent of deep learning, the recommendation algorithm has seen significant modifications in recent years, and now there is a better way to boost the recommendation system's performance [21]. Deep learning recommendation systems improve upon the quality of recommendations while avoiding the technological problems of the old recommendation mode. The study in [22] employs Long Short-Term Memory (LSTM) to acquire knowledge of the cited context and the scattered representation of scientific publications. Subsequently, this knowledge is used for further analysis, knowledge to measure the correlation and make recommendations based on scientific papers' features; Reference [23] extracts user data information using CNN (Convolutional Neural Networks) and Bi-LSTM (Bidirectional Long Short-Term Memory) networks, improves network performance, and makes targeted recommendations for network users.

A convolutional neural network-based tag recommendation technique collected and merged picture and user interaction elements to provide individualized tag suggestions [24]. A multilayer perceptron-based collaborative filtering recommendation method was examined. Deep learning was effectively used in creating neural collaborative filtering model [25]. Users' fluctuating interests were taken into account, and a customized recommendation approach based on the deep interest network to shape their interests was presented [26]. A system called Attention-based Deep Learning Point of interest Recommendation (ADPR) was introduced to extract user preferences for personalized suggestions. However, when applied to a large dataset, the process of extracting features using this recommendation approach is time-consuming and increases the number of parameters that need to be learned in each cycle [27]. A transfer learning-based deep neural network that used cross-domain information to improve tailored recommendations was examined. Despite high accuracy, these approaches can only extract superficial feature information and cannot extract much deeper information [28].

Similarly, a CNN-Bi-GRU e-commerce product recommendation system was built. To increase sentiment features in collected text, a sentiment dictionary weighed sentiment word vectors. Then, CNN and Bi-GRU models extracted essential context information, which the fully connected layer used to categorize sentiment features. This e-commerce product recommendation system was time-consuming because of deep-learning model hybridization [29]. Contextual feature vectors were mined from the dataset by means of GloVe (Global Vectors for Word Representation) word embedding and TF-IDF (Term Frequency-Inverse Document Frequency). Sentiment analysis was then performed using a hybrid CNN-LSTM architecture [30]. In LSTM network with enhanced word embedding requires more training data, increasing costs. The system becomes more complicated when many models are integrated [31]. Products were recommended using the Broyden–Fletcher Goldfarb–Shanno algorithm and a GRU model. In this investigation, the model took longer to analyze bigger datasets, which was a serious concern [32].

Skip-gram was used to extract contextual and semantic elements from the material. For sentiment analysis, the LSTM network received contextual and semantic data. LSTM network effectiveness was increased by improving weight parameters using adaptive particle swarm optimization. While computationally expensive, traditional LSTM networks learn better with more training data [33]. Collaborative filtering was employed to analyze product and user evaluations for sentiment. The proposed collaborative filtering method partially eliminates cold-start concerns [34]. Data was preprocessed using lemmatization, stemming, and stop-word removal. Possibilistic fuzzy c-means with LDA (Latent Dirichlet Allocation) modeled topics. CNN models categorized terms as neutral, positive, or negative. CNN model performance is somewhat affected by high-dimensional data [35].

Fraudulent reviews were detected using consumer emotions, word context, and a bag of words, outperforming baseline systems in F1-score, accuracy, and AUC (Area Under Curve) [36]. CNN and rapid text embedding increased model complexity [37]. A gated RNN for sentiment analysis faced gradient issues with large datasets [38]. Collaborative filtering faced scalability, cold-start, and synonym issues [39]. A bidirectional convolutional RNN with group-wise augmentation outperformed previous methods but struggled with disappearing gradients [40]. A stacked bi-directional LSTM for sarcasm detection showed promising accuracy but faced overfitting and gradient issues [41].

This study introduces the AM-Bi-LSTM (Adaptive Multi-modal Bidirectional Long Short-Term Memory) network for sequential recommendation modality detection, using bidirectional LSTM to capture multi-modal features and user activities. With a dataset of 14,941 hits from a Japanese teacher's website, a modality attention module enhances suggestion diversity and accuracy, achieving a recall of 0.1105 compared to early-fusion methods [42]. Another model, FDSA-CL (Feature-level Deeper Self-Attention Network with Contrastive Learning), uses vanilla attention and self-attention blocks for feature transitions in sequential recommendation, proving more effective than previous approaches on real-world datasets [43]. To improve user satisfaction, a GRU-based deep learning recommendation model was introduced to combat overfitting and enhance feature extraction through attention processes [44]. Tested on Amazon and MovieLens datasets, it showed improved personalization with variable-length mini-batch allocation. Another approach incorporated browsing duration data and adapted the GRU network to changing user interests, achieving RMSE (Root Mean

Squared Error) of 0.7257 and 0.7869, and MAE (Mean Absolute Error) of 0.5147 and 0.5893, on the Criteo and MovieLens-1M datasets, respectively [45].

Recent advancements in recommendation systems have introduced various innovative models to enhance accuracy and address challenges such as data sparsity, cold starts, and privacy concerns. The FedKGRec model incorporates knowledge graphs (KG) and local differential privacy to safeguard user data, outperforming federated methods in AUC, ACC, and F1 metrics [46]. A contrastive learning-based approach uses hierarchical learning and item relation learning to combat noise and data sparsity in KGs, improving performance on long-tail datasets [47]. GSRec, a combination of Graph Convolutional Networks (GCN) and Transformers, addresses sequence length limitations and demonstrates superior results on six datasets [48]. The DFM-GCN (DeepFM Graph Convolutional Network) model enhances click-through rate (CTR) predictions by integrating DeepFM and GCN, achieving state-of-the-art results on public datasets [49]. Variational autoencoders introduce stochasticity in collaborative filtering, improving latent space representation [50], while the KGG model fuses local and global user interest features for better KG-based recommendations [51]. CF-MGAN (Collaborative filtering with metadata-aware generative adversarial networks), a GAN-based framework, leverages metadata to improve collaborative filtering and prevent biased ratings [52]. Additionally, a Bidirectional Encoder Representations from Transformers (BERT)-based recommendation system is designed for eBay to address data privacy and scalability issues [53], and a combined sentiment analysis and collaborative filtering system enhances product recommendations [54]. Finally, a Transformer model that integrates utility matrix and textual data outperforms traditional models like MLP (Multilayer Perceptron) and SVD (Single Value Decomposition) in improving recommendation accuracy [55].

The developments in Self-Supervised Learning (SSL) on recommender systems have significantly improved accuracy while addressing key challenges like data sparsity, cold starts, and privacy concerns. Models like FedKGRec [46] leverage knowledge graphs and SSL-based local differential privacy to outperform traditional federated approaches in evaluation metrics, while contrastive learning techniques mitigate noise and enhance performance on long-tail datasets. GSRec, combining GCNs (Graph Convolutional Networks) and self-supervised Transformers, handles sequence length limitations effectively [48]. Additionally, DFM-GCN [49] and CF-MGAN [52] utilize SSL to achieve state-of-the-art results in click-through rate predictions and collaborative filtering, respectively, while VAEs [50] enhance latent space representation through stochasticity. BERT-based systems [53] also integrate SSL for privacy-sensitive applications like eBay, and sentiment analysis-driven collaborative filtering further boosts recommendation quality. SSL-driven Transformers, incorporating utility matrix and textual data, outperform traditional models, underscoring the power of self-supervised learning in modern recommendation systems.

The literature review highlights gaps in sentiment analysis and recommendation systems, such as scalability, efficiency with large datasets, cold-start problems, and sparse data issues. Current models struggle with feature extraction, interpretability, and integrating temporal and geographical context effectively, which are crucial for improving accuracy and user trust. Our study introduces the PAMR model to address key gaps in recommendation systems. It combines transfer learning with Bi-GRU networks to handle cold-start and sparsity issues by extracting meaningful features from limited data using pre-trained models (AlexNet, GoogleNet, and ResNet-50). Advanced feature extraction through feature fusion captures diverse and complementary information, while Bi-GRU refines features by capturing sequential dependencies and contextual information. The model effectively handles temporal sequences with an attention mechanism, and future work may incorporate geographical context. Transfer learning reduces computational costs and enhances generalization, leveraging knowledge from large-scale image recognition tasks for better performance and adaptability. These innovations make the PAMR model a robust solution for personalized adaptive multi-product recommendations, improving user experience and satisfaction.

## 3 Proposed System

The block diagram in Fig. 1 delineates the structured workflow of the new and innovate proposed model for an advanced recommendation system, alongside the introduced model, herein referred to as the Personalized Adaptive Multi-Product Recommendation System (PAMR). This system is specifically tailored to enhance the personalization of product recommendations. PAMR is a composite model that harnesses the analytical strengths of three transfer learning models—AlexNet, GoogleNet, and ResNet-50—to meticulously extract attribute-specific features critical for precise recommendation delivery. It is further refined through the application of a Bi-GRU network, which adeptly captures the temporal sequences of user-product interactions. The inclusion of an Attention Mechanism in the AUG-GRU (Attention mechanism-GRU) allows for selective concentration on pivotal aspects of the data, a crucial step for accurate prediction.

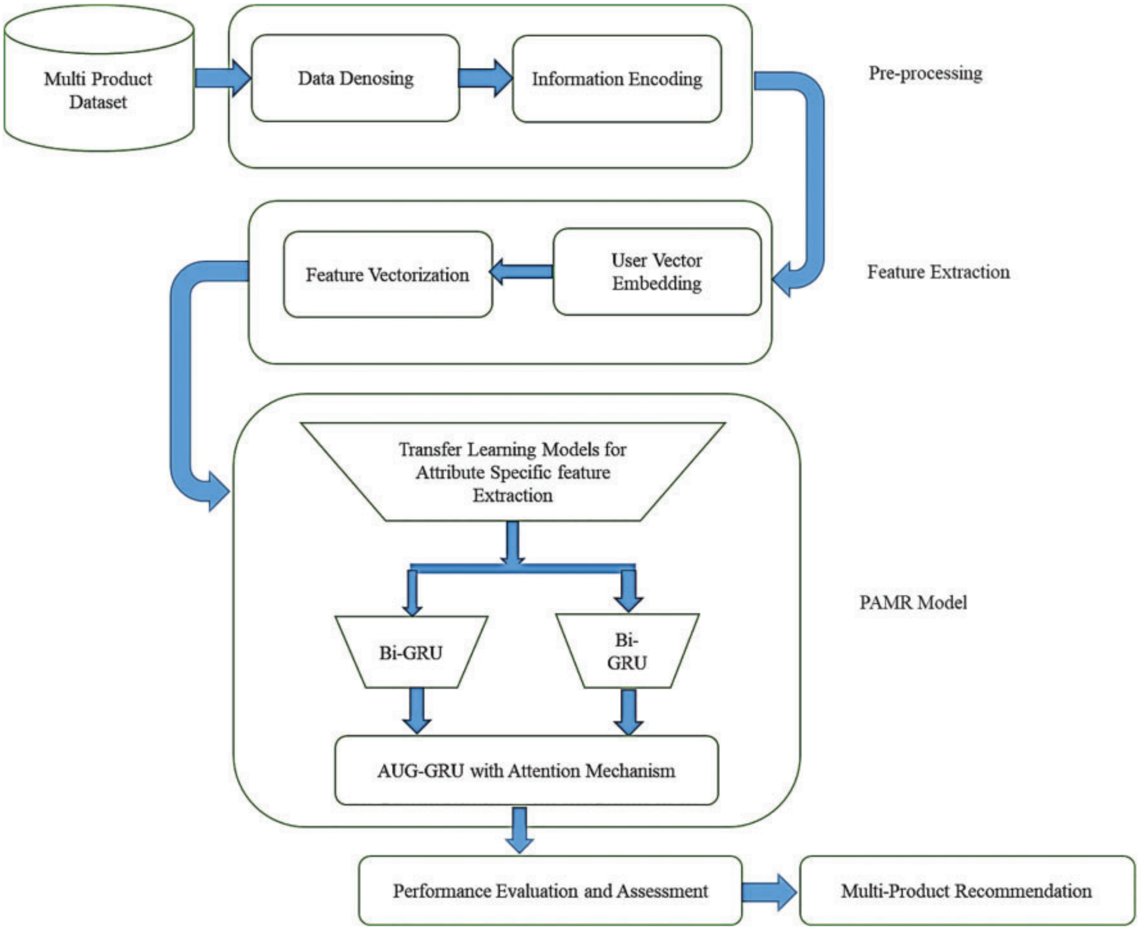
Multi Product Dataset
Data Denosing
Information Encoding
Pre-processing
Feature Vectorization
User Vector Embedding
Feature Extraction
Transfer Learning Models for Attribute Specific feature Extraction
Bi-GRU
Bi-GRU
PAMR Model
AUG-GRU with Attention Mechanism
Performance Evaluation and Assessment
Multi-Product Recommendation

**Figure 1:** Proposed system block diagram

The workflow is segmented into distinct phases, starting with Data Denoising to filter out noise from the raw dataset, followed by Information Encoding, which translates the denoised data into a format that is digestible by machine learning algorithms. Feature Vectorization is then applied, structuring the data into feature vectors, while User Vector Embedding concurrently develops a dense numerical representation of user profiles. These pre-processing steps set the stage for the intricate processing through the trio of cutting-edge CNN models, each fine-tuned for specific attribute extraction, feeding into the sophisticated sequence analysis by the Bi-GRU with attention. The workflow reaches its apex in the Performance Evaluation and Assessment Segment, where the efficacy of the model is rigorously tested against established benchmarks to ascertain the relevance and precision of the recommendations. This thorough process yields the final output of Multi-Product Recommendations, which are scrupulously tailored to the unique preferences of each user through a confluence of deep learning methodologies and sequential pattern analysis.

**Data Denoising**

In the Data Denoising section, noise is defined as any irrelevant or erroneous information, including outliers, duplicate records, and irrelevant text, which can negatively impact the recommendation system's performance. Missing input values are handled through imputation, or by excluding records with a high proportion of missing data. Normalization scales data to a [0, 1] range. Text data cleansing involves removing special characters and punctuation, lowercasing, removing stopwords, and applying lemmatization or stemming. Image data preprocessing includes resizing to a standard dimension, normalizing pixel values to a [0, 1] range, and enhancing images through techniques like histogram equalization and noise filtering to ensure uniform quality and consistency for model training. By denoising the dataset, recommendation systems can ensure that the underlying data used for training models is clean, reliable, and representative of user preferences and product characteristics, ultimately leading to more accurate and relevant product recommendations for users.

**Information Encoding**

Information encoding is the subsequent phase, and it involves converting raw item and user data that is based on text into numeric vectors of defined length. In this stage, we employ two distinct encoding techniques: one for user information and one for product information.

User information encoding in recommendation systems involves transforming various aspects of user behavior, preferences, and characteristics into structured representations. This process encompasses encoding demographic information, historical interactions with products, social network relationships, contextual factors like time and location, and temporal patterns. Techniques such as numerical or categorical encoding, graph-based representations, feature engineering, and temporal modeling are utilized to capture and encode user information effectively. Recommendation systems are able to provide more relevant and effective suggestions by taking into consideration each user's preferences, actions, and contextual aspects via extensive user data encoding. Data about products is encoded using a method called sequence-based encoding. The raw data is first turned into a n × 1024-sized stack of embeddings (E) when given an item information consisting of n words, $\{w_1 ...:w_n\}$. To mark the beginning of a text series, CLS (Classification Token) is a unique word embedding token. A 24-layered learning model takes the embeddings as input and uses them to encode E by computing 16 self-attentions per layer. The input size (n × 1024) is identical to the final encoded output size. Simply said, the encoded output of the token is the final encoded item information.

A user's preference may be represented in vector space by a user vector. You may usually find this kind of graphic in user-generated material like product reviews or profiles. User vectors may also be initialized at random if none of those pieces of information are supplied. Using user vectors to show similarity when several users share similar profile information or reviews is a benefit of this content-based approach.

**User Vector Embedding**

User content data is necessary for the content-based method of information encoding, although it is not always easily accessible. For example, user profiles could include difficult-to-collect private details; user reviews can only be found in certain online spaces where users are authorized to share their opinions on products. A possible alternative to using the given data is to employ vectors that are produced at random. However, recommendation results from random vectors are not up to par with content-based ones. In light of this shortcoming, we provide a new method—user vector embedding—to improve upon item vectors used to produce user vectors at random. Our method for finding user similarity makes use of user-item ratings ($\boldsymbol{(r_{ij})}$) and user vectors that are produced at random. The basic premise is that users will be treated similarly if they rate the same goods with the same ratings.

**Initial Random Vectors:** We start with a set of randomly generated user vectors $u_i$ for each user $i$.

**Item Vectors:** Each item $j$ in our dataset is represented by an item vector $v_j$.

**Objective:** The goal is to refine the user vectors $u_i$ based on the items they have rated. Specifically, for a given rating level $a$, we aim to minimize the following loss function:

$$J_a = -\sum_j log\left(\frac{e^{u_i^T v_j}}{\sum_{k=1}^N e^{u_i^T v_k}}\right) \tag{1}$$

Here, $\{v_k\}$ represents the collection of all item vectors.

**Gradient Computation:** To optimize the user vectors, we compute the gradient of $J_a$ with respect to $u_i$:

$$\frac{dJ_a}{du_i} = \sum\nolimits_j \left(-vj + \sum\nolimits_{k=1}^N \left(\frac{e^{u_i^T v_j}}{\sum_{k=1}^N e^{u_j^T v_k}}.v_k\right)\right) \tag{2}$$

User grouping and user vector embedding are the two phases of our user vector embedding method's deployment. First, the number of items in $\boldsymbol{R}$ is used to establish a list of empty sets, $\boldsymbol{Ia}$, in the user grouping phase. The algorithm will gather all user indices and assign them to the set of $\boldsymbol{Ia|j|}$ if given a specified rating level, $\boldsymbol{a}$. This will happen for each item $\boldsymbol{j}$ for which a rating has been assigned. By applying the grouping method to each potential rating level, $\boldsymbol{A} = \{\boldsymbol{a}\}$, the whole set of $\boldsymbol{I_A} = \boldsymbol{|ia|}$ may be calculated.

The user vector embedding approach proceeds to the second phase as follows: The input of the procedure consists of the entire collection of item vectors, $\boldsymbol{V} = \{\boldsymbol{v1}, \ldots, \boldsymbol{vj}, \ldots, \boldsymbol{vN}\}$, retrieved from the information encoding, and the classification information, $\boldsymbol{I_A} = \boldsymbol{|ia|}$. To compute $J_a$, the algorithm begins by randomly initializing a set of user vectors, $\boldsymbol{Ua}$, for each grouping set indicated by a particular rating, $\boldsymbol{Ia}$. The computer then calculates a gradient of $J_a$ with respect to each user $\boldsymbol{u_i}$, whereby $\boldsymbol{u_i}$ is the number of users who have provided a rating for an item $\boldsymbol{a}$. Once every piece of gradient data has been gathered, the user interface is updated via the use of gradient descent. This approach utilizes the aggregated gradient data, which consists of user vectors learned on all rating levels.

**Feature Vectorization**

This section describes how vectorization is performed using the TF-IDF technique after the acquired customer reviews have been denoised. The TF is determined as the ratio of the number of repetitive words in a denoised user review to the total number of words in the user review. Correspondingly, the *IDF* is the ratio of the number of customer reviews to the total number of customer reviews with repetitive words. The mathematical expressions for the *TF* and *IDF* are given in Eqs. (3)–(6), where the term frequency $t$ in a user review $d$ is represented as $f_d(t)$ and the corpus of customer reviews is denoted as $D$. The obtained vectors are passed to the PAMR model for product recommendation.

$$TF\,(t,d) = \frac{f_d(t)}{\max f_d(t)} \tag{3}$$

$$IDF\,(t,D) = ln\left(\frac{|D|}{|\{d \in D\colon\ t \in d\}\,|}\right) \tag{4}$$

$$TF - IDF\,(t,d,D) = TF\,(i,d)\ XIDF(t,D) \tag{5}$$

$$TF - IDF'\,(t,d,D) = \frac{IDF\,(t,D)}{|D|} + TF - IDF(t,d,D) \tag{6}$$

Transfer learning algorithms are used to extract high-quality features from product characteristics data. To optimize performance and reduce computational complexity, we employ AlexNet, GoogleNet, and ResNet-50 architectures on the product dataset. The resulting output is feature maps generated by these trained transfer learning models $F_{AlexNet}$, $F_{GoogleNet}$, and $F_{ResNet-50}$. The visuals based on the feature maps compressed by $F_{AlexNet}$, $F_{GoogleNet}$, and $F_{ResNet-50}$.

In this paper, AlexNet, GoogleNet, and ResNet-50 are referred to as “transfer learning architectures” because they leverage pre-trained models initially trained on large datasets such as ImageNet. Transfer learning involves using these pre-trained models to transfer learned features to a new, often smaller, dataset. This approach allows the model to apply learned representations from a large and diverse dataset to a more specific task with limited data, significantly enhancing performance and reducing training time. By utilizing the powerful feature extraction capabilities of these pre-trained models, we can more effectively handle the complex and varied data encountered in the recommendation system.

The choice of AlexNet, GoogleNet, and ResNet-50 as the pre-trained models is strategic, considering the different strengths each model offers. AlexNet is included because of its effectiveness in extracting essential visual features from simpler datasets. It excels at recognizing basic attributes like shapes, colors, and textures, making it suitable for tasks requiring visual recognition. GoogleNet, on the other hand, is known for its inception modules, which extract features at different scales, offering a more holistic view of the product attributes. This multi-scale feature extraction is particularly beneficial in handling complex datasets such as those involving health products, where multiple detailed attributes need to be analyzed simultaneously. ResNet-50 stands out for its ability to perform deep residual learning, allowing the model to go deeper without losing important information. This makes it ideal for datasets where fine-grained, complex details are essential for accurate product recommendations, as seen in e-commerce platforms like Flipkart.

Thus, proposed PAMR model is introduced at this stage. In the feature vectorization phase, features are extracted from the input data F using three different pre-trained models: AlexNet, GoogleNet, and ResNet-50. Each model extracts elevated features from the input as follows:

Apply AlexNet to $F$ to obtain deep features $F_{AlexNet}$:

$$F_{AlexNet} = AlexNet\,(F) \tag{7}$$

Apply GoogleNet to $F$ to obtain deep features $F_{GoogleNet}$:

$$F_{GoogleNet} = GoogleNet(F) \tag{8}$$

Apply ResNet-50 to $F$ to obtain deep features $F_{ResNet-50}$:

$$F_{ResNet-50} = ResNet - 50(F) \tag{9}$$

**Feature Fusion:**

The outputs from AlexNet, GoogleNet, and ResNet-50 are combined through a weighted feature fusion technique. This method ensures that the strengths of each pre-trained model are integrated optimally to form a robust and comprehensive feature representation.

The features are fused to create a comprehensive feature representation $F_{Fused}$, which leverages the strengths of each specific model. The feature fusion process is defined as:

$$F_{Fused} = \lambda F_{AlexNet} + (1 - \lambda)\, F_{GoogleNet} + (1 - \lambda) F_{ResNet-50} \tag{10}$$

where λ is a tunable parameter that determines the weight assigned to each model's features.

The fusion coefficient λ optimizing the parameter is crucial for achieving the optimum functioning of the system of recommendation. The process of determining λ typically involves, begin with an initial guess for λ, often set to 0.5 or based on heuristics. Perform a grid search over λ values (e.g., 0 to 1 in 0.1 increments) to find the optimal balance between model features. Use cross-validation to evaluate fused features' performance and test the selected λ on a separate set for generalization. This optimal fusion coefficient enhances the personalized adaptive multi-product recommendation system's performance and robustness.

This approach balances the high-level semantic features captured by AlexNet, the multi-scale features from GoogleNet, and the fine-grained details from ResNet-50. The fused features are then passed through a Bidirectional Gated Recurrent Unit (Bi-GRU), which models the temporal and sequential dependencies in user-item interactions, refining the recommendation process by focusing on relevant data patterns. This feature fusion process allows the system to handle challenges such as cold starts and sparse data effectively, resulting in more accurate and personalized recommendations across different product categories.

This study optimizes the GRU model, reduces the overfitting and accurately and efficiently extracts features from user data using the attention mechanism module. The goal is to attain efficient and accurate user custom-made recommendation by improving the recommendation algorithm's performance. At now, the GRU determines the hidden layer output $g_t$ by using the input $r_t$ and the hidden layer output $g_{t-1}$ from earlier in time. Eqs. (11)–(14) demonstrate the update of the GRU network's internal unit, which consists of simply an update gate $u_t$ and a reset gate $v_t$. The GRU network's internal unit structure is, nevertheless, rather basic.

$$Vt = \sigma(e_v.\,[g_{t-1}, r_t]) \tag{11}$$

$$u_i = \sigma(e_u.\,[g_{t-1}, r_t]) \tag{12}$$

$$\bar{g}_t = Tanh\left(w_{\bar{g}}.[v_t * g_{t-1}, r_t]\right) \tag{13}$$

$$g_t = (1 - u_t) * g_{t-1} + u_t * \bar{g}_t \tag{14}$$

The sigmoid function, often used in binary classification, includes a weight matrix $w$ and an activation function $\sigma$. Adjusting the update gate $u_t$ regulates the amount of prior state information added to the current state, with a higher value adding more information. The reset gate $v_t$ controls the previous state information is written into the current state's candidate set $\underline{d}_t$, with a lower value indicating less information is added.

As time goes on and the outside world exerts its constant pull, so do people's interests and hobbies. Users' interests and hobbies tend to cluster around certain times as well. Hence, relying just on the initial one-layer GRU is insufficient for simulating this shift in user interest.

By using the auxiliary loss function, the data extraction layer may extract the expression of interest features from the user interaction sequence. A layer of GRU must be added to the initial GRU network in to represent the learning of evolution and the attentiveness of user attention. Incorporating GRU's sequence learning capabilities with the attention mechanism's local activation ability allows us to model interest evolution as it actually occurs, with local activation performance simulating interest concentration and mitigating the effect of irrelevant interest on the model's output.

The attention $\boldsymbol{q_t}$ calculation method is illustrated in Eq. (15), where $\boldsymbol{s_q}$ is the sum of the candidate goods' embedding vectors, $c$ is the model parameter matrix, $\boldsymbol{c} \in \boldsymbol{R}^{\,nG\times nQ}$ where $\boldsymbol{nG}$ is the hidden layer dimension size, and $\boldsymbol{nQ}$ is the dimension size of the candidate goods' embedding vectors. The attention $\boldsymbol{q_t}$ value can represent the association among the contender goods and the input $\boldsymbol{r_t}$ .

$$q_t = \frac{exp\left(r_t c s_q\right)}{\sum_{i=1}^{n} exp\left(r_i c s_q\right)} \tag{15}$$

This research integrates the attention mechanism with the update gate in a GRU, creating the AUG-GRU to address dimension disparities often overlooked by using attention scores for state updates. To minimize the effect of irrelevant input on the model, the update gate in the suggested AUG-GRU network keeps the original input's dimensional information and scales the attention score to all the dimensions of the update gate. To rephrase, AUG-GRU is able to facilitate the model's interest extraction in a more seamless manner and more effectively prevent interference caused by abrupt changes in interest, Eqs. (16) and (17) illustrate the transformation of AUG-GRU into GRU, with $u_t$ representing the original GRU's update gate, $u_t$ denoting the redesigned AUG-GRU's update gate, and $g'_t$ and $g'_{t-1}$ denoting the hidden layer states of AUG-GRU.

$$\bar{u}_t = q_t * u_t \tag{16}$$

$$g'_t = (1 - \bar{u}_t)\, .g'_{t-1} + \bar{u}_t * g'_t \tag{17}$$

The utilization of a GRU to extract user sequence interest is the most significant enhancement of the suggested model when contrasted with the conventional recommendation system model. While doing so, we guarantee the correct extraction of users' sequence characteristics by using a variant GRU structure with an attention mechanism, which allows us to better adjust to the trend of users' interests as they evolve.

### *3.1 Proposed Model Architecture*

This nomenclature is presented in Fig. 2, which combines transfer learned features with GRU based recommendations and is specifically designed for personalized product recommendation in a multi-product environment. Fig. 2 illustrates a multi-stage deep learning architecture for product recommendation, starting with input features that undergo initial processing through three prominent convolutional neural networks—AlexNet, GoogleNet, and ResNet-50—each designed to extract complex attribute-specific features from data.

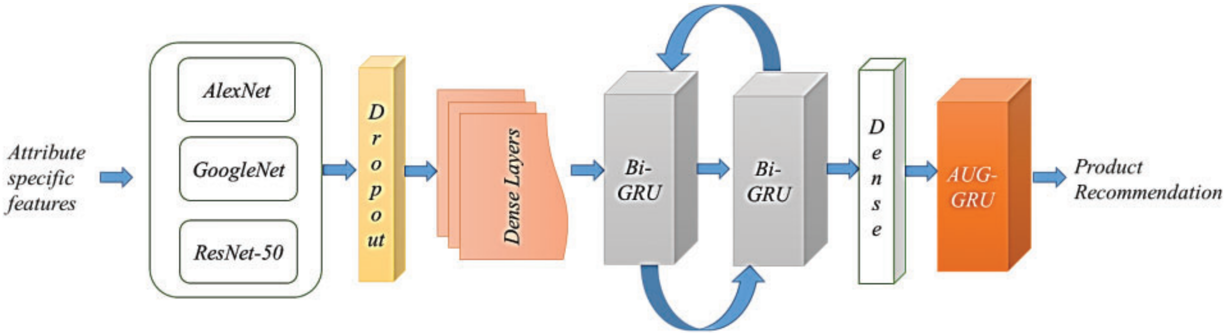

**Figure 2:** Proposed model architecture multi-product recommendation system

The proposed PAMR model utilizes three prominent convolutional neural networks—AlexNet, GoogleNet, and ResNet-50. These networks serve specific roles within the overall architecture, primarily focusing on efficient feature extraction from the dataset to enhance recommendation accuracy. In the PAMR model, AlexNet captures essential visual features, GoogleNet's inception modules extract diverse multi-scale features, and ResNet-50 captures fine-grained details with its deep residual learning, enhancing the recommendation system's ability to differentiate items and understand complex product relationships.

In the Personalized Adaptive Multi-Product Recommendation System (PAMR), specific layers from AlexNet, GoogleNet, and ResNet-50 are utilized for feature extraction. For AlexNet, the fc7 layer, a fully connected layer, is employed. This layer is chosen because it captures high-level semantic information, which is crucial for recognizing essential product attributes like color and shape in large-scale datasets. The GoogleNet model uses the last inception layer before classification. GoogleNet's inception modules capture features at multiple scales, allowing the model to analyze diverse product characteristics simultaneously, making it ideal for complex datasets. Lastly, ResNet-50 utilizes the average pooling layer (avg_pool), which extracts fine-grained details and ensures that the deeper layers maintain important representational power, especially in handling complex product interactions.

The features extracted by AlexNet, GoogleNet, and ResNet-50 are then fused to form a comprehensive feature representation. This fusion process, as described in Eq. (10), combines the strengths of each network, allowing the PAMR model to leverage a rich set of features for more accurate recommendations. The concatenated feature vector is then fed onto a bidirectional gated recurrent unit (Bi-GRU) the system, which captures both the spatial and sequential dependencies in the interactions between the user and the object. A technique of attention is used to direct emphasis towards the most relevant characteristics of the GRU outputs. The context vector, denoted as $\boldsymbol{c}$, is used to generate the ultimate prediction. The output layer generates the suggested goods by using the processed characteristics:

$$\hat{y} = \sigma\left(W_c c + b_c\right) \tag{18}$$

where $\boldsymbol{\sigma}$ is the activation function, $\boldsymbol{W_c}$ are the weights, and $\boldsymbol{b_c}$ is the bias term. These advanced convolutional neural networks for feature extraction, the PAMR model meritoriously addresses the challenges of data sparsity and cold starts, providing a robust and scalable solution for personalized adaptive multi-product recommendations.

**Bi-GRU Parameters and Configuration:**

The Bidirectional Gated Recurrent Unit (Bi-GRU) in the PAMR model is crucial for capturing both spatial and sequential dependencies in user-product interactions. Its bidirectional nature allows it to access information from both past and future interactions, which enhances its ability to model the temporal evolution of user preferences over time. In terms of configuration, the hidden layer size for the Bi-GRU is typically set between 128 to 256 units, which balances computational efficiency and accuracy. The model uses two layers of Bi-GRU to capture both short-term and long-term dependencies in user interactions. Additionally, dropout is applied with a rate of 0.5 to prevent overfitting and ensure generalization. The Adam optimizer, with a learning rate of 0.001, is used for training the Bi-GRU, allowing it to converge quickly while maintaining high accuracy.

**Attention Update Gate (AUG) Mechanism:**

The Attention Update Gate (AUG) mechanism plays a vital role in enhancing the feature extraction process within the Bi-GRU framework. The AUG mechanism is integrated to ensure that the model focuses on the most relevant features from user-product interactions, addressing the challenge of noisy or irrelevant data. The attention mechanism works by assigning attention scores using a context vector $C$ that weights the outputs of the Bi-GRU. The softmax function ensures that these attention scores are normalized, allowing the model to focus on the most significant features. The AUG mechanism modifies the update gate of the GRU, adjusting the amount of information retained or discarded based on the attention scores. This ensures that the most important information is kept in the current state while irrelevant features are minimized, thereby improving the model's overall learning efficiency.

**Optimization during Training:**

The optimization process for both the Bi-GRU and the AUG mechanism is essential for improving the model's accuracy. The PAMR model uses cross-entropy loss and focal loss functions to handle class imbalances, which are common in real-world datasets. During training, backpropagation through time (BPTT) is used to update the weights of the Bi-GRU and adjust the attention scores. To prevent overfitting, the model employs early stopping, ensuring that training halts when the validation loss no longer improves. Additionally, regularization techniques like L2 weight decay are applied to the GRU layers to maintain the model's generalization ability, and gradient clipping is used to prevent exploding gradients, ensuring stable training even with deep architectures.

The combination of the Bi-GRU's ability to model sequential data and the attention mechanism's capability to focus on relevant information leads to significant improvements in recommendation accuracy. By optimizing the attention scores and refining the update gate mechanism through the AUG-GRU framework, the model becomes highly efficient at handling noisy, high-dimensional datasets often encountered in real-world e-commerce systems. The result is a system that delivers highly accurate recommendations, as evidenced by the high precision, recall, and F1-scores achieved across various datasets. The PAMR model's ability to adapt to different user behaviors and product categories demonstrates its robustness and effectiveness in e-commerce recommendation systems.

**Sequence Modeling and Recommendation:**

For sequential and temporal modeling, a Bidirectional Gated Recurrent Unit (Bi-GRU) is used. The bidirectional structure of the GRU allows the network to access both past and future context, which is especially useful when dealing with time-sequenced data or when the input order is relevant to the output:

- Bi-GRU layers capture sequential patterns and dependencies in the data. Gated Recurrent Units are a variant of recurrent neural networks that are particularly good at preserving long-term dependencies and mitigating the vanishing gradient problem thanks to their gating mechanisms.
- AUG-GRU could suggest an 'Augmented GRU' or an 'Attention-based GRU'. It is not a standard term, so it would likely refer to a GRU that has been enhanced with either additional data augmentation techniques or an attention mechanism to improve the model's focus on relevant features for prediction.

### *3.2 Proposed Model Algorithm*

The PAMR (Personalized Adaptive Multi-Product Recommendation) algorithm described in Algorithm 1, leverages the strengths of multiple transfer learning architectures (AlexNet, GoogleNet, and ResNet-50) combined with a Bi-GRU (Bidirectional Gated Recurrent Unit) with an attention mechanism. This approach is designed to excel in multi-product recommendations by extracting and fusing high-level features, which are then processed to generate accurate recommendations.

**Algorithm 1:** PAMR model

```
% Step 1: Initialize PAMR
% Load pre-trained AlexNet, GoogleNet, and ResNet-50 models from Deep Learning Toolbox
alexNetModel = alexnet;
googleNetModel = googlenet;
resNetModel = resnet50;
% Prepare a GRU-based sequence model-Initialize with default parameters or custom settings
gruModel = [ ]; % Placeholder for the actual GRU model initialization
% Step 2: Input Preprocessing
% For each product input:
for i = 1:numProducts
inputAttributes = loadProductAttributes(attributePaths{i}); % Load attribute data
% Process attributes for model input requirements
processedAttributes = preprocessAttributes(inputAttributes);
% Step 3: Feature Extraction with CNNs
alexNetFeatures = activations(alexNetModel, processedAttributes, 'fc7'); % Feature extraction from AlexNet
googleNetFeatures = activations(googleNetModel, processedAttributes, 'last'); % Last layer before classification in GoogleNet
resNetFeatures = activations(resNetModel, processedAttributes, 'avg_pool'); % Average pooling layer in ResNet-50
% Step 4: Feature Fusion
combinedFeatures = fuseFeatures(alexNetFeatures, googleNetFeatures, resNetFeatures); % Fuse features from all three CNNs
% Step 5: Sequence Processing with GRU
sequenceOutput = processSequence(gruModel, combinedFeatures); % Process combined features with the GRU model
```

(Continued)

**Algorithm 1 (continued)**

```
% Prepare data for product recommendation-features and corresponding product scores or
    classifications
featureMatrix(i, :) = sequenceOutput; % Assuming each row is a feature vector
scoreVector(i) = productScores(i); % Assuming you have a corresponding score or classification for
    each product
end
% Step 6: Product Recommendation with Sequence Model
% Train the GRU sequence model with the combined features and product scores
gruModel = trainSequenceModel(featureMatrix, scoreVector);
% Step 7: Recommendation Generation
% For each new set of product attributes, repeat Steps 2–5 to obtain the feature vector
% Then, use the trained sequence model to predict recommended products
recommendedProducts = predict(gruModel, newFeatureMatrix);
% newFeatureMatrix obtained similarly to featureMatrix
% Step 8: Output Interpretation
% Analyze recommendedProducts to generate a list of product recommendations
% Present recommendations to the user or further refine the list based on additional user feedback.
```

## 4 Results and Analysis

Google's Tensorflow, a deep learning platform, is used for building the PAMR model, which is then tested for performance. The Tensorflow framework may simplify development by integrating models like GRU. In Table 1, we can see the exact setup of the experimental setting.

**Table 1:** Computing environment for experimental research

| | |
|---|---|
| CPU | Intel i5 |
| GPU | P-100 |
| RAM | 16 GB |
| Language | Python |
| Platform | Tensorflow |

The Personalized Adaptive Multi-Product Recommendation (PAMR) model was evaluated on Amazon, Flipkart, and a specialized Health Product dataset. This model contains hyperparameter settings which was trained for 30 epochs using the Adam optimizer with a learning rate of 0.001, a batch size of 32, and a dropout rate of 0.5. Regularization methods, including weight decay and L2, were applied, with data shuffling enabled. Model checkpointing callbacks were used, and cross-entropy loss and focal loss served as the loss functions. Softmax was employed as the activation function, ensuring consistent comparison across datasets.

In the study on the PAMR model, evaluate the model's performance in terms of accuracy and loss. The PAMR model was trained with consistent network parameter settings, including 30 training epochs, the Adam optimizer with a learning rate of 0.001, a batch size of 32, and a dropout rate of 0.50. Regularization techniques such as weight decay and L2 were employed, with data shuffling enabled.

Callback mechanisms were activated for model checkpointing, and the loss functions used were cross-entropy loss and focal loss. The softmax function served as the activation function. This setup ensured a robust and fair comparison of the model's effectiveness across various datasets.

### 4.1 Dataset Description

In our study, we utilized three extensive datasets from Amazon, Flipkart, and a specialized Health Product dataset. The Amazon dataset includes over 1.4 million products and 233.1 million reviews with ratings, prices, and sales data up to September 2023 [56]. The Flipkart dataset details around 21,889 health-related products, including names, prices, MRPs (Material Requirement Planning), discounts, and customer ratings [57]. The Health Product dataset includes reviews and ratings for 20,726 items, covering nutritional supplements and medical equipment, segmented into medical and fitness-related products [57]. These datasets are crucial for training and validating the PAMR model, ensuring its adaptability to diverse e-commerce environments and user preferences. Preprocessing involved noise reduction, normalization, and text vectorization to ensure high data quality. The scale and variety of these datasets enable robust testing of the model across different product types and consumer interactions, providing a solid foundation for evaluating its precision, adaptability, and effectiveness in enhancing personalized recommendations.

### 4.2 Performance Assessment

The proposed work's performance is assessed using accuracy, recall, precision, and F1-score metrics, offering a multifaceted view of model effectiveness. Accuracy reflects overall correctness, recall measures the model's ability to identify true positives, precision focuses on minimizing false positives, and the F1-score harmonically combines precision and recall for a balanced performance indicator. The accuracy, recall, precision and F1-score metrics [57] in the proposed work can be computed using the succeeding

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{19}$$

$$Recall = \frac{TP}{TP + FN} \tag{20}$$

$$Precision = \frac{TP}{(TN + FP)} \tag{21}$$

$$F1 - score = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{22}$$

Here, $TP$ represents True Positive, $TN$ represents True Negative, $FP$ represents False Positive, whereas $FN$ represents False Negative (not identified).

### 4.3 Performance on Amazon Dataset

The performance of various models on the Amazon dataset, as outlined in the Table 2, reveals that the AlexNet-Bi-GRU model outperforms the others with an impressive accuracy of 96.9%, precision of 96.2%, recall of 95.9%, and an F1-score of 96.97. This model demonstrates the highest overall effectiveness in correctly predicting and classifying instances within the dataset, combining a superior ability to identify relevant instances (high recall) with a remarkable precision that minimizes false positives. The GoogleNet-Bi-GRU and ResNet-50-Bi-GRU models follow in performance, showcasing commendable results but not reaching the high benchmark set by the AlexNet-Bi-GRU model. The excellence in both precision and recall, balanced effectively with the highest F1-score, underscores the AlexNet-Bi-GRU model's superior capability in handling the Amazon dataset, making it the best performing model in this assessment. When a fusion of models is applied on feature extraction, then model accuracy is enhanced to 97.1%.

**Table 2:** PAMR model performance on Amazon dataset

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|
| AlexNet-Bi-GRU | 96.9 | 96.2 | 95.9 | 96.97 |
| GoogleNet-Bi-GRU | 94.6 | 94.3 | 94.8 | 94.55 |
| ResNet-50-Bi-GRU | 95.8 | 95.82 | 95.71 | 95.77 |

Fig. 3 illustrates the training process of the PAMR model on the Amazon dataset, charting accuracy and loss over approximately 30 epochs. Initially, there is a rapid increase in training and validation accuracy and a sharp decrease in loss, indicative of significant learning. As the epochs progress, both accuracy metrics plateau, reflecting a stabilization of learning, while the loss metrics level off with a slight uptick in validation loss, hinting at potential overfitting. This trend suggests that the model has reached its optimal learning capacity within the given dataset, achieving a balance between understanding the training data and generalizing to new, unseen data.

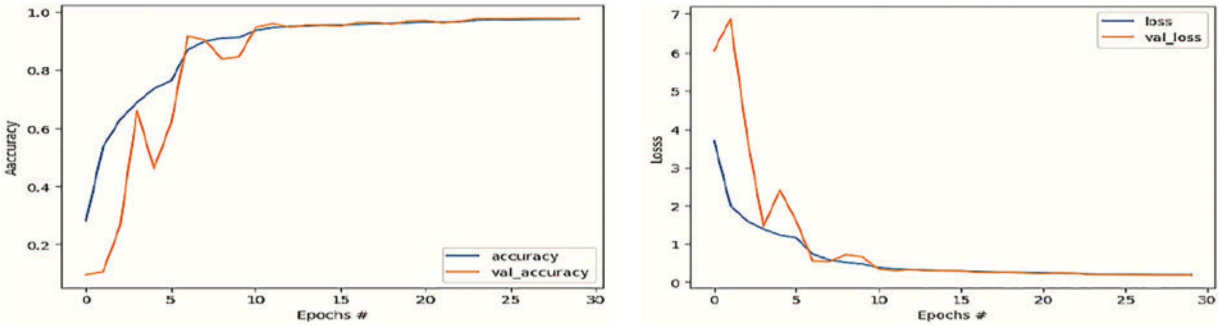

**Figure 3:** Performance of accuracy (%) and loss function of the PAMR model on Amazon dataset for product recommendation

### *4.4 Performance on Health Product Dataset*

Table 3 presents the performance of three PAMR models on the Health Product Dataset: AlexNet-Bi-GRU, GoogleNet-Bi-GRU, and ResNet-50-Bi-GRU. GoogleNet-Bi-GRU is the top performer with 96.25% accuracy, 95% precision, 97.5% recall, and a 96.22% F1-score. AlexNet-Bi-GRU achieves 95% accuracy but has lower precision (89%), recall (87%), and F1-score (88.42%). ResNet-50-Bi-GRU shows consistent performance with 94.6% accuracy, 94.3% precision, 94.8% recall, and a 94.55% F1-score. GoogleNet-Bi-GRU is the most effective model, and model fusion enhances accuracy to 96.8%.

**Table 3:** PAMR model performance on Health Product dataset

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|
| AlexNet-Bi-GRU | 95.0 | 89.0 | 87.0 | 88.42 |
| GoogleNet-Bi-GRU | 96.25 | 95.0 | 97.5 | 96.22 |
| ResNet-50-Bi-GRU | 94.6 | 94.3 | 94.8 | 94.55 |

Fig. 4, compares the performance of three PAMR models—AlexNet-Bi-GRU, GoogleNet-Bi-GRU, and ResNet-50-Bi-GRU—on the Health Product dataset, using the metrics of performance metrics. It illustrates that the GoogleNet-Bi-GRU model achieves the highest recall and a competitive F1-score, suggesting it is particularly adept at identifying relevant instances without compromising balance between precision and recall. Although the ResNet-50-Bi-GRU doesn't lead in any individual category, it shows consistently high performance across all metrics, which indicates robust overall performance. In contrast, the AlexNet-Bi-GRU, while offering commendable recall, lags in precision, which affects its F1-score. Consequently, GoogleNet-Bi-GRU and ResNet-50-Bi-GRU appear to be more effective for this dataset, with GoogleNet-Bi-GRU slightly edging out in terms of recall and F1-score.

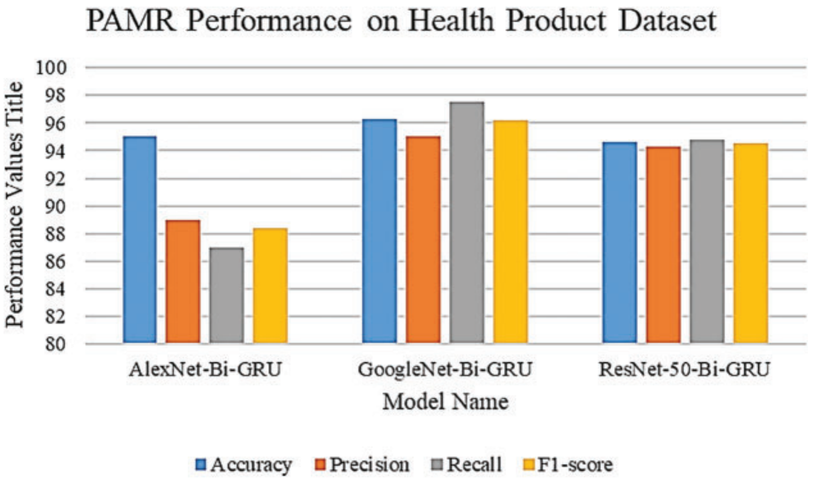

**Figure 4:** PAMR model performance on Health Product dataset

### *4.5 Performance on Flipkart Health Dataset*

Table 4 compares the performance of three PAMR models on the Flipkart Health dataset, focusing on performance metrics. ResNet-50-Bi-GRU attains the premier accuracy (95.79%), precision (95.82%), and F1-score (95.77%), making it the most effective model. GoogleNet-Bi-GRU also performs well with an accuracy of 92.81%, precision of 93.19%, and F1-score of 92.02%. AlexNet-Bi-GRU, while having the highest recall (94%), lags in accuracy (91%) and precision (89%), resulting in an F1-score of 91.87%. When feature extraction models are fused, the overall accuracy improves to 96.1%, highlighting the advantage of model fusion.

**Table 4:** PAMR model performance on Flipkart Health dataset

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|
| AlexNet-Bi-GRU | 91.0 | 89.0 | 94.0 | 91.87 |
| GoogleNet-Bi-GRU | 92.81 | 93.19 | 92.85 | 92.02 |
| ResNet-50-Bi-GRU | 95.79 | 95.82 | 95.71 | 95.77 |

The provided Fig. 5 compares the performance of the AlexNet-Bi-GRU, GoogleNet-Bi-GRU, and ResNet-50-Bi-GRU models on the Flipkart Health dataset across key performance metrics. ResNet-50-Bi-GRU leads in all categories, showcasing its superiority with the highest marks in accuracy, precision, and F1-score, indicating its excellent overall performance. GoogleNet-Bi-GRU also shows robust results with a slightly lower accuracy and F1-score than ResNet-50-Bi-GRU, but still above the AlexNet-Bi-GRU model, which has the lowest performance in accuracy and precision but not in recall. This suggests that while AlexNet-Bi-GRU is slightly better at identifying all relevant cases, it does not match the precision and overall accuracy of the ResNet-50-Bi-GRU model, making the latter the best performing model on the Flipkart Health dataset as per the chart.

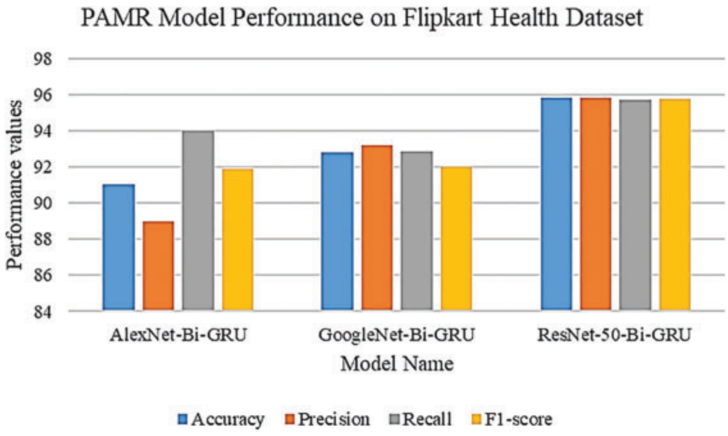

**Figure 5:** PAMR model performance on Flipkart dataset

To understand the impact of each module in the PAMR model, we conducted an ablation study detailed in Table 5. This study systematically removes or alters key components and observes the resulting changes in performance metrics. The components analyzed include the transfer learning models (AlexNet, GoogleNet, ResNet-50), the feature fusion process, and the Bi-GRU with an attention mechanism.

**Table 5:** Ablation study of the PAMR model

| Model variant | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|
| Base Model (BM) | 84.5 | 84.1 | 84.3 | 84.2 |
| AlexNet-Bi-GRU (AN) | 90.3 | 90.0 | 90.5 | 90.25 |

(Continued)

**Table 5 (continued)**

| Model variant | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|---|
| GoogleNet-Bi-GRU (GN) | 92.1 | 91.8 | 92.3 | 92.05 |
| ResNet-50-Bi-GRU (RN) | 93.4 | 93.1 | 93.6 | 93.35 |
| PAMR-woFF | 94.2 | 93.9 | 94.4 | 94.15 |
| PAMR-woAtt | 95.6 | 95.3 | 95.8 | 95.55 |
| Full PAMR | 96.9 | 96.2 | 96.5 | 96.97 |

The analysis shows that the base model with only Bi-GRU (BM) performs the lowest, emphasizing the need for transfer learning models. Each transfer learning model (AN, GN, RN) significantly improves performance, with ResNet-50 (RN) yielding the highest improvement. The PAMR-woFF variant's performance drop compared to the full PAMR model highlights the importance of feature fusion from multiple transfer learning models. The PAMR-woAtt variant's reduced performance underscores the crucial role of the attention mechanism in Bi-GRU. Overall, the full PAMR model outperforms all variants, validating the integration of transfer learning, feature fusion, and the attention mechanism within the Bi-GRU framework. This ablation study provides insights into the specific impact and importance of each component in the PAMR model, confirming that each element contributes significantly to the model's superior performance.

### *4.6 Performance Comparison with the Existing Studies*

Table 6 provides a performance comparison of various models on an unspecified dataset, ranked by accuracy. Among the models, the GoogleNet-Bi-GRU stands out with the highest accuracy of 96.25%, closely followed by the AlexNet-Bi-GRU at 95%. The ResNet-50-Bi-LSTM model also demonstrates robust presentation with an accuracy of 93%. These models outperform the more traditional LSTM-based networks (AlexNet-LSTM, GoogleNet-LSTM, ResNet-50-LSTM) and their bidirectional counterparts (AlexNet-Bi-LSTM, GoogleNet-Bi-LSTM), indicating the potential benefits of GRU cells over LSTM cells in these configurations. Additionally, the models surpass the W-RNN, i-SVM (support vector machines), Neural Collaborative Filtering (NCF), Graph Neural Networks (GNNRec), Attention-based Deep Learning (ADPR), Variational Autoencoders (VAE-CF) and BERT4Rec (Transformer-based) models, which show the W-RNN, i-SVM at lower accuracies at 91.46% and 92.7% respectively, further suggesting the superiority of the PAMR models that use Bi-GRU units for this specific task.

**Table 6:** Performance comparison of PAMR with existing models on Amazon Product dataset

| Model | Accuracy (%) |
|---|---|
| AlexNet-LSTM [57] | 86.23 |
| GoogleNet-LSTM [57] | 87.13 |
| ResNet-50-LSTM [57] | 88.12 |
| AlexNet-Bi-LSTM [57] | 84.62 |
| GoogleNet-Bi-LSTM [57] | 88.5 |

(Continued)

**Table 6 (continued)**

| Model | Accuracy (%) |
|---|---|
| ResNet-50-Bi-LSTM [57] | 93 |
| AlexNet-Bi-GRU | 95 |
| GoogleNet-Bi-GRU | 96.25 |
| ResNet-50-Bi-GRU | 94.6 |
| W-RNN [58] | 91.46 |
| i-SVM [59] | 92.7 |
| Neural Collaborative Filtering (NCF) [25] | 93.1 |
| Graph Neural Networks (GNNRec) [60] | 94.3 |
| Attention-based Deep Learning (ADPR) [27] | 95.2 |
| Variational Autoencoders (VAE-CF) [61] | 92.8 |
| BERT4Rec (Transformer-based) [62] | 96.3 |

In our proposed PAMR model, we have presented an explicit performance evaluation with contemporary literature on product recommendation under Table 5. A detailed comparative analysis of the PAMR model against various baseline models using the Health Product dataset. This dataset was chosen for its relevance and the rigorous challenge it poses due to its diverse and extensive data attributes, which are ideal for testing the robustness and efficacy of our recommendation system. It is important to note that we opted not to use large language models (LLMs) in our comparisons due to their extensive resource requirements and our focus on more computationally efficient solutions suitable for real-time e-commerce environments. Instead, our baselines include other sophisticated models such as GoogleNet-Bi-GRU and ResNet-50-Bi-GRU, which are more comparable in terms of computational efficiency and are currently considered advanced in the field of product recommendation systems.

The baseline comparison was conducted directly on the same datasets to ensure a fair and accurate assessment. Each model was evaluated based on performance metrics that comprehensively reflect the performance in a real-world application scenario. The PAMR model demonstrated superior performance across these metrics, confirming its effectiveness and the innovative integration of Bi-GRU and transfer learning techniques.

State-of-the-art collaborative filtering (CF) models such as Neural Collaborative Filtering (NCF), Variational Autoencoders (VAEs), matrix factorization with side information, Graph Neural Networks for Collaborative Filtering (GNNRec), and hybrid approaches provide robust solutions for handling sparse and noisy data in recommendation systems. NCF uses deep learning techniques to model complex user-item interactions and typically achieves an accuracy of around 93.1% on standard recommendation datasets. VAEs introduce stochasticity in the latent space to generalize better and handle noisy, sparse datasets, with reported accuracy levels of 92.8%. Matrix factorization models augmented with side information incorporate user and item metadata to enhance predictions, filling gaps left by sparse data, and can reach an accuracy of 93%. GNNRec models user-item interactions as graphs, effectively capturing higher-order dependencies, and showing superior performance with an accuracy of 94.3%, especially in sparse data environments. Hybrid approaches combine collaborative filtering with content-based or context-aware methods, enriching user-item matrices with additional information such as text reviews, and can achieve accuracy levels of around 95.2%. These advanced models significantly address data sparsity and noise challenges, offering valuable insights that could further enhance the performance of the PAMR system.

This approach not only adheres to the comparative standards typically expected in contemporary computational research but also highlights the specific advantages of PAMR in handling large, complex datasets typical of modern e-commerce platforms. These details underscore the advancements our model introduces to the domain of personalized product recommendations.

### *4.7 Discussion*

The research presents a comparative analysis of different models, focusing on the PAMR (Personalized Adaptive Multi-Product Recommendation System) across Amazon, Health Product, and Flipkart Health datasets. AlexNet-Bi-GRU excels on the Amazon dataset with 96.9% accuracy and a 96.97% F1-score. GoogleNet-Bi-GRU leads on the Health Product dataset, achieving 96.25% accuracy and a 96.22% F1-score, while ResNet-50-Bi-GRU performs best on the Flipkart Health dataset with 95.79% accuracy and a 95.77% F1-score. GoogleNet-Bi-GRU shows the highest overall accuracy (96.25%), followed by AlexNet-Bi-GRU (95%) and ResNet-50-Bi-GRU (94.6%). These PAMR models, particularly those using Bi-GRU units, outperform traditional LSTM-based networks, highlighting the effectiveness of Bi-GRU for recommendation systems. Our research enhances the theoretical foundation of machine learning in e-commerce by demonstrating the effectiveness of integrating transfer learning with Bi-GRU architectures. The PAMR model dynamically adapts to user preferences and diverse products, outperforming traditional models that often miss temporal and complex user behavior patterns. Incorporating an Attention Update Gate-GRU (AUG-GRU) with an attention mechanism refines our model's focus on relevant features, crucial for personalized recommendations. This extends current deep learning theories, emphasizing context-aware, adaptable models. Practically, the PAMR model significantly improves accuracy and adaptability in e-commerce recommendation systems, achieving up to 96.9% accuracy and a high F1-score on datasets from Amazon and Flipkart. Its robustness across product categories and capacity to handle large e-commerce platforms aim to enhance its recommendation engines and boost customer satisfaction. Unlike many current systems that rely on filtering methods, our approach utilizes a sophisticated machine learning architecture that learns from vast and varied data inputs. The practical success of the PAMR model in surpassing traditional LSTM-based models and its capability to outperform other contemporary algorithms also underscore its novelty. This is not only a technical improvement but also a methodological shift that could influence future designs of recommendation systems.

The distinct performances of the model variants (AlexNet-Bi-GRU, GoogleNet-Bi-GRU, and ResNet-50-Bi-GRU) on different datasets can be attributed to the unique strengths of each model architecture in handling the characteristics of the respective datasets. AlexNet, with its relatively simpler architecture, excels at capturing essential visual features, making it highly effective for the Amazon dataset, which includes a vast and varied product catalog, achieving high accuracy and F1-scores. On the other hand, GoogleNet, known for its inception modules and ability to capture multi-scale features, performs best on the Health Product dataset, which likely requires more precision in identifying intricate product attributes. ResNet-50, with its deep residual learning capabilities, is particularly effective at handling the complex patterns in the Flipkart Health dataset, offering superior accuracy and balanced performance. These performances are also influenced by the nature of the product categories, data sparsity, and the inherent complexity of each dataset. Furthermore, the integration of Bi-GRU across all variants enables the models to capture both spatial and sequential dependencies, contributing to their overall effectiveness. The diverse strengths of each model architecture are reflected in their optimal performance on specific datasets, highlighting the importance of matching model architecture to dataset characteristics for improved recommendation accuracy.

## 5 Conclusions

In conclusion, the study introduces the PAMR model, a combination method that combines transfer learning and Bidirectional Gated Recurrent Units (Bi-GRU). Its goal is to improve personalized and flexible multi-product recommendation systems in the rapidly growing e-commerce market. The PAMR model has great performance metrics, as shown by its accuracy of 96.9%, precision of 96.2%, and impressive F1-score of 96.97% on the Amazon dataset. These metrics were measured across a wide range of datasets from well-known platforms like Amazon and Flipkart. Additionally, GoogleNet-Bi-GRU and ResNet-50-Bi-GRU work very well on the Health Product and Flipkart datasets, respectively, which makes the proposed model even more flexible and accurate. As a future research endeavor, we could focus on further enhancing the PAMR model's capabilities and addressing existing challenges. There is a need to delve deeper into enhancing personalization within recommendation systems by integrating advanced techniques such as user behavior analysis, context-aware recommendations, and robust user feedback mechanisms.

**Acknowledgement:** The authors are extremely grateful to the editors and reviewers for their meticulous evaluations and sage advice.

**Funding Statement:** The authors received no specific funding for this study.

**Author Contributions:** The authors confirm contribution to the paper as follows: Study conception and design: Buchi Reddy Ramakantha Reddy; Data collection: Ramasamy Lokesh Kumar; Analysis and interpretation of results: Buchi Reddy Ramakantha Reddy; Draft Manuscript preparation: Buchi Reddy Ramakantha Reddy; Supervision: Ramasamy Lokesh Kumar. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** Upon reasonable request, the datasets generated and/or analyzed during the present project may be provided by the corresponding author.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

## References

[1] R. Liang and J. Wang, "A linguistic intuitionistic cloud decision support model with sentiment analysis for product selection in e-commerce," *Int. J. Fuzzy Syst.*, vol. 21, no. 3, pp. 963–977, Jan. 2019. doi: 10.1007/s40815-019-00606-0.

[2] Q. Sun, J. Niu, Z. Yao, and H. Yan, "Exploring eWOM in online customer reviews: Sentiment analysis at a fine-grained level," *Eng. Appl. Artif. Intell.*, vol. 81, no. 2, pp. 68–78, Feb. 2019. doi: 10.1016/j.engappai.2019.02.004.

[3] B. Ozyurt and M. A. Akcayol, "A new topic modeling based approach for aspect extraction in aspect-based sentiment analysis: SS-LDA," *Expert Syst. Appl.*, vol. 168, Jan. 2021, Art. no. 114231. doi: 10.1016/j.eswa.2020.114231.

[4] A. Iftikhar, M. A. Ghazanfar, M. Ayub, Z. Mehmood, and M. Maqsood, "An improved product recommendation method for collaborative filtering," *IEEE Access*, vol. 8, pp. 123841–123857, Jun. 2020. doi: 10.1109/ACCESS.2020.3005953.

[5] S. G. K. Patro *et al.*, "A hybrid action-related K-nearest neighbour (HAR-KNN) approach for recommendation systems," *IEEE Access*, vol. 8, pp. 90978–90991, May 2020. doi: 10.1109/ACCESS.2020.2994056.

[6] M. Shaheen, S. M. Awan, N. Hussain, and Z. A. Gondal, "Sentiment analysis on mobile phone reviews using supervised learning techniques," *Int. J. Mod. Educ. Comput. Sci.*, vol. 11, no. 7, pp. 32–43, Jul. 2019. doi: 10.5815/ijmecs.2019.07.04.

[7] T. U. Haque, N. N. Saber, and F. M. Shah, "Sentiment analysis on large scale Amazon product reviews," in *Proc. IEEE Int. Conf. Innov. Res. Dev. (ICIRD)*, Bangkok, Thailand, May 11–12, 2018, pp. 1–6.

[8] S. Wassan, X. Chen, T. Shen, M. Waqar, and N. Z. Jhanjhi, "Amazon product sentiment analysis using machine learning techniques," *Rev. Argent. Clín. Psicol.*, vol. 30, no. 1, Mar. 2021, Art. no. 965. doi: 10.24205/03276716.2020.2065.

[9] F. Zhang, X. Hao, J. Chao, and S. Yuan, "Label propagation-based approach for detecting review spammer groups on e-commerce websites," *Knowl.-Based Syst.*, vol. 193, no. 2, Mar. 2020, Art. no. 105520. doi: 10.1016/j.knosys.2020.105520.

[10] A. S. Ghabayen and B. H. Ahmed, "Polarity analysis of customer reviews based on part-of-speech subcategory," *J. Intell. Syst.*, vol. 29, no. 1, pp. 1535–1544, Jan. 2020. doi: 10.1515/jisys-2018-0356.

[11] F. Xu, Z. Pan, and R. Xia, "E-commerce product review sentiment classification based on a naïve Bayes continuous learning framework," *Inf. Process. Manag.*, vol. 57, no. 5, Sep. 2020, Art. no. 102221. doi: 10.1016/j.ipm.2020.102221.

[12] X. Zhang, H. Luo, B. Chen, and G. Guo, "Multi-view visual Bayesian personalized ranking for restaurant recommendation," *Appl. Intell.*, vol. 50, no. 9, pp. 2901–2915, Sep. 2020. doi: 10.1007/s10489-020-01703-6.

[13] X. Zheng, J. Li, and Q. Wu, "A light recommendation algorithm of We-Media articles based on content," *Int. J. Digit. Crime Forensics*, vol. 12, no. 4, pp. 68–71, Oct. 2020. doi: 10.4018/IJDCF.2020100106.

[14] M. Wang, "Applying Internet information technology combined with deep learning to tourism collaborative recommendation system," *PLoS One*, vol. 15, no. 12, Dec. 2020, Art. no. e0240656. doi: 10.1371/journal.pone.0240656.

[15] J. Zhang, F. Gu, Y. J. Ji, and J. F. Guo, "Personalized scientific and technological literature resources recommendation based on deep learning," *J. Intell. Fuzzy Syst.*, vol. 41, no. 2, pp. 2981–2996, Jan. 2021. doi: 10.3233/JIFS-210043.

[16] J. Wang and Y. Zhang, "Using cloud computing platform of 6G IoT in e-commerce personalized recommendation," *Int. J. Syst. Assur. Eng. Manag.*, vol. 12, no. 4, pp. 654–666, Aug. 2021. doi: 10.1007/s13198-021-01059-1.

[17] K. Damak, O. Nasraoui, and W. S. Sanders, "Sequence-based explainable hybrid song recommendation," *Front. Big Data*, vol. 4, Jul. 2021, Art. no. 693494. doi: 10.3389/fdata.2021.693494.

[18] J. Q. Zhang, D. J. Wang, and D. J. Yu, "TLSAN: Time-aware long- and short-term attention network for next-item recommendation," *Neurocomputing*, vol. 441, no. 1, pp. 179–191, Jan. 2021. doi: 10.1016/j.neucom.2021.02.015.

[19] Z. Liu, S. Guo, L. Wang, B. Du, and S. Pang, "A multi-objective service composition recommendation method for individualized customer: Hybrid MPA-GSO-DNN model," *Comput. Ind. Eng.*, vol. 128, no. 2, pp. 122–134, Dec. 2019. doi: 10.1016/j.cie.2018.12.042.

[20] J. -W. Chang *et al.*, "Music recommender using deep embedding-based features and behavior-based reinforcement learning," *Multimed. Tools Appl.*, vol. 80, no. 26–27, pp. 34037–34064, Nov. 2021. doi: 10.1007/s11042-019-08356-9.

[21] J. Sun, M. Zhu, Y. Jiang, Y. Liu, and L. Wu, "Hierarchical attention model for personalized tag recommendation," *J. Assoc. Inf. Sci. Technol.*, vol. 72, no. 2, pp. 173–189, Feb. 2021. doi: 10.1002/asi.24400.

[22] L. Yang *et al.*, "A LSTM based model for personalized context-aware citation recommendation," *IEEE Access*, vol. 6, pp. 59618–59627, Nov. 2018. doi: 10.1109/ACCESS.2018.2872730.

[23] Q. Li, X. Li, B. Lee, and J. Kim, "A hybrid CNN-based review helpfulness filtering model for improving e-commerce recommendation service," *Appl. Sci.*, vol. 11, no. 18, Sep. 2021, Art. no. 8613. doi: 10.3390/app11188613.

[24] H. T. H. Nguyen, M. Wistuba, J. Grabocka, L. R. Drumond, and L. Schmidt-Thieme, "Personalized deep learning for tag recommendation," in *Proc. Pacific-Asia Conf. Knowl. Discov. Data Min.*, Cham, Germany, Apr. 2017, pp. 186–197.

[25] X. He, L. Liao, H. Zhang, L. Nie, X. Hu and T. S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, Perth, Australia, Apr. 2017, pp. 173–182.

[26] Y. Feng *et al.*, "Deep session interest network for click-through rate prediction," in *ICAI'19: Proc. 28th Int. Joint Conf. Artif. Intell.*, Macao, China, Aug. 10–16, 2019, pp. 2301–2307.

[27] J. Yin, Y. Li, Z. Liu, J. Xu, B. Xia and Q. Li, "ADPR: An attention-based deep learning point of interest recommendation framework," in *Proc. 2019 Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary, Jul. 2019, pp. 1–8.

[28] H. Zhang, S. Wei, X. Hu, Y. Li, and J. Xu, "On accurate POI recommendation via transfer learning," *Distrib. Parallel Databases*, vol. 38, no. 3, pp. 585–599, Sep. 2020. doi: 10.1007/s10619-020-07299-7.

[29] L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment analysis for e-commerce product reviews in Chinese based on sentiment lexicon and deep learning," *IEEE Access*, vol. 8, pp. 23522–23530, Jan. 2020. doi: 10.1109/ACCESS.2020.2969854.

[30] A. Onan, "Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks," *Concurrency Comput. Pract. Exp.*, vol. 33, no. 23, Dec. 2021, Art. no. e5909. doi: 10.1002/cpe.5909.

[31] A. Alsayat, "Improving sentiment analysis for social media applications using an ensemble deep learning language model," *Arab J. Sci. Eng.*, vol. 47, no. 2, pp. 2499–2511, Feb. 2022. doi: 10.1007/s13369-021-06227-w.

[32] A. Suresh and M. J. C. M. Belinda, "Online product recommendation system using gated recurrent unit with Broyden Fletcher Goldfarb Shanno algorithm," *Evol. Intell.*, vol. 15, no. 3, pp. 1861–1874, Sep. 2022. doi: 10.1007/s12065-021-00594-x.

[33] J. Shobana and M. Murali, "An efficient sentiment analysis methodology based on long short-term memory networks," *Complex Intell. Syst.*, vol. 7, no. 5, pp. 2485–2501, Oct. 2021. doi: 10.1007/s40747-021-00436-4.

[34] Y. Zhang, Z. Liu, and C. Sang, "Unifying paragraph embeddings and neural collaborative filtering for hybrid recommendation," *Appl. Soft Comput.*, vol. 106, no. 6, Aug. 2021, Art. no. 107345. doi: 10.1016/j.asoc.2021.107345.

[35] T. Mandhula, S. Pabboju, and N. Gugulotu, "Predicting the customer's opinion on Amazon products using selective memory architecture-based convolutional neural network," *J. Supercomput.*, vol. 76, no. 8, pp. 5923–5947, Aug. 2020. doi: 10.1007/s11227-019-03081-4.

[36] P. Hajek, A. Barushka, and M. Munk, "Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining," *Neural Comput.*, vol. 32, no. 23, pp. 17259–17274, Dec. 2020. doi: 10.1007/s00521-020-04757-2.

[37] M. Umer *et al.*, "Impact of convolutional neural network and FastText embedding on text classification," *Multimed. Tools Appl.*, vol. 82, no. 4, pp. 5569–5585, Feb. 2023. doi: 10.1007/s11042-022-13459-x.

[38] S. Sachin, A. Tripathi, N. Mahajan, S. Aggarwal, and P. Nagrath, "Sentiment analysis using gated recurrent neural networks," *SN Comput. Sci.*, vol. 1, no. 2, Apr. 2020, Art. no. 74. doi: 10.1007/s42979-020-0076-y.

[39] N. Ghasemi and S. Momtazi, "Neural text similarity of user reviews for improving collaborative filtering recommender systems," *Electron. Commer. Res. Appl.*, vol. 45, Feb. 2021, Art. no. 101019. doi: 10.1016/j.elerap.2020.101019.

[40] A. Onan, "Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 5, pp. 2098–2117, Mar. 2022. doi: 10.1016/j.jksuci.2022.02.025.

[41] A. Onan and M. A. Toçoglu, "A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification," *IEEE Access*, vol. 9, no. 8, pp. 7701–7722, Jan. 2021. doi: 10.1109/ACCESS.2021.3049734.

[42] K. Ohtomo, R. Harakawa, M. Iisaka, and M. Iwahashi, "AM-Bi-LSTM: Adaptive multi-modal Bi-LSTM for sequential recommendation," *IEEE Access*, vol. 12, pp. 12720–12733, Feb. 2024. doi: 10.1109/ACCESS.2024.3355548.

[43] Y. Hao, T. Zhang, P. Zhao, Y. Liu, V. S. Sheng and J. Xu, "Feature-level deeper self-attention network with contrastive learning for sequential recommendation," *IEEE Trans. Knowl. Data Eng*, vol. 35, no. 10, pp. 10112–10124, Oct. 2023. doi: 10.1109/TKDE.2023.3250463.

[44] F. Zeng, R. Tang, and Y. Wang, "User personalized recommendation algorithm based on GRU network model in social networks," *Mob. Inf. Syst.*, vol. 2022, no. 12, pp. 1–8, Jan. 2022, Art. no. 1487586. doi: 10.1155/2022/1487586.

[45] H. Guo, Z. Guo, and Z. Liu, "Personalized recommendation model based on improved GRU network in big data environment," *J. Electr. Comput. Eng.*, vol. 2023, no. 4, pp. 1–9, Aug. 2023, Art. no. 3162220. doi: 10.1155/2023/3162220.

[46] X. Ma, H. Zhang, J. Zeng, Y. Duan, and X. Wen, "FedKGRec: Privacy-preserving federated knowledge graph aware recommender system," *Appl. Intell.*, vol. 54, no. 19, pp. 9028–9044, Jan. 2024. doi: 10.1007/s10489-024-05634-4.

[47] X. Shen and Y. Zhang, "A knowledge graph recommendation approach incorporating contrastive and relationship learning," *IEEE Access*, vol. 11, pp. 99628–99637, Sep. 2023. doi: 10.1109/ACCESS.2023.3310816.

[48] X. Ma, J. Tan, L. Zhu, X. Yan, and X. Kong, "GSRec: A graph-sequence recommendation system based on reverse-order graph and user embedding," *Mathematics*, vol. 12, no. 1, Jan. 2024, Art. no. 164. doi: 10.3390/math12010164.

[49] Y. Xiao, C. Li, and V. Liu, "DFM-GCN: A multi-task learning recommendation based on a deep graph neural network," *Mathematics*, vol. 10, no. 5, Mar. 2022, Art. no. 721. doi: 10.3390/math10050721.

[50] J. Bobadilla, F. Ortega, A. Gutiérrez, and Á. González-Prieto, "Deep variational models for collaborative filtering-based recommender systems," *Neural Comput. Appl.*, vol. 35, no. 10, pp. 7817–7831, Feb. 2023. doi: 10.1007/s00521-022-08088-2.

[51] X. Song, J. Qin, and Q. Ren, "A recommendation algorithm combining local and global interest features," *Electronics*, vol. 12, no. 8, 2023, Art. no. 1857. doi: 10.3390/electronics12081857.

[52] R. Nahta, G. S. Chauhan, Y. K. Meena, and D. Gopalani, "CF-MGAN: Collaborative filtering with metadata-aware generative adversarial networks for top-N recommendation," *Inf. Sci.*, vol. 689, no. 2, Jan. 2025, Art. no. 121337. doi: 10.1016/j.ins.2024.121337.

[53] K. Xu *et al.*, "Intelligent classification and personalized recommendation of e-commerce products based on machine learning," Mar. 2024, *arXiv:2403.19345*.

[54] R. Thomas and J. R. Jeba, "A novel framework for an intelligent deep learning based product recommendation system using sentiment analysis (SA)," *Automatika*, vol. 65, no. 2, pp. 410–424, Apr. 2024. doi: 10.1080/00051144.2023.2295148.

[55] T. -L. Ho, A. -C. Le, and D. -H. Vu, "Multiview fusion using transformer model for recommender systems: Integrating the utility matrix and textual sources," *Appl. Sci.*, vol. 13, no. 10, May 2023, Art. no. 6324. doi: 10.3390/app13106324.

[56] "Amazon Products Dataset," 2023. Accessed: Apr. 03, 2024. [Online]. Available: https://www.kaggle.com/datasets/asaniczka/amazon-canada-products-2023-2-1m-products

[57] B. R. Reddy and R. L. Kumar, "An e-commerce based personalized health product recommendation system using CNN-Bi-LSTM model," *Int. J. Intell. Eng. Syst.*, vol. 16, no. 6, pp. 398–410, Dec. 2023. doi: 10.22266/ijies2023.1231.33.

[58] Z. Wang *et al.*, "An interactive personalized garment design recommendation system using intelligent techniques," *Appl. Sci.*, vol. 12, no. 9, May 2022, Art. no. 4654. doi: 10.3390/app12094654.

[59] L. Zhou, "Product advertising recommendation in e-commerce based on deep learning and distributed expression," *J. Electron. Commer. Res.*, vol. 20, no. 2, pp. 321–342, Jun. 2020. doi: 10.1007/s10660-020-09411-6.

[60] X. Wang, X. He, M. Wang, F. Feng, and T. -S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, Paris, France, Jul. 21–25, 2019, pp. 165–174. doi: 10.1145/3331184.3331267.

[61] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. 2018 World Wide Web Conf. (WWW)*, Lyon, France, Apr. 23–27, 2018, pp. 689–698. doi: 10.1145/3178876.3186150.

[62] F. Sun *et al.*, "Sequential recommendation with bidirectional encoder representations from transformers," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Beijing, China, Nov. 3–7, 2019, pp. 1441–1450. doi: 10.1145/3357384.3357895.