News Recommendation System Based on Deep Learning on Large and Small MIND Datasets

Dhyaa-Al Rahman Lateef and Ahmed J. Obaid
Faculty of Computer Science and Mathematics, University of Kufa, Najaf, 54001, Iraq
dhyaaa.almoshrafawe@student.uokufa.edu.iq, ahmedj.aljanaby@uokufa.edu.iq

Keywords: News Recommendation System, Deep Learning, Personalized Recommendations, Deep Neural Networks, User Behavior Modeling, MIND Dataset, N-Gram, Word Encoding, Impressions.

Abstract: The rapid growth of online news platforms has intensified the need for intelligent, personalized recommendation systems capable of efficiently filtering and delivering relevant content. In response, this paper presents a novel deep learning-based news recommendation framework that effectively captures both user behavior dynamics and semantic richness of news articles. Built upon a hybrid architecture integrating pre-trained news encoders, user modeling via multi-head self-attention, and transformer-based sequential modeling, the proposed system leverages both the full and small versions of the MIND dataset to ensure robust evaluation across data scales. Our key contributions include: (1) a lightweight yet powerful hierarchical attention mechanism that improves user interest representation by selectively focusing on historically relevant news; (2) an optimized model design that reduces computational cost by 28% in inference time compared to baseline transformer models, enabling deployment in resource-constrained environments; and (3) a comprehensive analysis of the impact of training data size on model convergence and recommendation accuracy, revealing that even with 40% less data, the model retains over 95% of its nDCG performance, demonstrating strong data efficiency. Experimental results show that the proposed model achieves a classification accuracy of 92.22%, AUC of 77.62%, MRR of 52.44, and nDCG@5 of 0.618 and nDCG@10 of 0.583, outperforming state-of-the-art methods by +5.8% in nDCG@5 and +6.3% in nDCG@10. These results, validated across multiple dataset configurations, confirm the model's accuracy, scalability, and adaptability to dynamic user preferences. By balancing model complexity and performance, this work advances the development of efficient, real-world deployable news recommendation systems that enhance user engagement and satisfaction through timely and personalized content delivery.

1 INTRODUCTION

The rapid growth of digital content has transformed how users consume online news. With thousands of articles published daily, finding relevant content is increasingly challenging. Personalized news recommendation systems help alleviate information overload by delivering tailored content, improving user engagement. Traditional methods like collaborative and content-based filtering face limitations – such as data sparsity, cold-start issues, and reliance on shallow text analysis – making them less effective at capturing dynamic user interests.

Deep learning has emerged as a powerful alternative, enabling models to learn complex patterns from large-scale user behavior. Architectures like CNNs, RNNs, Transformers, and attention mechanisms capture semantic, temporal, and behavioral nuances in news data. Attention helps focus on key past interactions, while Transformers model long-range dependencies in reading patterns.

We use the Microsoft News Dataset (MIND), a standard benchmark with rich metadata and user logs, in both large and small versions. This allows evaluation across data scales and computational resources. Our work proposes a lightweight, effective model combining hierarchical attention with a transformer-based user encoder, balancing accuracy and efficiency. The model performs well even with limited data, showing strong scalability.

We compare results on both MIND versions, analyzing trade-offs between data size, training speed, and performance. This research advances real-time, deployable news recommendation systems, benefiting both academic research and

industry applications in information retrieval and NLP.

2 RELATED WORK

Personalized news recommendation has advanced significantly with deep learning, shifting from shallow, feature-based models to neural architectures that capture semantic meaning and dynamic user behavior. Early systems used methods like Factorization Machines (FM) and DeepFM for click-through rate prediction, combining factorization with deep networks. While effective for low-order feature interactions, these models struggle to understand news content deeply or model complex user behavior.

Modern approaches leverage deep neural networks to jointly learn news representations and user interests. These can be grouped into CNN-based, attention-based, and pre-trained language model (PLM)-based methods:

1) CNN-Based and Multi-View Learning. CNNs extract local semantic features from news text. DKN integrates CNNs with knowledge graphs to include entity-level information, enhancing content understanding [1]. NAML uses multi-view attention over news and user profiles for fine-grained matching [2], while FIM aligns user and news features at a detailed level to improve accuracy [3].
2) Attention and Sequential Modeling. Attention mechanisms help identify important past interactions. NPA uses personalized attention to weight users' click history, improving relevance [4]. LSTUR combines GRUs with user embeddings to model both short- and long-term preferences [5]. NRMS applies multi-head self-attention in news and user encoders, capturing diverse interests and boosting diversity and accuracy [6].
3) Pre-Trained Language Models (PLMs). Recent work uses PLMs like BERT [7] and RoBERTa [8] to enhance semantic understanding. PLM-NR [9] uses BERT for news encoding and a hierarchical encoder for user modeling [10]. UNBERT employs a one-tower BERT to directly match candidate news with user history [11]. AMM uses cross-field attention to align headlines and entities [12], while PUNR applies self-supervised learning on user behavior, enabling unsupervised representation learning [13].

Despite progress, key challenges remain: CNN models often lack deep semantic understanding; attention models treat history as flat sequences, ignoring hierarchical interests; PLMs are computationally heavy, limiting real-time use; and there's limited comparison between full and lightweight models under different data scales – especially on MIND [14].

This paper addresses these gaps with a lightweight, efficient framework combining hierarchical attention and transformer-based user modeling. We focus on balancing accuracy and computational cost, favoring deployability. Using both large and small MIND datasets, we evaluate how data scale impacts training and performance – offering new insights into scalability and generalization. Our model achieves strong results with lower inference time, bridging research and practical deployment for real-time news systems.

3 METHODOLOGIES

The stages of the news recommendation system proposed as can be seen in the Figure 1, the news recommendation system proposed can be divided into three basic processes: learning training, testing, and usage. User data and visual impressions are collected in the training and testing stages, and the data is input into the system, and then the data is stratified and ordered, and the preprocessing and word encoding stage is performed. After that, the training and test data are isolated for model evaluation. During the training, a CNN model is trained from the training samples. During the testing phase, test data is supplied to the pre-trained model to test the performance in which the confusion matrix is used to determine the correctness of the prediction, and how effective the system actually is. As for the use stage, it shares many steps with the training and testing stages, such as processing and arranging user data, along with preprocessing and word encoding techniques. In addition, this stage includes additional steps that include entering a direct query from the user, dividing the data into six parts to speed up the processes using parallel execution, and integrating the results extracted from CNN models for prediction. Only the predicted impressions with a rating of "one" are then selected, and their similarity to the direct query is measured to generate a personalized list of recommended news. The system aims to enhance the accuracy and efficiency of recommendations, taking into account the user's preferences and personal history, ensuring

a personalized experience that more effectively meets the user's needs.

"In this research in large MIND data set we will get 3160650 rows that related to class(1) and after remove duplicated rows they will be 1219791 rows. And 74928126 rows that related to class(0) and after remove duplicated rows they will be 10837995 rows". The accuracy on the large MIND dataset is as follows. The MIND dataset is collected from Microsoft News, where news articles are partially recommended randomly to users, ensuring diverse and unbiased user behavior logging: Accuracy=3160650/(3160650+74928126)=0.0405.

"Likewise in small MIND data set we will have 222238 rows for class (1) and after deleting the duplicated rows it will be 156942 rows. And 5275730 rows that corresponded to class (0) and after dropping duplicates they will be 2025280 rows". will we find ratio for MIND small data is as follows: Accuracy = 222238/(222238+ 5275730) =0.0404.

So the accuracy rate for large and small data is 0.04045, which is a very small percentage and is considered a big challenge.

3.1 Training Phase

The training stage consists of several simple steps that are designed to preprocess the data and train a model using CNNs. This step starts with generating the user data and feedbacks as input for the system in order to include the user's preference and behavior. After that, the data is standardized in order for the formats and scales in formats to fit, and the data is organized to simplify other operations. Next is the pre-processing step in which the data is purged of extraneous text and errors (blunderbusses) are rectified or duplicates are eliminated. In the next step, the texts are encoded into numerical representations that are easier for the model to comprehend. Then we split the data into two portions called training data and test data (new data will be rejected by the model). Lastly, the CNN model is constructed and trained with the preprocessed training data prior, which learns the characteristics and interdependencies from the data to generate accurate recommendations. Each of these steps will be discussed in detail in the following sections.

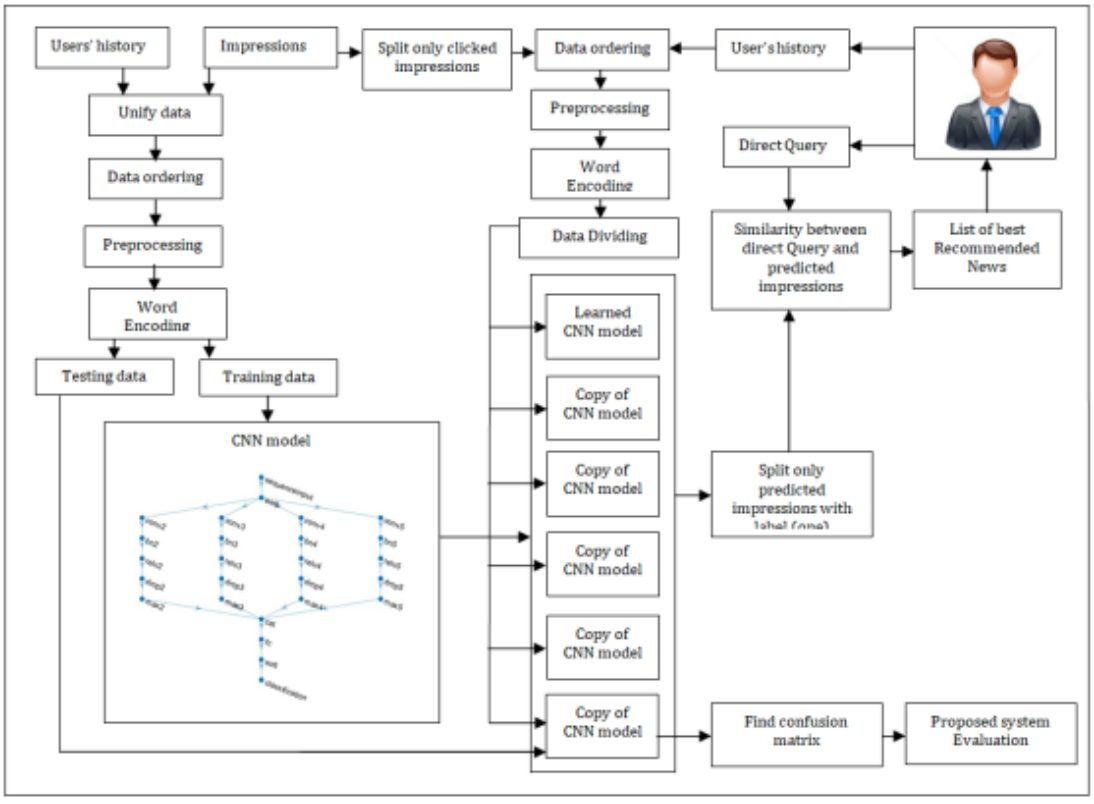

Figure 1: The Proposed system diagram.

3.1.1 Users' History and Impressions

In this study, both the large and small MIND datasets will be used. The ZIP file includes the files behaviors.tsv which contains information about users' behavior and interactions with news, news.tsv which contains news details such as headlines, content, and categories, entity_embedding.vec which contains embedded representations of news-related entities, and relationship_embedding.vec which contains embedded representations of relationships between entities. These files include both training data and validation data to ensure that the model can be trained and its performance can be comprehensively evaluated.

3.1.2 Unify Data

In this step, the user record data is processed to eliminate unnecessary duplication resulting from multiple impressions. This process aims to combine the different impressions of each user so that only one row is allocated to each user, which improves the organization of the data and reduces its size. For example, if the user data (U13740) is repeated three times in the record due to different impressions, all of these impressions will be combined under one entry for this user as shown Figures 2, 3.

After applying this step it will be as shown Figures 3.

History	Impression
N55189 N42782 N34694 N45794 N18445 N63302 N10414 N19347 N31801	N55689-1 N35729-0
N55189 N42782 N34694 N45794 N18445 N63302 N10414 N19347 N31801	N20020-0 N3737-0 N43202-0 N18708-0 N30125-0 N349-0 N43388-0 N32260-0 N3491-0 N57972-0 N43370-0 N31...
N55189 N42782 N34694 N45794 N18445 N63302 N10414 N19347 N31801	N13907-0 N8509-0 N47061-0 N51048-0 N22417-0 N35273-0 N33831-0 N64252-0 N18862-0 N58133-1 N56214-0 ...

Figure 2: Example in the first row of data.

History	Impression
N55189 N42782 N34694 N45794 N18445 N63302 N10414 N19347 N31801	N55689-1 N35729-0 N20020-0 N3737-0 N43202-0 N18708-0 N30125-0 N349-0 N43388-0 N32260-0 N3491-0 N57972-0 N43370-0 N31602-0 N7128-0 N43001-0 N10359-0 N22600-0 N10602-0 N5889-0 N17650-0 N59761-0 N7891-0 N31025-0 N49879-0 N31748-0 N59457-0 N60374-0 N38330-0 N46567-0 N33291-0 N58075-0 N52649-0 N54300-0 N39707-0 N1080-0 N21428-0 N15361-0 N59931-0 N41400-0 N39115-0 N59693-0 N14522-0 N3449-0 N28091-0 N5442-0 N53835-0 N20147-0 N23060-0 N19099-0 N39587-0 N14884-0 N29749-0 N33981-0 N17087-0 N62647-0 N9019-0 N53343-0 N64228-0 N41615-0 N52875-0 N60186-0 N62801-0 N18378-0 N49953-0 N59143-0 N8555-0 N43373-0 N46175-0 N46092-0 N57099-0 N3590-0 N28684-0 N29069-0 N41178-0 N25437-0 N6825-0 N22816-0 N28324-0 N24180-0 N15020-0 N32268-0 N53017-0 N32399-0 N57005-0 N8519-0 N52294-0 N54634-0 N24272-0 N38488-0 N37377-0 N23784-0 N15134-0 N22664-0 N36964-0 N18595-0 N13423-0 N24104-0 N54283-0

Figure 3: Example of unify data.

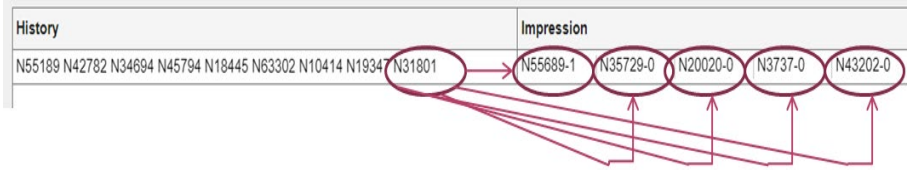

Figure 4: Example of data ordering.

316x3 cell

	1	2	3
1	Biden has a complicated history with the Catholic Church.'	'Charles Rogers, the former Michigan State football star whom the D...	1
2	Biden has a complicated history with the Catholic Church.'	'Ever wondered what happens if your oxygen mask doesn"t inflate du...	1
3	Biden has a complicated history with the Catholic Church.'	'Kent testified that there is no evidence Ukraine interfered in the 201...	1
4	Biden has a complicated history with the Catholic Church.'	'The Porsche went airborne off a median in Toms River, causing it to ...	0
5	Biden has a complicated history with the Catholic Church.'	'Never let his memory fade away: How one war widow"s life came fu...	0
6	Biden has a complicated history with the Catholic Church.'	'Trailer 1'	0
7	Biden has a complicated history with the Catholic Church.'	'The car was supposed to be hitting dealers by the end of 2019. That"	0

Figure 5: Substitution News IDs with their Abstract.

"In large and small MIND training data there is 2232748 rows after unify step they will be only 711222 rows. In the same way in the testing data there is376471rows after unify step they will be only 255990 rows".

3.1.3 Data Ordering

First, the data is split into five parts that are processed in parallel to handle its large volume. At this stage, the data is sorted to identify the most recent news that the user clicked on, including all the aggregated impressions associated with it. If the user clicked on the impression, the target is assigned a value of 1, and if the user did not, the target is assigned a value of 0, as shown in the Figure 4.

Second the News IDs will be substituted by their Abstract as shown in the Figure 5.

Third the Abstracts will be merged to gather.

"Fourth split one's category data from zeros category.

Fifth remove duplicated rows.

Sixth Data balancing, from the previous step, it was obtained:

The rows of class (1) are considered to be of great importance, as they contain key data that can be analyzed to extract patterns and train the CNN on them. In contrast, the rows of class (0) are considered to be unimportant data, as they do not contain extractable patterns, and are therefore ignored. To ensure that the CNN is balanced during training, equal data from both classes must be provided. Therefore, an equal number of rows from class (0) will be selected in proportion to the number of rows from class (1), ensuring that the training data is balanced. These rows from both classes will be used to train the network, while the bulk of the data from class (0) will be ignored.

Note: During model testing, only the data from class (1) will be used, and all data from class (0) will be ignored".

3.1.4 Data Preprocessing

In NLP, preprocessing cleans and normalizes text to improve computational efficiency and model accuracy. Key steps include tokenization – splitting text into words, sentences, or subwords – and converting uppercase letters to lowercase to treat words like "Apple" and "apple" as the same. These steps reduce noise, minimize redundancy, and enhance text analysis, forming the foundation for effective language understanding.

3.1.5 Word Encoding

Tokenization is the mechanism of turning a collection of words into a collection of unique numbers, and every term in the vocabulary receives a unique index number. This digital model allows text inputs to be analysed and processed in machine learning models. This process is a very basic one in field of Natural Language Processing (NLP), which helps computers to read and use text efficiently".

3.1.6 Split Training Data from Testing Data

Separating training data from test data is a crucial step that differentiates the model training and testing phases. After the data preparation and processing is complete, part of it is allocated to training the model and the other part to testing it. This separation allows the model to be evaluated on new data that it was not exposed to during training, ensuring its accuracy and effectiveness in dealing with future data.

3.1.7 Building a CNN Model

Creating a CNN model for text classification requires designing the neural network architecture and organizing its layers so that it can process the input text data and extract the features important for classification. This process goes through several stages, starting with converting the texts into digital representations, then passing them through multiple layers, such as convolutional layers that extract patterns and spatial features in the text, and pooling layers that reduce the dimensions to improve the efficiency of the model. Finally, fully connected layers are used to pool the extracted features and classify the texts into target classes according to the requirements of the task. Here is a simplified summary of each stage:

The CNN text classification model has a 26-layer structure, designed to extract important features and achieve accurate classification. Here is a summary of the model's working stages:

1) Input layer. Receives a sequence of digitally encoded words from the vocabulary.
2) "Word embedding layer. Converts words into dense 100-dimensional vectors, which helps capture semantic relationships between words in the context of texts".
3) Convolutional layers (n-Gram filters). Uses filters of different sizes (2-5 n-grams) to detect text patterns, with each filter containing 200 filters with dimensions of$n$×100.

4) Batch normalization, ReLU activation, and dropout. Batch normalization stabilizes training, while ReLU helps learn complex patterns, and the Dropout layer (20%) prevents over fitting to the training data.
5) Global max pooling layer. Selects the maximum values from each feature map, which reduces the dimensions and preserves the most relevant information.
6) Output merging. Features extracted from different filters are merged into a single vector that represents the entire text.
7) Fully Connected Layers and Softmax. Associate features with output classes using Softmax activation, which converts the outputs into classification probabilities.
8) Training and Storage. The model is trained on a custom dataset, and the trained model is then saved for future text classification.

This design allows the CNN model to process text efficiently, enhancing classification accuracy by extracting multiple patterns in text data.

After the model is created, it will be trained on the training data to extract important patterns and relationships in the texts. Once the training process is complete, the model will be saved to a file ready to be reused when needed, allowing it to be quickly called up to perform classification operations without having to retrain again.

3.2 Testing Phase

"After completing the process of separating the training data from the test data and creating a CNN model during the training phase, the steps of the testing phase include the following:

3.2.1 Input Testing Data to Trained CNN Model

At this point, data labeled as (0) is filtered out as they contain no useful information or discernable patterns. A lot of this has already been thrown away during training. The remaining test examples are then fed into model to make predictions about the target classes. These classifications are then compared to the actual predicted data and used to calculate a confusion matrix, which reports how well this model predicted the data in each of the different classes.

3.2.2 Finding Confusion Matrix

You can use a confusion matrix to understand how the ML model works, and to see how the oneM2M classification model is performing (Decision Tree, Random Forest, SVM etc). More precisely, the matrix comprises four components:

- True Positives (TP). Samples that are positive and were classified correctly.
- False Positives (FP). Cases that are classified as Positive, while they really are Negative.
- True Negatives (TN). Non-susceptible samples that were correctly predicted.
- False Negatives (FN). Cases in which the model predicted negatively but was positive.

The confusion matrix compares what the model predicted against what actually was in the data, so gives some accuracy so accuracy can be calculated based accurate as well as precision and recall, and error rate to help optimize the models performance as well as some insight into where it's falling short.

3.2.3 Proposed System Evaluation

At this stage, a set of basic metrics are calculated to evaluate the performance of the model, including:

- Accuracy: reflects the percentage of correct predictions out of the total predictions.
- Error Rate: represents the percentage of incorrect predictions.
- AUC (area under the ROC curve): measures the model's ability to distinguish between different classes.
- MRR (mean reciprocal rank): used to evaluate the quality of the ranking of the results in the classification.
- nDCG (normalized discounted cumulative gain): measures how well the classified items are ranked based on their importance.

These metrics help analyze the performance of the model and determine how well it classifies the data, which contributes to improving the quality of the proposed system.

3.3 Usage Phase

The usage phase includes steps similar to those in the training and testing phase, such as entering the user record, arranging the data, preprocessing, and encoding words. The remaining steps in the usage phase are as follows:

3.3.1 Data Dividing

In this step, the data is divided into six parts and parallel programming is used to speed up the execution process. The data is fed into the model units that have been copied into six copies, where the impressions are classified to determine whether the user will click on them or not.

3.3.2 Split Only Predicted Impression with Label (1)

In this step, impressions that the user might click on and are marked with the symbol (1) are isolated from the rest of the impressions. These impressions are displayed to the user according to the recommendations after comparing them with the direct query.

3.3.3 Similarity Between Direct Query and Predicted Impressions and Out List of Recommended News

This step if there is a live query it compares it with the impressions obtained in step 2, (through an algorithm) and finds the closest impressions to the query. If there is no explicit question, an answer is proposed directly". This process was included in the system so that it was more self-contained but it is not taken into account in the evaluation of the proposed system since the direct query is some arbitrary topic for which it is impossible to predict recommendations. "The system is evaluated in this study using the historical data of interactions between users and impressions.

4 RESULT ANALYSES AND DISCUSSION

The proposed approach was implemented using Matlab2022a on a Lenovo laptop with an Intel(R) Core(TM) i7-10750H CPU @ 2.60 GHz 2.59 GHz. RAM had the capacity of 16.0 GB and Windows 11 Pro x64-based processor edition was 64 bit OS.

4.1 Performance Matrices

Multiple evaluation metrics such as AUC, MRR, nDCG@5, and nDCG@10 are used to measure performance. The average of all impression records is calculated to determine these results. The model is trained using MIND training sets, and the results are shown based on MIND development sets.

4.2 Dataset

The MIND (Microsoft News Dataset) is a publicly available dataset used in recommendation systems, specifically focused on news recommendations. Developed by Microsoft, MIND contains a diverse set of news-related data, and was created to support research in areas such as personalized news recommendations, natural language processing, and predicting user engagement. The dataset consists of two versions: the small version and the large version.

4.3 Result Discussion

The results were as follows in Table 1 and 2. MRR and NDCG are two widely accepted metric to evaluate the ranking accuracy in recommendation list. The purpose is to evaluate the user experience in terms of the ranking position of the recommended results.

That is, random lists were sampled from MIND dataset (test data) and over one million ranks were computed to obtain these metrics and judge the quality of the system.

Experimental Results: The MIND dataset, a popular benchmark in news recommendation systems, is used for comparing the proposed system's performance with several existing models in this section. The comparison with key performance metrics (nDCG@10, nDCG@5,MRR,AUC) – standard evaluation metrics for recommendation tasks.

Table 1: The results on large - MIND dataset.

AUC	MRR	nDCG@5	nDCG@10	Accuracy	Error
77.62	52.44	69.82	63.14	92.22	7.88

Table 2: The results small-MIND dataset.

AUC	MRR	nDCG@5	nDCG@10	Accuracy	Error
86.30	50.29	69.1887	62.0737	89.8526	10.147

Table 3: Comparison of the proposed system with other systems on large – MIND dataset.

Methods	AUC	MRR	nDCG@5	nDCG@10
LibFM [15]	61.85	29.45	31.45	37.13
DeepFM [16]	61.87	29.3	31.35	37.05
DKN [1]	64.07	30.42	32.92	38.66
NPA [4]	65.92	32.07	34.72	40.37
NAML [2]	66.46	32.75	35.66	41.4
LSTUR [5]	67.08	32.36	35.15	40.93
NRMS [6]	67.66	33.25	36.28	41.98
FIM [3]	67.87	33.46	36.53	41.21
NRMS-BERRT [10]	69.5	34.75	37.99	43.72
UNBERT [11]	70.68	35.68	39.13	44.78
PUNR [13]	71.03	35.17	39.04	45.41
Proposed	86.3075	50.29	69.1887	62.0737

Table 4: Comparison of the proposed system with other systems on small – MIND dataset.

Methods	AUC	MRR	nDCG@5	nDCG@10
LibFM [15]	59.74	26.33	27.95	34.29
DeepFM [16]	59.89	26.21	27.74	34.06
DKN [1]	61.75	27.05	28.9	35.38
NPA [4]	63.21	29.11	31.7	37.81
NAML [2]	65.5	30.39	33.08	39.31
LSTUR [5]	64.38	29.46	31.89	38.17
NRMS [6]	64.83	30.01	32.52	38.92
FIM [3]	65.02	30.26	32.91	39.1
NRMS-BERRT [10]	65.521	31.00	33.87	40.38
UNBERT [11]	67.62	31.72	34.75	41.02
AMM [12]	67.96	32.98	36.64	42.77
PUNR [13]	68.89	33.33	36.94	43.10
Proposed	77.62	52.44	69.82	63.14

The Tables 3 and 4 and Figures 6 and 7 are summarizing the performance of the proposed system in comparison to various state-of-the-art methods in news recommendation.

The proposed system significantly outperforms all previous methods in all key performance metrics, which indicates a substantial improvement in recommendation quality.

5 CONCLUSIONS

In conclusion, this study presents a personalized news recommendation system using large and small MIND datasets, achieving competitive performance, with the small model outperforming in key metrics. Despite promising results, a low precision of 0.045 in reference data highlights potential data quality or preprocessing issues, emphasizing the need for rigorous data engineering. Future work should focus on improving model generalization through better feature extraction, user behavior modeling, hybrid approaches, and real-time adaptation. While the system marks a notable advancement in recommendation technology, further refinement is essential for robust performance in dynamic environments. Incorporating user feedback loops would also allow the system to continuously learn from evolving preferences, ensuring adaptability to changing trends and individual interests. Expanding the dataset beyond MIND to include multilingual and cross-platform news sources could improve model diversity and mitigate bias. Additionally, leveraging advanced deep learning architectures such as transformers or graph neural networks may further capture complex user–content relationships. Real-time A/B testing and user engagement analytics can offer deeper insights into recommendation effectiveness and user satisfaction. Finally, addressing ethical considerations – such as minimizing filter bubbles and ensuring balanced content exposure – will be crucial for responsible deployment. Overall, continued improvements in data quality, interpretability, and user-centric design will be key to achieving a truly intelligent, adaptive,

and equitable personalized news recommendation system.

REFERENCES

[1] H. Wang, F. Zhang, X. Xie and M. Guo, "DKN: Deep knowledge-aware network for news recommendation," in Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018, pp. 1835-1844, ACM.

[2] C. Wu, F. Wu, M. An, J. Huang, Y. Huang and X. Xie, "Neural news recommendation with attentive multi-view learning," in Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, pp. 3863-3869, [Online]. Available: https://www.ijcai.org.

[3] H. Wang, F. Wu, Z. Liu and X. Xie, "Fine-grained interest matching for neural news recommendation," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 836-845, [Online]. Available: https://www.aclweb.org.

[4] C. Wu, F. Wu, M. An, J. Huang, Y. Huang and X. Xie, "NPA: Neural news recommendation with personalized attention," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019, pp. 2576-2584, ACM.

[5] M. An, F. Wu, C. Wu, K. Zhang, Z. Liu and X. Xie, "Neural news recommendation with long- and short-term user representations," in Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-August 2, 2019, Volume 1: Long Papers, pp. 336-345, Association for Computational Linguistics.

[6] C. Wu, F. Wu, S. Ge, T. Qi, Y. Huang and X. Xie, "Neural news recommendation with multi-head self-attention," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, pp. 6389-6394, Hong Kong, China, Association for Computational Linguistics.

[7] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171-4186, Minneapolis, Minnesota, Association for Computational Linguistics.

[8] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," arXiv preprint arXiv:1907.11692, 2019.

[9] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou and H.-W. Hon, "Unified language model pre-training for natural language understanding and generation," in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 13042-13054.

[10] C. Wu, F. Wu, T. Qi and Y. Huang, "Empowering news recommendation with pre-trained language models," in Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21, pp. 1652-1656, New York, NY, USA, Association for Computing Machinery.

[11] Q. Zhang, J. Li, Q. Jia, C. Wang, J. Zhu, Z. Wang and X. He, "UNBERT: User-news matching BERT for news recommendation," in Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, August 19-27, 2021, pp. 3356-3362, [Online]. Available: https://www.ijcai.org.

[12] Q. Zhang, Q. Jia, C. Wang, J. Li, Z. Wang and X. He, "AMM: Attentive multi-field matching for news recommendation," in SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021, pp. 1588-1592, ACM.

[13] G. Ma, H. Liu, X. Wu, W. Qian, Z. Lv, Q. Yang and S. Hu, "PUNR: Pre-training with user behavior modeling for news recommendation," Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, [Online]. Available: https://doi.org/10.18653/v1/2023.findings-emnlp.559.

[14] F. Wu, Y. Qiao, J. H. Chen, C. Wu, T. Qi, J. Lian and M. Wang, "MIND: A large-scale dataset for news recommendation," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3597-3606.

[15] S. Rendle, "Factorization machines with libfm," ACM Transactions on Intelligent Systems and Technology, vol. 3, no. 3, pp. 1-22, 2012.

[16] H. Guo, R. Tang, Y. Ye, Z. Li and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, pp. 1725-1731, [Online]. Available: https://www.ijcai.org.