Enhancing Product Recommendation Accuracy Using Bipartite Link Prediction and Long Short-Term Memory in Retail Industry

[1]Ivan Michael Siregar, [2]Firlie Resti Rosdiana
[1,2]Department of Information Systems, Faculty of Computer Science,
Institut Teknologi Harapan Bangsa, Indonesia
Email: [1]ivan@ithb.ac.id, [2]firlierr30@gmail.com

Article Info

Article history:
Received Mar 3rd, 2025
Revised Apr 12th, 2025
Accepted May 26th, 2025

Keyword:
Bipartite Link Prediction
Enhancing Product
Long Short-Term Memory
Recurrent Neural Networks

ABSTRACT

As competition in the retail sector intensifies, the demand for accurate customer-product recommendation systems has grown. Traditional similarity-based approaches such as common neighbor, Jaccard, Adamic Adar, preferential attachment, and resource allocation have been widely adopted in many business applications. However, these methods often struggle with capturing complex purchasing behaviors, product heterogeneity, temporal demand variations, and scalability challenges. This study introduces a deep learning-based recommendation model that integrates bipartite link prediction networks with Long Short-Term Memory (LSTM) to improve predictive accuracy. The bipartite network represents customer-product interactions, while the LSTM model captures sequential purchasing patterns to forecast future transactions. Experimental evaluation on a real-world building materials retail dataset comprising 389,087 transactions demonstrates the effectiveness of the proposed approach, achieving a Precision of 0.8223, Recall of 0.8034, F1-score of 0.8128, NDCG of 0.8601, and overall prediction accuracy of 0.854. The results indicate that the proposed model significantly outperforms similarity-based techniques, offering a robust solution for enhancing recommendation performance in dynamic retail environments.

Corresponding Author:
Ivan Michael Siregar,
Department of Information Systems, Faculty of Computer Science,
Institut Teknologi Harapan Bangsa
Dipati Ukur Street, No.80-84, Dago, Coblong, Bandung City, West Java, Indonesia 40132.
Email: ivan@ithb.ac.id

DOI: http://dx.doi.org/10.24014/ijaidm.v8i2.36052

1. INTRODUCTION

The growing competition in the retail industry, particularly in the building materials sector, has led businesses to explore advanced technologies to maintain competitiveness, increase sales, and retain customers. Companies must now deal with dynamic market fluctuations, including changes in raw material prices, supplier proliferation, and shifting customer preferences. To respond effectively, many retailers are turning to Artificial Intelligence (AI)-driven recommender systems, which have shown promise in optimizing marketing strategies and enhancing customer engagement [1][3].

In recent years, recommender systems have evolved significantly with the adoption of deep learning models such as Recurrent Neural Networks (RNN) and Graph Neural Networks (GNN). These models have been proven to capture sequential and structural patterns in user behavior more effectively than traditional techniques. For instance, RNN-based models such as the Long Short-Term Memory (LSTM) network address limitations like the vanishing gradient problem, enabling better modeling of long-term user interactions [2][3]. In retail scenarios—especially in session-based or project-driven domains like building materials—such sequential learning techniques are essential to accurately predict future purchases.

Another promising development in recommendation systems is the application of graph-based learning models. Many recommendation problems can be naturally represented as bipartite graphs, where nodes represent users and items, and edges represent interactions such as purchases or ratings. Graph-based models, including link prediction techniques, are able to uncover latent relationships between entities and infer potential future interactions. For example, models that integrate node similarity measures (e.g., Common Neighbors, Jaccard Index) have shown effectiveness in small-scale datasets but struggle with sparse and large-scale transactional data [1][4]. This shortcoming has driven research toward integrating graph structures with deep learning models like GNN or LSTM to capture both structural and temporal dynamics of customer behavior [2][5].

However, most deep learning approaches require substantial labeled data and computing resources constraints that are often present in niche industries such as building materials retail. Moreover, customer purchasing behavior in this industry often follows specific sequences due to the nature of construction or renovation projects (e.g., purchasing cement before bricks or insulation before drywall). These patterns are rarely captured adequately by static recommendation models. By leveraging bipartite graph structures as input for LSTM networks, it is possible to capture both the structure of customer-product interactions and the sequential patterns within them [2][6].

Therefore, this study is motivated by the need to improve product recommendation accuracy in the building materials retail industry by modeling the sequential nature of transactions and structural interactions between customers and products. The objective is to develop a hybrid recommender system that combines link prediction based on graph structures and sequential learning using LSTM, with the aim of increasing product sales at PT. XYZ. This research is expected to contribute by providing insights into customer behavior patterns, improving decision-making processes for marketing strategies, and helping the company offer more personalized recommendations to their customers.

The remainder of this paper is organized as follows: Section 2 discusses related works concerning recommendation methods using graphs and deep learning. Section 3 explains the methodology, including dataset descriptions and system design. Section 4 presents results and evaluations. Finally, Section 5 concludes the paper and outlines possible directions for future research.

2. RESEARCH METHOD

2.1 Notation and Problem Definition

A bipartite network consists of two disjoint sets of nodes, where edges only exist between nodes from different sets. Formally, a bipartite graph is represented as a graph $G = (X, Y, E)$ where X is the set of customer nodes $\{c_1, c_2, \ldots, c_n\}$, Y is the set of product nodes $\{p_1, p_2, \ldots, p_m\}$. Nodes X and Y are disjoint $X \cap Y = \emptyset$ and edge $E \subseteq X \times Y$ represents the edges (interactions or transactions) between X and Y with no self loop within the same node.

This structure is widely used in recommender systems, where the goal is to predict missing links (i.e., future interactions) based on existing connections. In this study, a bipartite graph is constructed to model customer-product interactions. Customers and products form distinct node sets, with edges representing transactions. The graph-based recommendation approach has been widely used in various domains, including e-commerce and social media. The implementation follows the methodology of previous works on graph-based recommendation systems [1][12].

Figure 1 Bipartite graph representation between customer nodes and product nodes, where solid lines indicate existing transactions and dashed lines represent potential transactions predicted using link prediction methods.

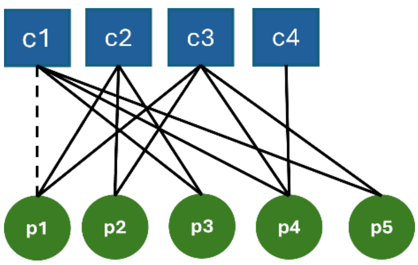

Figure 1. Bipartite graph

Figure 1 illustrates the link prediction structure using a bipartite graph, where customer nodes (c_1, c_2, c_3, c_4) are connected to product nodes (p_1, p_2, p_3, p_4, p_5). The dashed line represents the predicted link between c_1 and p_1. The algorithm calculates the similarity score by identifying common neighbors of product nodes connected to c_1 and customers who purchased the same products, excluding c_1 itself. The final similarity score is obtained by dividing the number of common neighbors by the total unique customer nodes.

2.1.1 Representation

In the context of bipartite graphs, customer-product interactions are represented as sequential events. The bipartite graph is transformed into an input sequence where each time step corresponds to a customer's transaction with a specific product. The input at each time step consists of node embeddings that encode node features or one-hot encoded representations.

The input sequence $X = \{x_1, x_2, \ldots, x_t\}$ is then processed by the LSTM model, which generates hidden states representing the temporal evolution of customer preferences. The final hidden state ht is used to predict the likelihood of future interactions, facilitating link prediction between customers and products.

2.1.2 Problem Difinition

Using a bipartite graph representation of historical transactions in building materials retail, the primary objective is to predict potential customer-product links, indicating future purchases, as illustrated by the dashed line in Fig. 1. The second objective is to enhance recommendation accuracy by integrating graph-based link prediction with sequential learning models

2.2 Link Prediction Concept and Methods

2.2.1 Concept

Link prediction in bipartite graphs estimates the likelihood of future customer-product interactions (edges) based on historical transaction data (nodes and existing edges). This approach is widely used in recommender systems to predict unobserved connections [1][12]. Traditional similarity-based methods are limited in capturing dynamic retail purchasing patterns, necessitating hybrid graph-neural approaches [28][45].

2.2.2 Implemented Methods

We evaluate four link prediction techniques, selected for their complementary strengths in bipartite networks:

1. Katz Index
   Computes weighted paths between nodes, favoring shorter paths [26]:

   $$S_{Katz}(u,v) = \sum_{l=1}^{\infty} \beta^l \,.\, |\, paths\ \frac{(l)}{u,v}\ | \tag{1}$$

   where β is a decay factor (empirically set to 0.05). Outperforms basic path-counting in sparse retail graphs [45].

2. Personalized PageRank (PPR)
   Adapts PageRank for bipartite structures using random walks with restart [12][16]:

   $$r = (1-\alpha)\,.\,M\,.\,r + \alpha\,.\,p \tag{2}$$

   where α = 0.15 (restart probability) and M is the transition matrix. Effective for capturing global network topology [1].

3. Adamic-Adar Index
   Measures weighted common neighbors, emphasizing rare connections [38]:

   $$S_{AA}(u,v) = \sum_{z \epsilon \Gamma(u)\Gamma(v)} \frac{1}{\log\ |\Gamma(z)|} \tag{3}$$

   Particularly suitable for retail data with skewed product popularity [22].

4. Jaccard Similarity
   Computes neighborhood overlap [38]:

   $$S_{Jaccard}(u,v) = \frac{|\Gamma(u) \cap \Gamma(v)|}{|\Gamma(u) \cup \Gamma(v)|} \tag{4}$$

   Serves as a baseline for local structural similarity [45].

2.2.3 Evaluation Metrics

Performance is quantified using:

1. Precision@K: Proportion of correct predictions in top-K recommendations [26].
2. Recall@K: Coverage of actual purchases in top-K results [1].
3. F1-Score: Harmonic mean of precision and recall [45].
4. NDCG: Ranks recommendation quality with position discounts [12].

2.3 Methodology

2.3.1 Research Gap and Contribution

Despite the effectiveness of traditional similarity-based methods, they struggle to capture dynamic purchasing patterns and product heterogeneity in large-scale datasets. This study addresses this gap by proposing a hybrid link prediction model that integrates bipartite graph-based and LSTM deep-learning-based.

Traditional similarity-based methods (e.g., Jaccard Index, Adamic-Adar) face three key limitations in retail recommendation systems:

1. Temporal Dynamics: Fail to capture evolving customer preferences over time [28][45]
2. Heterogeneity: Struggle with diverse product attributes and purchasing patterns [1][26]
3. Scalability: Performance degrades with large transaction volumes (>100K edges) [12][38]

Our solution is a hybrid model combining:

1. Bipartite graph embeddings for structural patterns [16]
2. LSTM networks for sequential learning [7][24]
3. Integrated attention mechanisms for dynamic weighting [45]

2.3.2 Research Design

This research employs a structured methodology to develop and evaluate an AI-driven recommender system integrating Graph-Based Link Prediction and RNN with LSTM. The methodological framework consists of five key stages: (1) data preprocessing, (2) bipartite graph construction, (3) link prediction model implementation, (4) RNN model training, and (5) evaluation and validation of performance. The research follows a structured methodology consisting of the following key steps:

1. Data Preprocessing: Handling missing values, encoding categorical data, and structuring sequential transactions [1][6][26][38].
2. Bipartite Graph Construction: Creating the customer-product interaction network [12].
3. Link Prediction Model Implementation: Applying similarity-based and machine learning-based link prediction techniques [26][38].
4. RNN Model Training: Training an LSTM-based recurrent neural network for sequential purchase prediction [24][48].
5. Evaluation and Validation: Assessing model performance using ranking and classification metrics [26].

2.3.3 Dataset and Preprocessing

The dataset consists of 389,087 transaction records from the building materials retail industry, spanning 21 attributes and 120,894 unique transactions. The preprocessing steps include:

1. Handling Missing Values: Rows with significant missing values (>50%) are removed, while numerical missing values are imputed using the median method [6][38].
2. Encoding Categorical Data: Product categories and customer IDs are converted into numerical representations using Label Encoding [1][26][38].
3. Sequence Data Formation: Transaction sequences per customer are structured for RNN training, ensuring sequential integrity [24][48].

2.4 The Proposed Model

Our proposed Bipartite Link Prediction model based on LSTM aims to enhance the prediction accuracy of product selection by leveraging both structural and sequential information. The bipartite network captures customer-product interactions, while the LSTM model learns temporal purchasing patterns to predict future transactions. By integrating these approaches, the model effectively addresses data sparsity, dynamic demand shifts, and complex purchasing behaviors, leading to more accurate and personalized product recommendations.

Figure 2 illustrates the diagram workflow for a bipartite link prediction model using RNN LSTM, with PCA-based attribute filtering for promotional strategy optimization. The processes are outlined as 4 main phases:

1. Preprocessing. As the workflow was initiated, the dataset was cleaned and refined to construct well-structured raw data in a CSV file. Then, the dataset was split into two categories: training data for model learning and optimization and testing data for evaluating predictive performance and generalization Training data is filtered based on attributes customer and products to used for bipartite network construction with nodes representing them where one set of nodes represents customers, and the other represents products and edge represents purchase.

2. Modelling. As the bipartite network is created, it serves as input to the LSTM model. The training process is conducted to enable sequential learning, allowing the model to capture temporal purchase patterns and evolving customer-product interactions. Training continues until the model converges to an optimal result. The optimum result indicates the model's ability to accurately predict future purchases, balancing precision, recall, and overall recommendation performance for enhanced product selection.

3. Inferencing. The trained model is used to generate link prediction scores, representing the likelihood of a customer purchasing a specific product. Additionally, Principal Component Analysis (PCA) is applied to the dataset to identify key influential attributes (e.g., price, discount, purchase frequency) that impact customer decisions. The results from PCA provide valuable insights for optimizing pricing strategies, discounts, and targeted marketing campaigns. These insights are then utilized to develop a data-driven promotional strategy aimed at enhancing customer engagement and sales performance.

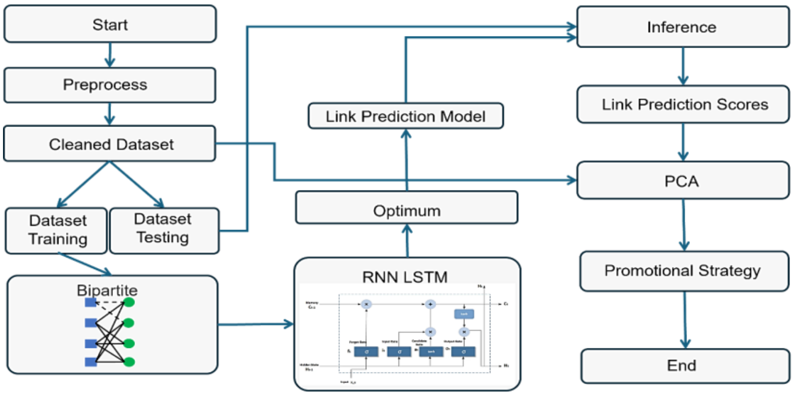

Figure 2. Hybrid Model Integrates Graph-Based Bipartite Link Prediction and Deep Learning-Based LSTM

2.4.1 Bipartite Link Prediction based on LTSM

RNNs have been extensively employed in various sequential prediction tasks due to their ability to process data with temporal dependencies [24]. The implementation of RNNs for bipartite graphs aims to model the sequential interactions between two distinct node sets, such as customers and products, for link prediction purposes. In this study, we leverage the LSTM variant of RNNs, which is particularly effective in capturing long-term dependencies in sequential data [24]. The LSTM architecture is chosen for its ability to mitigate the vanishing gradient problem, a common issue in standard RNNs, through its gating mechanisms that regulate the flow of information over time [24]. This makes LSTM particularly suitable for modeling customer purchasing sequences, where the timing and order of transactions are crucial for accurate predictions.

The LSTM model is designed to capture dynamic relationships in bipartite graphs by learning temporal patterns from sequential customer-product interactions. Each transaction between a customer and a product is represented as an ordered pair, which forms the basis of the input sequence. The sequential nature of the data allows the LSTM to model both short-term preferences and long-term purchasing habits of customers. This temporal modeling provides a more comprehensive understanding of customer behavior, which is essential for accurate link prediction [11] demonstrated the potential of RNNs in memory and information processing, particularly in capturing temporal patterns.

To represent customer-product interactions effectively, node embeddings are utilized as input features. These embeddings encode the unique characteristics of each node in the bipartite graph, providing a dense representation of customer and product attributes. The embeddings can be generated through one-hot encoding or pre-trained embedding techniques [45]. The LSTM processes these embeddings over multiple time steps, updating its hidden state to encapsulate the evolving preferences of customers [24]. The final hidden state serves as a summary of the entire sequence, which is then fed into a fully connected Dense Layer with a sigmoid activation function to predict the likelihood of future interactions.

Hyperparameter tuning plays a significant role in optimizing the performance of the LSTM model. Key hyperparameters such as the number of LSTM units, learning rate, batch size, and the number of training epochs are carefully selected through grid search experiments. The model is trained using the Adam optimizer with a learning rate of 0.001 and a batch size of 32, balancing convergence speed and model performance [24]. The choice of hyperparameters directly influences the model's ability to generalize across different customer behaviors and purchasing patterns.

The loss function employed in this study is Binary Cross Entropy, which measures the divergence between the predicted probability and the actual label. This loss function is particularly suitable for binary classification tasks, such as link prediction, where the goal is to distinguish between existing and potential interactions. The model's performance is evaluated using standard metrics including Precision, Recall, F1-Score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). These metrics provide a comprehensive assessment of the model's predictive capabilities, especially in imbalanced datasets where accurate identification of positive instances is critical [26].

The training process involves several key steps: data preprocessing, model initialization, forward propagation, backpropagation, and parameter updates. During preprocessing, customer-product interactions are transformed into sequential data, and node embeddings are generated. The LSTM model processes these input sequences, updating hidden states at each time step and producing output probabilities for link prediction. The training continues until the model converges or a predefined number of epochs is reached.

To visualize the model's performance, training and validation loss curves are plotted to illustrate convergence over time. Additionally, precision-recall curves and ROC curves provide insights into the model's predictive accuracy and robustness.

The combination of graph-based bipartite link prediction and deep learning-based LSTM architecture presents a hybrid model that integrates both structural and sequential information. This hybrid approach enhances the model's capability to recommend products more accurately by leveraging both temporal dependencies and graph topology. The proposed model serves as a powerful framework for improving recommendation systems in the building materials retail industry, where understanding customer behavior is vital for boosting sales performance.

2.4.2 Recurrent Neural Networks (RNNs)

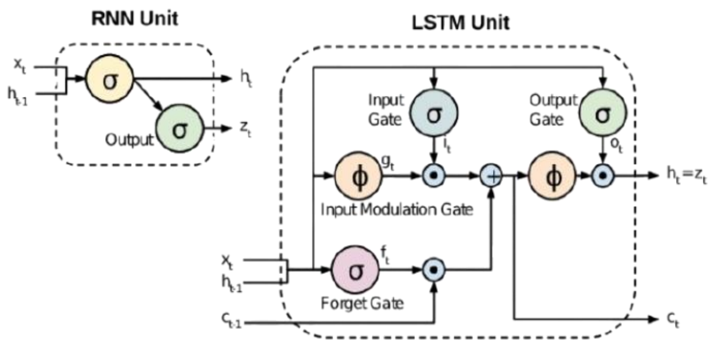

Figure 3. RNN and LSTM Architecture

RNNs are widely used for sequential data processing due to their capability to maintain information across time steps [24]. The RNN architecture, as depicted in Figure 3, consists of an input *xt*, a hidden state from the previous time step *ht-1*, and a non-linear activation function σ that computes the current hidden state *ht* [24]. The hidden state is then used to generate the output *zt*, which makes RNNs suitable for time series prediction and natural language processing tasks.

The RNN can be mathematically represented as equation 5 and 6.

$$ht = \sigma(Whx\ xt + whh\ ht - 1 + bh) \quad (5)$$

$$zt = Wzh\ ht + bz \quad (6)$$

However, standard RNNs struggle to capture long-term dependencies due to the vanishing gradient problem, where gradients diminish exponentially as they propagate through many time steps, limiting the model's ability to learn from distant past information [24]

2.4.3 Long Short-Term Memory (LSTM)

To overcome the limitations of standard RNNs, LSTM networks were introduced. The LSTM architecture, shown in Figure 3, incorporates memory cells and gating mechanisms to regulate information flow, allowing the model to capture long-term dependencies more effectively [24].

An LSTM-based RNN is utilized to predict future purchases. The LSTM unit consists of the following gates:

1. Forget Gate *ft*: Determines which information from the previous memory cell state should be retained or discarded.

$$f_t = \sigma(w_{x_f}\, x_t + W_{h_f} h_{t-1} + b_f \tag{7}$$

2. Input Gate *it*: Controls the amount of new information from the current input *xt* that is added to the cell state.

$$i_t = \sigma(w_{x_i}\, x_t + W_{hi} h_{t-1} + b_i) \tag{8}$$

3. Output Gate *ot* : Modulates the hidden state output based on the current cell state.

$$o_t = \sigma(w_{xo}\, x_t + W_{ho} h_{t-1} + b_o) \tag{9}$$

The cell state is updated using the following equations:

4. Input Modulation Gate *gt* : Transforms the input before adding it to the cell state.

$$g_t = tanh\,(w_{xg}\, x_t + W_{hg} h_{t-1} + b_g c_t = f_t \odot c_{t-1} + i_t \odot g_t h_t = o_t \odot tanh\, tanh\, c_t \tag{10}$$

The gates in LSTM units allow selective updating of the cell state, which helps prevent the vanishing gradient problem and preserves long-range dependencies across time steps [24]. Both RNN and LSTM architectures play a crucial role in sequential prediction tasks, with LSTMs demonstrating superior performance for datasets requiring long-term dependency learning.

2.5 Evaluation Metrics Formulas

1. Precision

$$Precision = \frac{TP}{TP+FP} \tag{11}$$

Measures recommendation accuracy by calculating the proportion of correctly recommended products among all recommendations [26].

2. Recall

$$Recall = \frac{TP}{TP+FN} \tag{12}$$

Quantifies the coverage of relevant products in recommendations [1].

3. F1-Score

$$F1 = 2\, \frac{Precision\ x\ Recall}{Preciion+Recall} \tag{13}$$

Balances precision and recall for imbalanced datasets [45].

4. Accuracy

$$Accuracy = \frac{TP+TN}{TP+TN\ FP+FN} \tag{14}$$

Evaluates overall prediction correctness [48].

5. Normalized Discounted Cumulative Gain (NDCG)

$$DCG \sum_{i=1}^{p} \frac{rel_i}{log_{2(i+1)}} \tag{15}$$

$$NDCG = \frac{DCG}{IDCG} \tag{16}$$

Assesses ranking quality by considering recommendation positions [26].

This structured approach ensures a rigorous evaluation of both graph-based link prediction and sequential learning models for product recommendation by computing all evaluation metrics at K=10 (top-10 recommendations) [1][26], monitoring training convergence through loss curve analysis [24], and employing 5-fold cross-validation for robustness [48].

3 RESULTS AND ANALYSIS

This section presents the research results and provides a comprehensive discussion on the performance of the proposed recommender system. The results are analyzed through multiple evaluation metrics, including Precision, Recall, F1-score, and NDCG. The findings are supported by figures, graphs, and tables to facilitate reader comprehension. The discussion is structured into the following subsections.

3.1 Dataset

The dataset used in this study consists of sales transaction data from the building materials retail industry, covering the period from January 2, 2020, to December 31, 2023. This dataset includes various columns representing transaction information, as shown in Table 1. As the goal is to predict future links indicating potential purchases, the fundamental structure must focus on customer-product interactions rather than its attributes. All attributes are then analyzed using Principal PCA is applied to identify key features influencing transaction patterns, reduce dimensionality, and eliminate redundancy while retaining essential information. This transformation improves computational efficiency and model interpretability. By using PCA, the study ensures that only the most relevant attributes are considered, reducing noise and enhancing recommendation accuracy.

Table 1. Dataset Building Materials Retail Industry

No.	Attribute	Remark	PCA Covariance	Included
1.	Transaction Date	Date of transaction	-	No (unimportant)
2.	Invoice Number	Unique invoice identifier	-	No (unimportant)
3.	Customer ID	Unique customer identifier	Main node	Yes (primary node)
4.	Customer Name	Name of customer	-	No (already represented by ID)
5.	Product	Product description	Main node	Yes (primary node)
6.	Invoice Quantity	Quantity listed in invoice		No (unimportant)
7.	Unit	Unit of measurement product	-	No (unimportant)
8.	Price/Invoice Quantity	Unit price per invoice quantity	-	No (unimportant)
9.	Quantity per Piece	Quantity per individual unit	Main node	No (unimportant)
10.	Total (Pre-disc)	Total price before discount	-	No (unimportant)
11.	% Discount	Discount rate applied	Main node	Yes (promotional strategy)
12.	Total Price (After Disc)	Total price after discount	Main node	Yes (promotional strategy)
13.	Salesperson Code	Code of salesperson	-	No (unimportant)
14.	COGS	Cost of Goods Sold	Main node	Yes (promotional strategy)
15.	Gross Profit	Profit before tax	-	Yes (promotional strategy)
16.	Quartal	Quarter of the transaction	-	Yes (promotional strategy)
17.	Year	Year of the transaction	-	Yes (promotional strategy)
18.	Per Month	Alternative month format	-	Yes (promotional strategy)
19.	Per Quartal	Alternative quarter format	-	Yes (promotional strategy)
20.	Invoice Value	Value per invoice	-.	Yes (promotional strategy)
21.	Product Category	Product category based on brand	Main Node	Yes, (promotional strategy)
22.	Price/piece (Nett)	Price per pcs Nett (after discount)	Main Node	Yes, (promotional strategy)

3.2 Experiment Setup

The experiment was conducted on the Google Colab platform using a CPU accelerator, 12 GB of RAM, and 100 GB of storage. Python served as the primary programming language, with additional libraries including NetworkX, PyTorch, and sklearn. To optimize performance, hyperparameters were set as follows:

100 epochs, a learning rate (α) of 0.001, a momentum rate (β) of 0.003, a 3-layer architecture with dimensions [16, 16, 16], a dropout rate of 0.6, and early stopping to prevent overfitting.

3.3 Experiment

This section provides a detailed overview of the experimental setup, including the methodology and key steps involved. It covers the experimental environment and execution, data preprocessing techniques, the construction of the bipartite graph representing customer-product interactions, and the optimization strategies applied to improve prediction performance. Each stage is designed to ensure the accuracy and efficiency of the proposed model in generating reliable recommendations.

3.4 Preprocessing

The preprocessing steps were performed to ensure the dataset is ready for analysis and modeling, including data cleaning, feature engineering, categorical encoding, and dataset filtering. The specific steps performed in this study are as follows:

1. Handling Missing Value. Missing data can affect model performance. Therefore, the first step is to handle missing values using the following methods:
    a. Removing Rows/Columns: If a column has too many missing values (e.g., more than 50% missing), it can be removed. Similarly, rows with excessive missing values may be deleted if they are not significant.
    b. Data Imputation: For numerical columns, missing values can be filled using the median or mean. For example, if the "Price per Kts. Invoice" column has missing values, they can be replaced with the median or mean of that column.

2. Feature Engineering. To create new features that improve model performance. The following steps are applied:
    a. Calculating Nett Price per Unit: A new feature is calculated using the formula: Nett Price per Unit = Total Price / Quantity per pcs. This feature helps in understanding the net price per product unit.
    b. Adding Transaction Time Attributes: For time-based analysis, new columns such as "Month" and "Year" are extracted from the "Invoice Date" column. This enables the analysis of purchasing patterns over time.

3. Encoding Product Categories. Categorical columns such as "Product Category" need to be converted into numerical representations for machine learning algorithms. The encoding techniques used include:
    a. Label Encoding: Assigning a unique numerical value to each category.
    b. One-Hot Encoding: Creating binary columns for each category. The choice of encoding technique depends on the requirements of the algorithm used.

4. Filtering the Dataset. To focus the analysis, only relevant columns are selected, including: Invoice Date, Customer Code/Customer Name, Product Description, Nett Price per Unit, Quantity per pcs, Total Price, Product Category.

3.5 Graph Construction, and Representation

The bipartite graph is constructed using the NetworkX library, which provides tools for creating, manipulating, and analyzing graph structures. Using file csv as input, the process of construction the bipartite graph was primarily divided into three sub-process, as shown in Algorithm 1 and graph bipartite top 5 customer-product can be seen in Figure 4.

1. Dataframe preparation:
    a. The raw transaction data is loaded from a CSV file using pandas.
    b. Relevant columns such as Customer ID, Product ID, Transaction Date, Quantity, Price, and Discount are selected.
    c. Data cleaning is performed by handling missing values, removing duplicates, and standardizing categorical values.
    d. Additional computed attributes, such as total purchase value (quantity × price – discount), are derived for further analysis.

2. Node Assignment:
    a. One set of nodes represents customers (using customer codes as identifiers).

   b. The other set represents products (using product descriptions as identifiers).
3. Edge Assignment:
   a. Edges between customers and products are added based on transaction records, where each transaction creates an edge linking a customer to the purchased product.
   b. Edge weights can be assigned based on the total transaction value or purchase frequency.

3.6 LSTM Approach.

LSTM networks are a type of RNN designed to capture sequential dependencies in data. Unlike traditional RNNs, LSTMs use gating mechanisms to regulate the flow of information and mitigate the vanishing gradient problem [24]. LSTM networks have been widely applied in session-based recommendations [48] and sequential purchasing behavior [28]. The model's ability to preserve long-term dependencies makes it particularly effective in predicting future purchases based on historical transaction sequences. Algorithm 2 illustrates the process of utilizing the bipartite graph and sequential data as inputs, processing them through the LSTM model to generate link prediction scores, and subsequently evaluating the model's performance.

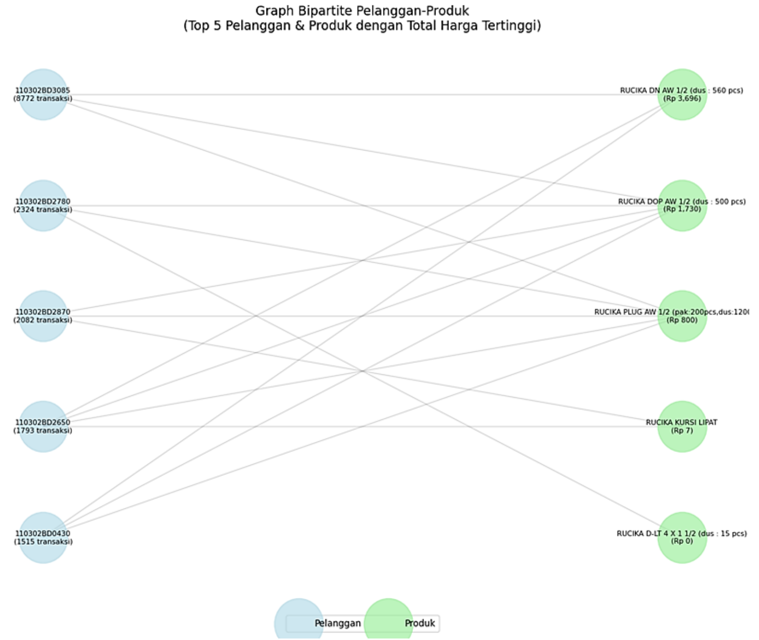

Figure 4. Graph Bipartite Top 5 Customer-Product

Algorithm 1. Bipartite Graph Construction

```
Input
#1.  File dataset.csv

Process
#1. Load libraries
#2. Dataframe preparation
#3. Column name standardization
#4. Data cleaning
#5. Filtering required nodes
#6. Data shuffle
#7. Bipartite construction

    G = nx.Graph()

    # Add nodes for Customers and Products
    G.add_nodes_from(df['Customer'].unique().tolist(), bipartite=0)
    G.add_nodes_from(df['Product'].unique().tolist(), bipartite=1)

    # Add edges for Customer-Product relationships
    for _, row in dataset.iterrows():
        G.add_edge(row['Customer'], row['Product'])

Output
#1. A rendered bipartite network
```

Algorithm 2: LSTM Model Training

```
Input
#1. Bipartite network of customer and product
#2. Interaction sequences S
#3. Node embeddings E
#4. Hyperparameters (Epochs, Batch Size, Learning Rate, Hidden Units)

Process
#1. Data preprocessing
#2. Model initialization
#3. Forward propagation
#4. Link prediction score calculation
#5. Loss function (Binary Cross Entropy)
#6. Backpropagation
#7. Training loop
       model = LSTM(input_dim=d, hidden_dim=h, num_layers=n_layers)
       optimizer = Adam(model.parameters(), lr=0.001)
       loss_fn = nn.BCELoss()

       for epoch in range(epochs):
           optimizer.zero_grad()
           output = model(S)
           loss = loss_fn(output, target)
           loss.backward()
           optimizer.step()

Output
#1. Link prediction scores
#2. Evaluation metrics (Precision, Recall, F1-Score, AUC-ROC)
```

3.7 Performance Evaluation of Link Prediction Methods

To assess the effectiveness of the link prediction models, experiments were conducted using various methods, including Katz Index, Personalized PageRank, and Adamic-Adar Index. The evaluation metrics used in this assessment are Precision, Recall, and F1-score, as summarized in Table 2.

Table 2. Performance Comparison of Link Prediction Methods

Method	Precision	Recall	F1-Score
Katz Index	0.79	0.75	0.77
PageRank	0.82	0.78	0.80
Adamic-Adar	0.76	0.74	0.75

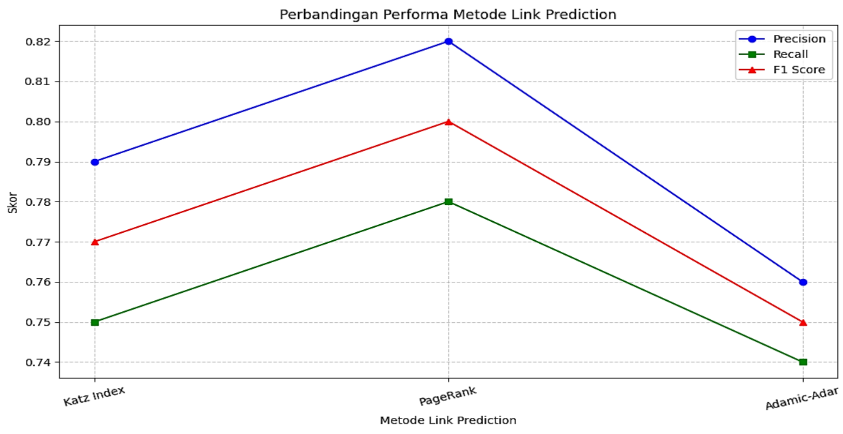

Figure 5. Link Prediction Performance Across Different Methods

From Figure 5, it can be observed that Personalized PageRank achieved the highest performance across all evaluation metrics, indicating its suitability for predicting relevant product links in the bipartite graph.

3.8 Performance Evaluation of RNN-Based Recommendation System

The next phase evaluates the LSTM-based RNN for sequential purchase prediction. The dataset was split into training and testing sets, and the model was trained using a categorical cross-entropy loss function. The evaluation results are summarized in Table 3.

Table 3. Performance of LSTM-Based Recommender System

Metric	Value
Accuracy	0.854
Precision	0.82
Recall	0.80
F1-score	0.81
NDCG	0.86

As shown in Table 3, the LSTM model achieved an accuracy of 85.4%, demonstrating its ability to predict the next product purchase effectively. The NDCG score of 0.86 further indicates that the ranking of recommended items closely aligns with actual user preferences.

3.9 Loss Curve Analysis

To ensure the stability of the training process, a loss curve analysis was conducted. Figure 6 illustrates the convergence of the loss function over multiple epochs.

The loss curve indicates that the model converges after approximately 20 epochs, suggesting optimal learning without overfitting.

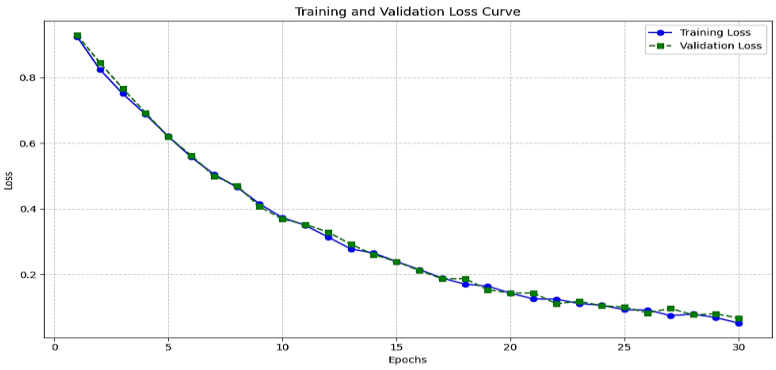

Figure 6. Training and Validation Loss Curve

3.10 Comparative Analysis Deep Learning-Based vs Similarity-Based

To further validate the effectiveness of the proposed system, a comparison with traditional similarity-based methods such as Jaccard Similarity, Common Neighbors (CN), and Adamic/Adar Index was conducted. The results are summarized in Table 4.

Table 4. Comparative Performance of Recommender Systems

Method	Precision	Recall	F1-score	NDCG
Jaccard	0.68	0.65	0.66	0.70
Common Neighbors	0.72	0.70	0.71	0.75
Adamic/Adar	0.74	0.72	0.73	0.77
Proposed Method (LSTM)	0.82	0.80	0.81	0.86

In Figure 7, the LSTM-based approach consistently outperforms traditional methods across all evaluation metrics, demonstrating its capability to capture sequential purchase patterns effectively.

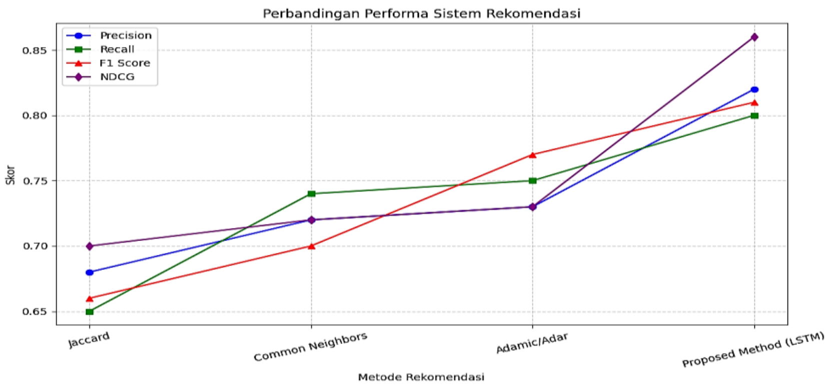

Figure 7. Performance Comparison of Recommendation Methods

3.11 Promotional Strategy

After conducting Principal PCA, the top five attributes that significantly influence customer purchasing behavior were identified and are presented in Table 5. These attributes serve as the cornerstone for devising targeted promotional strategies aimed at increasing customer engagement and purchase frequency, particularly among frequent buyers.

Top 5 Influential Attributes:

1. Total Price – Represents the total amount spent per transaction, playing a pivotal role in understanding customer purchasing capacity and behavior.
2. Quantity per Pcs – Reflects the volume of items purchased per transaction, which is crucial for demand forecasting and inventory management.
3. % Discount – Highlights the impact of promotional offers on customer purchase decisions, providing insights into price sensitivity and promotion effectiveness.
4. Gross Profit – Serves as an indicator of product profitability, helping to prioritize products for promotional campaigns.
5. COGS Value (Cost of Goods Sold) – Offers a detailed understanding of product cost structure, aiding in the formulation of competitive pricing strategies.

These influential attributes form the basis for personalized promotional campaigns by dynamically adjusting pricing, offering discounts, and designing tailored marketing messages. Table 5 outlines three exemplary promotional strategies for a selected customer (ID: 123312), who has made 12 purchases with an average minimum of two items per transaction.

By aligning promotional strategies with customer purchasing patterns and preferences, the company can bolster customer satisfaction, foster brand loyalty, drive repeat purchases, and ultimately enhance overall profitability.

Table 5. Recommended Promotional Strategies

No.	Product	Score	Normal Price	Disc%	Adjustment Price	Time-Limited Promotion	Loyalty Points	Remarks
1.	RUCIKA STANDARD AW 1/2	0.1321	16800	16+3+4	13141	1 week	1000	minimum order 50 btg
2.	RUCIKA KNEE AW 1/2	0.1423	2500	16+3+4	1955	1 week	1000	minimum order 2 box
3.	ONDA BC 1/2	0.2123	34000	25+7,5	23587	1 week	1000	minimum order 1 box

3.12 Discussion

The experimental results confirm that the proposed system effectively enhances product recommendations in the building materials retail industry. Several key insights can be drawn:

1. Graph-Based Link Prediction significantly improves product association accuracy, with Personalized PageRank achieving the best performance.
2. LSTM-based RNNs effectively predict sequential purchases, surpassing traditional methods in accuracy and ranking effectiveness.
3. The high NDCG score (0.86) indicates that recommended products align well with actual user preferences, highlighting the system's real-world applicability.

4. CONCLUSION

Most existing link prediction studies primarily rely on similarity-based approaches. While these methods have demonstrated effectiveness on small-scale datasets, they often struggle to capture complex interaction patterns and sequential dependencies present in large-scale transactional environments. This study proposes a deep learning-based recommender system that integrates RNN with LSTM and graph-based link prediction, specifically designed to handle the vast and intricate nature of retail transaction data in the building materials industry. By modeling customer-product interactions as a bipartite graph and incorporating sequential purchase behavior through LSTM, the proposed approach effectively captures temporal dynamics and complex relationships. Experimental results on a large-scale dataset confirm that the model significantly outperforms traditional similarity-based methods—such as Jaccard, Adamic-Adar, and Common Neighbor—when evaluated using precision, recall, F1-score, and NDCG metrics. Additionally, Principal PCA is applied to reduce feature dimensionality and highlight key attributes influencing purchasing decisions. This allows for more personalized and accurate product recommendations, particularly for frequent buyers, enabling the implementation of data-driven promotional strategies aimed at boosting customer retention and sales.

In the future, we plan to further our research in two directions. First, by integrating weighted labels on customer and product nodes to represent interaction strength (i.e., purchasing frequency, spending, quantity), enhancing the expressiveness of the bipartite structure. Second, by capturing temporal aspects of purchasing behavior through advanced sequence modeling techniques, enabling the system to better adapt to seasonal trends, project-based demands, and evolving customer preferences over time. With these future directions, we aim not only to refine recommendation accuracy but also to deliver a more adaptive and dynamic recommendation system that aligns with evolving customer behaviors and industry demands.

REFERENCES

[1] H. Zhang, J. Wu, H. Huang, and S. Wang, "A Survey on Recommender Systems Based on Graph Neural Networks," IEEE Access, vol. 9, pp. 155791–155809, 2021. doi: 10.1109/ACCESS.2021.3129293.
[2] Xue, H., Yang, L., Jiang, W., Wei, Y., Hu, Y., Lin, Y., 2021. Modeling Dynamic Heterogeneous Network for Link Prediction Using Hierarchical Attention with Temporal RNN. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 12457 LNAI, 282–298. https://doi.org/10.1007/978-3-030-67658-2_17
[3] Cui, Q., Wu, S., Liu, Q., Zhong, W., Wang, L., 2020. MV-RNN: A Multi-View Recurrent Neural Network for Sequential Recommendation. IEEE Trans. Knowl. Data Eng. 32, 317–331. https://doi.org/10.1109/TKDE.2018.2881260
[4] M. Sholeh, R. Y. Rachmawati, and E. Susanti, "Pemodelan Basis Data Graph dengan Neo4j (Studi Kasus: Basis Data Sistem Informasi Penjualan pada UMKM)," J. Teknol. Inf. dan Terap., vol. 7, pp. 25–32, 2020, doi: 10.25047/jtit.v7i1.129.
[5] B. Susilo, Y. Setiawan, and Aryani, "Perancangan Sistem Rekomendasi Pemilihan Cinderamata Khas Bengkulu Berbasis E-Marketplace," J. Rekursif, vol. 7, pp. 70–76, 2019.
[6] Xu, M., Liu, W., Xu, J., Xia, Y., Mao, J., Xu, C., Hu, S., Huang, D., 2022. Recurrent Neural Network Based Link Quality Prediction for Fluctuating Low Power Wireless Links. Sensors 22. https://doi.org/10.3390/s22031212
[7] S. Saadah and P. E. Yunanto, "Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) Methods to Forecast Daily Turnover at BM Motor Ngawi," Indones. J. Artif. Intell. Data Min., vol. 7, no. 2, pp. 141–147, 2024.
[8] L. Wu, P. Sun, Y. Fu, and X. He, "Graph Neural Networks in Recommender Systems: A Survey," arXiv preprint arXiv:2012.00973, 2020.
[9] A. Rath and S. R. Sahu, "Recurrent Neural Networks for Recommender Systems," Comput. Intell. Mach. Learn., vol. 1, no. 1, pp. 31–36, 2020, doi: 10.36647/ciml/01.01.a004.
[10] I. D. Mienye, T. G. Swart, and G. Obaido, "Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications," Information, vol. 15, no. 9, p. 517, 2024, doi: 10.3390/info15090517.
[11] Livathinos, N., Berrospi, C., Lysak, M., Kuropiatnyk, V., Nassar, A., Carvalho, A., Dolfi, M., Auer, C., Dinkla, K., Staar, P., 2021. Robust PDF Document Conversion Using Recurrent Neural Networks. 35th AAAI Conf. Artif. Intell. AAAI 2021 17A, 15137–15145. https://doi.org/10.1609/aaai.v35i17.17777
[12] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph-based Recommendation in E-commerce: Methods and Applications," in Proceedings of the 13th ACM Conference on Recommender Systems, 2018, pp. 538–539. doi: 10.1145/3298689.3347047.
[13] N. R. Madjid, Y. Vitriani, E. Haerani, and F. Kurnia, "Sistem Rekomendasi Hotel di Provinsi Riau dengan Metode AHP dan SAW," KLIK Kaji. Ilm. Inform. dan Komput., vol. 3, no. 6, pp. 945–956, 2023, doi: 10.30865/klik.v3i6.931.
[14] L. Pereira, T. Hanggari, C. Rini, and M. Sorong, "Pengaruh Lingkungan Eksternal dan Lingkungan Internal terhadap Kinerja UKM melalui Keunggulan Bersaing pada UKM di Kota Sorong," J. Ilm. Nas., vol. 4, no. 1, pp. 162–169, 2022.
[15] H. Huang, Y. Fan, L. Li, J. Chen, and T. Zhou, "A Novel News Recommendation Model with Knowledge Enhancement and Stability," Eur. J. Artif. Intell., 2025, doi: 10.1177/30504554241301391.
[16] R. van den Berg, T. N. Kipf, and M. Welling, "Graph Convolutional Matrix Completion," arXiv preprint arXiv:1706.02263, 2017.
[17] Xie, Y., Li, C., Yu, B., Zhang, C., Tang, Z., 2020. A Survey on Dynamic Network Embedding.
[18] Cai, X., Shu, J., Al-Kali, M., 2019. Link Prediction Approach for Opportunistic Networks Based on Recurrent Neural Network. IEEE Access 7, 2017–2025. https://doi.org/10.1109/ACCESS.2018.2886360
[19] Li, F.-F., Johnson, J., Yeung, S., 2018. Lecture 10: Recurrent Neural Networks.
[20] Hamilton, W.L., 2003. Inductive Representation Learning on Large Graphs. Rev. Prat. du Froid du Cond. d'Air 59.
[21] M. Muhajir and P. Canas, "Indonesian Inflation Forecasting with Recurrent Neural Network Long Short-Term Memory (RNN-LSTM)," J. Statistika dan Sains Data, vol. 4, no. 2, pp. 132–142, 2024.
[22] Qandi, G.A., Rakhmawati, N.A., 2021. Implementasi Algoritma Link Prediction Untuk Mencari Kesamaan Antara Calon Legislatif Pemilihan Umum Indonesia 2019. J. SIMETRIS 12, 1–8.
[23] Taiwo, R., Bello, I.T., Abdulai, S.F., Yussif, A.-M., Salami, B.A., Saka, A., Zayed, T., 2024. Generative AI in the Construction Industry: A State-of-the-art Analysis.
[24] Sherstinsky, A., 2020. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Phys. D Nonlinear Phenom. 404, 1–43. https://doi.org/10.1016/j.physd.2019.132306
[25] I. Maula, A. Tholib, and M. N. F. Hidayat, "Faktorisasi Matriks Menggunakan Stochastic Gradient Descent untuk Optimasi Sistem Rekomendasi Hotel," NJCA, vol. 9, no. 1, pp. 19–25, 2024.

[26] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, “Neural Graph Collaborative Filtering,” in Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 165–174. doi: 10.1145/3331184.3331267.

[27] Wang Tingwu, 2016. Why do we need Recurrent Neural Network? p. 41.

[28] Wu, C., Wang, Y., Jia, T., 2023. Dynamic Link Prediction Using Graph Representation Learning with Enhanced Structure and Temporal Information. Proc. 2023 26th Int. Conf. Comput. Support. Coop. Work Des. CSCWD 2023 279–284. https://doi.org/10.1109/CSCWD57460.2023.10152711

[29] Oktavia, C., Nurkholis, A., 2022. Artificial Intelligence Untuk Keberlangsungan Bidang Konstruksi. JUMATISI J. Mhs. Tek. Sipil 3, 244–249. https://doi.org/10.24127/jumatisi.v3i2.4114

[30] Afifah, K.H., 2023. Analisis SWOT Pada Toko Tradisional Bahan. Universitas Semarang.

[31] Yolanda, G., 2019. Analisis Strategi Bersaing UD Duta Keramik di Jember. J. Agora 7, 35–57.

[32] Maretasari, L., 2024. Analisis Lingkungan Internal Dan Eksternal Pada Bidang Usaha Distributor Bahan Bangunan (Studi Kasus CV Mitra Karya Sukses).

[33] Liu, C., Han, Y., Xu, H., Yang, S., Wang, K., Su, Y., 2024. A Community Detection and Graph-Neural-Network-Based Link Prediction Approach for Scientific Literature. Mathematics 12, 1–20. https://doi.org/10.3390/math12030369

[34] Badiati, N., Zulistiani, 2023. Peningkatan Daya Saing Toko Bangunan Sumber Rejeki Kabupaten Blitar Melalui Strategi Pemasaran Berbasis Analisis SWOT. J. Simki Econ. 6, 452–462.

[35] Febriana Fatimah Putri1*, M.F., 2024. Perancangan Sistem Rekomendasi Sabun Cuci Muka Menggunakan Algoritma TOPSIS Hal. 1166-, 1166–1172.

[36] Y. Ma, J. Tang, X. Wang, A. M. Jalali, J. Li, and D. Yin, “Streaming Recommender Systems,” in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1369–1378. doi: 10.1145/3397271.3401114.

[37] Teknologi, S.R., Chicaiza, J., Komputer, D.I., Técnica, U., Loja, P. De, 2021. Survei Komprehensif Grafik Pengetahuan Berbasis Sistem Rekomendasi: Teknologi, Pengembangan, dan Kontribusi.

[38] Siregar, Ivan M., Pratama, D., Himawan, C., 2024. Penggunaan Jaccard Similarity Coefficient dalam Optimasi Proses Rekrutmen Karyawan Berbasis Profil dan Kompetensi. SINTECH (Science Inf. Technol. J. 7, 101–111. https://doi.org/10.31598/sintechjournal.v7i2.1617

[39] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, “GNNExplainer: Generating Explanations for Graph Neural Networks,” in Advances in Neural Information Processing Systems, vol. 32, 2019.

[40] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, “Graph Neural Networks: A Review of Methods and Applications,” AI Open, vol. 1, pp. 57–81, 2020. doi: 10.1016/j.aiopen.2021.01.001.

[41] Q. Wu, X. He, H. Zhang, and T.-S. Chua, “Session-based Recommendation with Graph Neural Networks,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 346–353.

[42] C. Fan, X. Ma, Q. Zhang, Y. Li, X. Xie, and D. Jin, “A Graph Neural Network Framework for Understanding Customer Behavior in E-commerce,” arXiv preprint arXiv:1905.12356, 2019.

[43] R. Rossi, A. Rao, and N. Ahmed, “Temporal Graph Networks for Deep Learning on Dynamic Graphs,” arXiv preprint arXiv:2006.10637, 2020.

[44] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural Collaborative Filtering,” in Proceedings of the 26th International Conference on World Wide Web (WWW), 2017, pp. 173–182. doi: 10.1145/3038912.3052569.

[45] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," Advances in Neural Information Processing Systems, 2019, pp. 5165–5175.

[46] M. Fan, P. Wang, Y. Huang, Q. He, and X. He, “Graph Neural Networks for Social Recommendation,” in Proceedings of the Web Conference 2019 (WWW), 2019, pp. 417–426. doi: 10.1145/3308558.3313416.

[47] E. Choi, S. Biswal, B. Malin, J. Duke, W. Stewart, and J. Sun, “Generating Multi-label Discrete Patient Records using Generative Adversarial Networks,” in Machine Learning for Healthcare Conference, 2017, pp. 286–305.

[48] J. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, “Session-based Recommendations with Recurrent Neural Networks,” in Proceedings of the 4th International Conference on Learning Representations (ICLR), 2016.

[49] M. Fan, J. Ma, P. Wang, Y. Huang, and X. He, “Learning to Recommend with Multiple Graphs,” in Proceedings of the 14th ACM Conference on Recommender Systems (RecSys), 2020, pp. 525–534. doi: 10.1145/3383313.3412234.

[50] L. Wu, Y. Sun, Y. Fu, and X. He, “A Comprehensive Survey on Graph Neural Networks,” IEEE Transactions on Neural Networks and Learning Systems, early access, 2021. doi: 10.1109/TNNLS.2020.2978386.

[51] B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online Learning of Social Representations,” in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710. doi: 10.1145/2623330.2623732.

BIBLIOGRAPHY OF AUTHORS

Ivan Michael Siregar earned both his Bachelor's and Master's degrees in Informatics Engineering from Institut Teknologi Bandung. As an information technology consultant, he currently assists various companies with Odoo implementation. In addition, he is a lecturer at Institut Teknologi Harapan Bangsa (ITHB) in Bandung, Indonesia. He is currently pursuing a Ph.D. in Computer Science, specializing in Artificial Intelligence, at Universiti Kebangsaan Malaysia.

Firlie Resti Rosdiana is a final-year undergraduate student in the Information Systems program at Institut Teknologi Harapan Bangsa (ITHB), Bandung, Indonesia, with a concentration in Data Science. She has over 5 years of professional experience in data analysis, logistics, and administration in the retail building materials industry. Her research interests include graph-based recommendation systems, link prediction, and data-driven decision-making. During her academic journey, Firlie has worked on projects involving bipartite graphs to improve the accuracy of product recommendation systems by leveraging link prediction techniques. She is also actively involved in various community projects as an event organizer and has experience in data visualization using Looker Studio and Excel. Her passion for data science and commitment to technological innovation drives her to explore how AI-based systems can enhance customer experience and business performance.