Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 24, No. 2, March 2025, pp. 207∼218
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v24i2.4641

Revealing Interaction Patterns in Concept Map Construction Using Deep Learning and Machine Learning Models

F.ti Ayyu Sayyidul Laily[1], Didik Dwi Prasetya[1], Anik Nur Handayani[1], Tsukasa Hirashima[2]
[1]Universitas Negeri Malang, Malang, Indonesia
[2]Hiroshima University, Hiroshima, Japan

Article Info

Article history:

Received November 28, 2024
Revised December 18, 2024
Accepted January 28, 2025

Keywords:

Concept map;
Interaction pattern;
K-Nearest Neighbors;
Long Short-Term Memory;
Random Forest.

ABSTRACT

Concept maps are educational tools for organizing and representing knowledge, enhancing comprehension, and memory retention. In concept map construction, much knowledge can be utilized. Still, concept map construction is complex, involving actions that reflect a user's thinking and problem-solving strategies. Traditional methods struggle to analyze large datasets and capture temporal dependencies in these actions. To address this, the study applies deep learning and machine learning techniques. This research aims to evaluate and compare the performance of Long Short-Term Memory (LSTM), K-Nearest Neighbors (K-NN), and Random Forest algorithms in predicting user actions and uncovering user interaction patterns in concept map construction. This research method collects and analyzes interaction logs data from concept map activities, using these three models for evaluation and comparison. The results of this research are that LSTM achieved the highest accuracy (83.91%) due to its capacity to model temporal dependencies. Random Forest accuracy (80.53%), excelling in structured data scenarios. K-NN offered the fastest performance due to its simplicity, though its reliance on distance-based metrics limited accuracy (70.53%). In conclusion, these findings underscore the practical considerations in selecting models for concept map applications; LSTM demonstrates effectiveness in predicting user actions and excels for temporal tasks, while Random Forest and K-NN offer more efficient alternatives in computational.

Corresponding Author:

Didik Dwi Prasetya, +6281297000116,
Department of Electrical and Informatics Engineering,
Universitas Negeri Malang, Malang, Indonesia,
Email: didikdwi@um.ac.id.

How to Cite:
F. Laily, D. Prasetya, A. Handayani, and T. Hirashima, "Revealing Interaction Patterns in Concept Map Construction Using Deep Learning and Machine Learning Models", *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 24, No. 2, pp. 207-218, March, 2025.

1. INTRODUCTION

A concept map is a graphical tool that serves to organize and represent one's knowledge and has been used in various disciplines for learning, teaching, and assessing learning outcomes [1]. Concept maps provide an interactive method for visualizing and organizing knowledge, connecting concepts through nodes and links that reflect meaningful relationships [2] and provides a visual representation of a person's comprehension of a given subject [3]. This structured format aids comprehension and memory retention by encouraging learners to engage with content actively, constructing and reconstructing ideas in a format that emphasizes connections. Concept maps are composed of three main parts: nodes that contain a concept, lines that connect one node to another, and phrases as labels that describe the relationship between nodes [4]. Two nodes connected with a labeled connecting line are called propositions. As the smallest part of a concept map, the proposition determines the quality of knowledge understood by each individual because each proposition will represent its meaning in a concept map. As learners interact with these maps, their behaviors, such as adding new nodes, forming links between concepts, and restructuring content, generate unique interaction patterns [5]. Concept maps can be made in two fundamental ways: closed-ended and open-ended [6]. In open-ended concept maps, people can create and add their concepts and associations to an open-ended concept map without any restrictions or preset framework. Because students can build the concept map according to their understanding and insights, this method fosters creativity and flexibility [7]. Conversely, creating a closed-ended concept map adheres to a predetermined framework or template that includes particular concepts and connections. This approach offers a more directed and organized learning process, emphasizing particular ideas and connections that the teacher or curriculum determines are crucial [8].

In recent years, Machine learning (ML) and deep learning (DL) techniques have become fundamental to educational technology [9, 10], it offers the potential to analyze data, identify user behavior patterns, and make predictive adjustments that enhance personalization and user experience. User interaction data is a valuable resource in understanding and predicting user behavior, particularly in educational tools such as concept map applications [11]. Recognizing and predicting user interaction patterns in these applications can provide valuable insights, from enhancing personalized learning experiences to automating feedback and support systems. Traditional methods of analyzing these processes are often limited by their inability to handle large datasets or capture the temporal dependencies in the sequence of actions. This research addresses these challenges by applying machine learning and deep learning techniques K-Nearest Neighbors (K-NN), Random Forest, and Long Short-Term Memory (LSTM) to predict user actions and reveal hidden patterns in constructing concept maps. One supervised machine-learning approach that is primarily utilized for classification is the traditional K-NN algorithm [12]. The number of "nearest neighbors," or k, is a changeable parameter in the procedure. Finding the closest data point or neighbors from a training dataset for a query is how the K-NN algorithm works. The closest distances from the query point are used to determine which data points are closest [13]. On the other hand, a random forest algorithm is a nonlinear model that combines several decision trees into a forest is the random forest algorithm [14]. Random sampling and majority voting are crucial concepts to remember when analyzing random forests. In particular, the training set for every decision tree is chosen at random from the complete sample set [15]. For the DL, one of the most popular models for time series prediction is the LSTM model [16]. The bidirectional LSTM model can also be used to predict user interaction patterns because it clearly outperforms the LSTM in considering both past and future information.

Among these, K-NN, Random Forest, and LSTM represent diverse approaches in data processing, each bringing unique advantages for recognizing patterns and making accurate predictions. That is why this study used these 3 models. LSTM networks, as sequential models, are well-suited for analyzing temporal user interactions due to their ability to capture long-term dependencies and patterns over time [17]. K-NN, on the other hand, is a simpler yet powerful, proximity-based model that identifies patterns by calculating similarity among data points [18]. Meanwhile, Random Forest, a robust ensemble method built on multiple decision trees, offers strength in handling complex, high-dimensional data with high predictive accuracy [19]. Exploring the application of these models within the specific domain of interactive concept map construction can reveal insights that lead to more adaptive and responsive learning systems. While individual learning models like LSTM, K-NN, and Random Forest have been widely used for classification and prediction tasks, their specific applications in predicting user actions and uncovering interaction patterns during concept map construction remain underexplored. Existing studies often focus on traditional educational analytics or single-model approaches but lack a comprehensive comparison of these models in the context of temporal sequence analysis (e.g., LSTM) [20] versus instance-based learning (K-NN) [21] and ensemble learning (Random Forest) [22]. Moreover, there is limited work analyzing how these models can synergistically provide insights into interaction dynamics, offering actionable feedback for personalized learning systems or collaborative tools. This creates an opportunity to compare model performance in predicting user behavior and extracting interaction patterns. Investigate the unique strengths and weaknesses of these algorithms in handling diverse datasets. This research addresses these gaps by evaluating the models in-depth and connecting their outputs to meaningful interpretations for concept map applications.

This study presents a comparative analysis of LSTM, K-NN, and Random Forest models for recognizing user interaction patterns and predicting future actions within concept map construction. For instance, LSTM's strength in sequential data processing may make it better suited for detecting gradual changes in user behavior [23]. Random Forest's versatility with structured data could make it effective for identifying specific decision-making patterns [24]. Meanwhile, K-NN's proximity-driven metrics provide a straightforward method for analyzing local patterns in interaction data. This analysis not only identifies the strengths and weaknesses of each model in educational contexts but also highlights the discovery of patterns in user interaction. By evaluating these models on metrics such as accuracy, precision, recall, and computational efficiency, the aim is to identify the strengths and limitations of each approach [25]. This research contributes to advancing intelligent systems for educational applications, with implications for improving adaptive learning environments and enhancing user engagement in concept map-based platforms. The importance of this research extends beyond technical evaluation; it contributes to the broader goal of advancing adaptive educational technologies that respond to individual learning needs. The findings from this study could inform the design of learning systems that offer data-driven personalization, adapting to users' activity in building concept maps as they navigate learning materials. This study builds on existing state-of-the-art research in ML-DL-based adaptive learning, pushing the boundaries of what is possible with user-centered educational tools. By systematically comparing LSTM, K-NN, and Random Forest, the research aims to equip developers, educators, and researchers with insights that can shape the next generation of intelligent, intuitive, and effective educational data. Ultimately, this work seeks to transform the concept map into a predictive, interactive learning tool, where technology supports the learner's journey by recognizing and adapting to their unique path toward knowledge.

2. RESEARCH METHOD

The methodology of this study is divided into several stages, including data collection, data preprocessing, data splitting, model development, and evaluation [26] for each phase can be seen in figure 1. Each phase is described in detail below to provide a comprehensive understanding of the approach taken to analyze user interaction patterns and predict actions within the context of concept map construction.

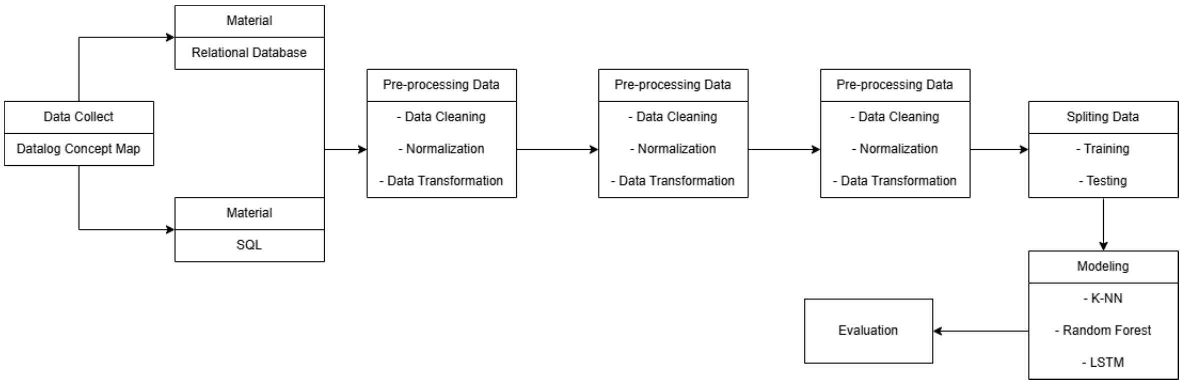

Figure 1. Workflow of the research

2.1. Data Collection

Data collection is the foundational phase where relevant interaction data from users constructing concept maps is gathered [27]. The taken from the datalog of concept map creation by several students in several classes (SQL and database). This include actions such as concept creation, link formation, and modification of content, which collectively represent each user's engagement and behavioral patterns within the concept map environment. The data for this study was gathered from a logging system that recorded users' actions while creating concept maps. The dataset included interaction logs collected from concept map construction activities focused on two materials: relational database and SQL. It consists of 7133 rows of data on database relational material and 8061 rows of data on SQL material. These materials are used because they are structured, allowing for consistent data patterns suitable for predictive modeling. The log data captured multiple aspects of user interactions, such as id, date and time, component creation, action taken, component manufacturing coordinates, along with timestamps and sequence information. This data served as the foundation for analyzing and recognizing patterns in how users construct concept maps over time. Raw data and its attribute's descriptions as shown in Table 1 and 2.

Table 1. Sample of Raw Data

Raw Data								
id	**date**	**time**	**uid**	**comp_type**	**comp_id**	**action**	**locx**	**locy**
25653	11/02/19	8:18:17	658	0	0	0	100	-35
25654	11/02/19	8:18:17	658	0	0	1	542	-283
25655	11/02/19	8:18:20	658	0	0	0	542	-283
25656	11/02/19	8:18:20	658	0	0	1	524	-289
25657	11/02/19	8:18:39	658	1	0	0	300	-35
25658	11/02/19	8:18:39	658	1	0	1	705	-289

Table 2. Description of Atribut

Field Name	Data Type	Description
id	int	A unique identity that becomes the primary key. Formed every time a user performs an action in concept map creation. IDs are sorted in time order.
date	date	the time (date) of each action taken by the user during concept map creation.
time	time	timestamp of each action taken by the user during concept map creation.
uid	int	User ID in concept map creation.
component_type	int	The components in a concept map are divided into three categories: Concept, denoted by number 0; Link, denoted by number 1; and Connector, denoted by number 2.
component_id	int	ID of each component created by the user during concept map creation.
action	int	Each action taken by the user when creating a concept map is divided into 4 categories: Click is denoted by number 0, Move is denoted by number 1, Connect is denoted by number 2, and Disconnect is denoted by number 3.
locx	float	Location of user action on the X-axis
locy	float	Location of user action on the Y-axis

2.2. Data Pre-processing

To ensure the quality and consistency of the data, several preprocessing steps were applied: (1) Data Cleaning: This step involved identifying and removing incomplete, irrelevant, or erroneous records [28]. For instance, logs with missing timestamps or invalid actions (e.g., missing value, unrecorded actions and unnecessary column) were removed to ensure the dataset's integrity; (2) Normalization: To prepare the data for effective machine learning, normalization was applied to numeric features, locx and locy attributes normalized using minmax scaller. This process ensured that all features contributed equally to the model without being biased by their scale [29]; (3) Transformation: Temporal sequences in user interactions were transformed to a structured format that could be utilized effectively by the machine learning models [30]. Specifically, sequential user actions were transformed into time-series data for the LSTM model, it convert data into sequential form to allow the LSTM model to learn sequential patterns and actions were encoded into numerical representations such as one-hot encoding, so that the labels are suitable for multi-class classification [31]. while converts data into a sequential format, were prepared for K-NN and Random Forest. For the random forest forming the data sequence into a one-dimensional vector so that the random forest model can use it as a feature.

2.3. Data Spliting

The preprocessed dataset was divided into training and testing sets to evaluate model performance. A standard 80:20 split was used, with 80% of the data allocated for training and 20% for testing [32]. The training set was used to build the models, while the testing set was reserved for final evaluation to ensure an unbiased assessment of model accuracy [33]. Stratified sampling was employed to maintain the distribution of actions in both the training and testing sets, ensuring that all action types were well-represented and the model's performance could be generalized across various user behaviors.

2.4. Modeling

For both treatments, action prediction and uncovering user interaction pattern three model algorithms were selected for the task:

1. K-NN is a simple, distance-based classification algorithm that predicts user actions by identifying the most similar interaction patterns in the training dataset [34]. It was chosen for its simplicity and interpretability, though its performance can be affected

by the scale of the data and the presence of noise. In this implementation, a K-Nearest Neighbors model with n_neighbors=3 is instantiated. The value of k=3 was likely chosen experimentally, balancing bias and variance [35]. Generally, a lower K can make the model more sensitive to local variations, which may improve classification performance in scenarios with distinct, localized patterns. The distance used in the K-NN algorithm is usually the Euclidean distance. The formula is shown in Equation 1. Where, p and q are two points in an n-dimensional space, p_i and q_i are coordinates of p and q along the i-th dimension, and d (p,q) represents the Euclidean distance between p and q.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{1}$$

2. Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions to achieve robust results. It was selected due to its effectiveness in handling complex, non-linear data and its resistance to overfitting [36]. Random Forest's capability of identifying important features also provided insights into which aspects of user interactions were most influential in predicting actions. In this implementation, the random forest architecture consists of the following parts: (1) Bagging: Each of the 100 trees is trained on a random subset of the training data (sampling with replacement), known as bagging. This approach ensures that each tree sees a slightly different version of the dataset, promoting diversity among trees and making the forest resilient to noisy data; (2) Feature Selection for Each Split: At each split within a decision tree, Random Forest considers a random subset of features rather than all features. This randomness further diversifies the trees, as each tree may focus on a different aspect of the data, which prevents any single feature from dominating the model's decisions; (3) Voting Mechanism: For each input sequence, each tree predicts an action. The final prediction for the action class is made by majority voting, where the class with the most votes from all trees is chosen. This voting strategy improves stability and accuracy, as errors by individual trees tend to cancel out. Majority Voting Formula for Classification is as follows in equation 2. Here, T(X) is the final class prediction, and "mode" refers to the most frequently occurring class among all T_n (x).

$$T(X) = modeT_{1(x)}, T_{2(x)}, \ldots, T_n(x) \tag{2}$$

3. LSTM, a type of Recurrent Neural Network (RNN), is well-suited for sequential data and was employed to capture temporal dependencies in user interactions [37]. The LSTM model can retain information from previous actions, making it highly effective for predicting subsequent actions based on past interactions. Due to its sequential nature, the LSTM model required transformation of the data into time-series format, with sequences of actions as input and the next action as the target variable. The architecture of LSTM shown in Figure 2.

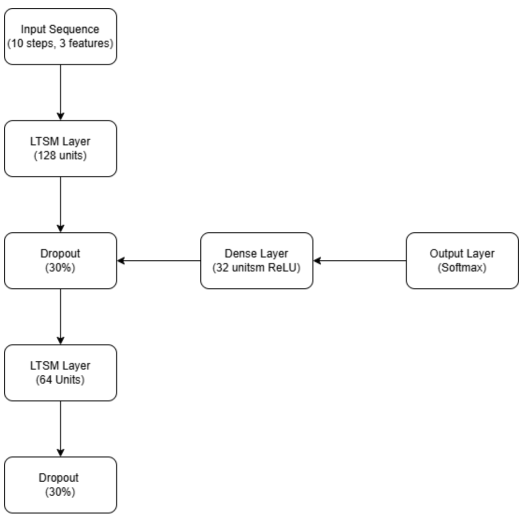

Figure 2. LSTM architecture

2.5. Evaluation

The performance of each model was evaluated using several metrics: accuracy, precision, and recall. These metrics provided a comprehensive assessment of the models' abilities to correctly predict user actions. Accuracy measured the overall correctness of the models' predictions, while precision evaluated the proportion of true positive predictions among all positive predictions, indicating the models' ability to avoid false positives. Recall assessed the proportion of true positive predictions out of all actual positives, indicating the models' ability to capture relevant actions without missing any important patterns [38]. Each metric was calculated for the testing dataset, providing an unbiased evaluation of model performance. Additionally, error analysis was conducted to identify common misclassification patterns and to assess each model's strengths and weaknesses in recognizing complex or subtle user behaviors [39]. The evaluation results informed the selection of the most suitable model for predicting user actions in concept map construction.

3. RESULT AND ANALYSIS

The primary goal of this study was to evaluate and compare the performance of LSTM, K-NN, and Random Forest models in recognizing user interaction patterns and predicting subsequent actions within a concept map construction platform. Concept mapping, as an educational tool, requires capturing and interpreting user interaction data to provide timely and adaptive feedback, making accurate action prediction essential. The purpose of this analysis was twofold: to predict the next action a user might take based on their historical interactions and to detect common patterns that frequently recur in the data. Using interaction logs from concept map construction sessions, each model was trained and evaluated in terms of accuracy, precision, and recall. By examining model performance in accuracy, precision, recall, and computational efficiency, this study seeks to inform best practices in selecting machine learning and deep learning approaches for complex, sequential user interactions in educational settings. The matrix evaluation results of sequential action prediction, interaction pattern recognition and computational efficiency can be seen in Tables 3, 4, 5 and 6.

Table 3. Evaluation of Action Prediction "Database Material"

	Database Material		
	Accuracy	**Precision**	**Recall**
K-NN	70.53%	70.0%	70.1%
Random Forest	80.35%	79.0%	80.4%
LSTM	**83.58%**	**83.6%**	**83.6%**

Table 4. Evaluation of Action Prediction "SQL"

	SQL Material		
	Accuracy	**Precision**	**Recall**
K-NN	73.85%	73.0%	73.9%
Random Forest	82.36%	81.0%	82.4%
LSTM	**83.91%**	**84.3%**	**83.9%**

Table 5. Evaluation of User Pattern Interaction

	Accuracy Table	
	Database	**SQL**
K-NN	80.29%	84.72%
Random Forest	**87.78%**	89.32%
LSTM	87.55%	**89.40%**

Table 6. Computational Efficiency

	Action Prediction		**User Pattern Interaction**	
	Database	**SQL**	**Database**	**SQL**
K-NN	**1s**	**2s**	**4s**	**3s**
Random Forest	4s	6s	5s	5s
LSTM	1m	1m	1m	1m

The results of the analysis reveal distinct strengths and limitations among the three models. The LSTM model achieved the highest accuracy at 83.91%, with a precision of 84.28% and a recall of 83.9%. This superior performance is attributed to LSTM's ability to learn from temporal dependencies, which is particularly relevant for predicting user actions based on sequential interaction data. However, the LSTM model was also the most computationally intensive, resulting in longer training times and higher memory requirements due to its recurrent nature. In contrast, K-NN demonstrated the shortest training time but only reached an accuracy of 70.53%, K-NN's lower performance is likely due to its reliance on distance-based metrics, which may be insufficient for capturing the nuanced, sequential patterns inherent in user interaction data. Random Forest, positioned between LSTM and K-NN, achieved a moderate accuracy of 80.53%. Unlike LSTM, Random Forest does not natively process sequential information, but its ensemble structure allowed it to generalize reasonably well across different action types, balancing predictive power with computational efficiency.

This capability allowed LSTM to excel in scenarios where user actions followed a complex and dependent sequence, which is typical in concept mapping. In comparison, K-NN demonstrated moderate accuracy, performing adequately in cases where recent actions had similarities with past interactions. However, its limitations became evident with longer, more complex sequences, as it lacks an inherent capacity to capture temporal dependencies. Random forests also have good value, but the LSTM works better for this case. While highly effective in certain applications, Random Forest exhibited limitations in sequential prediction due to its approach of treating each action independently; thus, its prediction accuracy was slightly lower in this context. User actions involving complex decision-making or sequences were often incorrectly predicted. The LSTM model, despite its strength in sequence learning, encountered challenges with actions that were sporadic or lacked clear temporal continuity, leading to misclassifications in cases where interaction patterns were sparse or irregular. Computationally, the LSTM model was the most resource-intensive due to its architecture, while Random Forest benefited from parallel processing capabilities, making it faster to train and test. K-NN, while quick in training, requires more computational resources during prediction as it calculates distances to all training points, which may limit its scalability with larger datasets. The analysis results also show a linear relationship between the two materials, "relational database" and "SQL." This linearity indicates a consistent connection between the two, where all the models work linearly on both materials. In practical terms, SQL is the primary language for interacting with relational databases, providing data retrieval, manipulation, and management commands. Relational databases, structured in tables with defined relationships, rely on SQL to execute queries, manage transactions, and ensure data integrity.

Both Random Forest and LSTM models performed well in interaction pattern recognition analysis, showcasing their strengths in identifying recurring patterns in user interaction data. However, when focusing on specific materials, subtle differences emerged in model performance between "relational database" and "SQL" material. Random Forest showed a slight advantage for relational databases, likely due to its ability to capture intricate, non-sequential patterns in user interactions with structured data. Random Forest's approach of evaluating feature importance and handling complex decision boundaries made it particularly effective in recognizing patterns associated with relational database interactions, where actions often depend on hierarchical and relational structures. Conversely, for SQL material, the LSTM model demonstrated marginally better performance. LSTM's strength in handling sequences and capturing dependencies across time steps allowed it to model the progression of SQL operations better, enhancing its predictive accuracy on SQL-specific tasks. While Random Forest excelled slightly more on relational database material and LSTM on SQL material, the difference in accuracy between the two models was insignificant. Both models were highly effective across materials, indicating that either could be reliably used depending on task requirements. However, these small differences suggest that leveraging each model according to its specific strengths could yield optimal results, especially in applications where even minor performance gains are beneficial. But, overall, the deep learning LSTM model performs better.

K-NN is easy to implement and interpret, it has some notable limitations, particularly in the context of interaction pattern recognition. First, K-NN is computationally inefficient, as it requires storing the entire training dataset and calculating distances between the query point and all training samples during prediction. This becomes increasingly costly with larger datasets. Second, K-NN is sensitive to irrelevant features; as all features contribute equally to the distance metric, its performance can degrade when dealing with high-dimensional data or when irrelevant features are present. Most critically, K-NN lacks the ability to consider temporal dynamics, which limits its effectiveness in tasks that rely on understanding the order of user actions. From the best model, the results of patterns that often appear in the construction of concept maps can be seen in Table 7 and in Figure 3 show the distribution of the patterns.

Table 7. Common Pattern Set

Common Pattern	*Frequency*
Move ⟶ Click ⟶ Move ⟶ Click ⟶ Move ⟶ Click ⟶ Move ⟶ Click ⟶ Move ⟶ Click	239
Click ⟶ Move ⟶ Click ⟶ Move ⟶ Click ⟶ Move ⟶ Click ⟶ Move ⟶ Click ⟶ Move	224
Click ⟶ Move ⟶ Click ⟶ Move ⟶ Connect ⟶ Click ⟶ Move ⟶ Connect ⟶ Click ⟶ Move	98
Click ⟶ Move ⟶ Connect ⟶ Click ⟶ Move ⟶ Click ⟶ Move ⟶ Connect ⟶ Click ⟶ Move	85
Move ⟶ Click ⟶ Move ⟶ Click ⟶ Move ⟶ Connect ⟶ Click ⟶ Move ⟶ Connect ⟶ Click	76
Connect ⟶ Click ⟶ Move ⟶ Click ⟶ Move ⟶ Click ⟶ Move ⟶ Connect ⟶ Click ⟶ Move	71
Move ⟶ Click ⟶ Move ⟶ Connect ⟶ Click ⟶ Move ⟶ Click ⟶ Move ⟶ Connect ⟶ Click	67

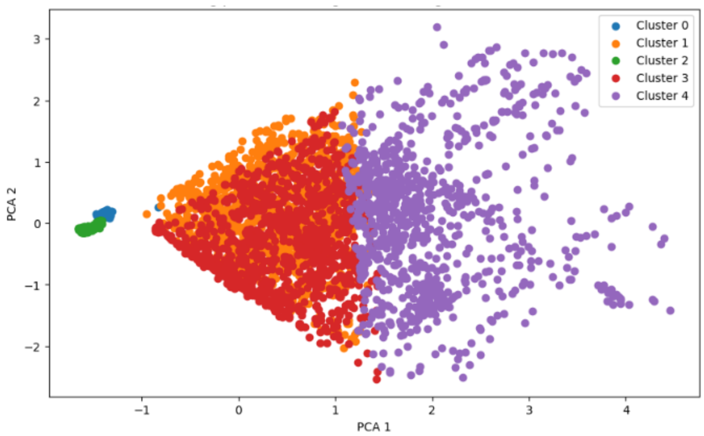

Figure 3. Distribution of User Pattern Interaction

It illustrates a summary of frequent action patterns and a visual representation of clusters derived from user behavior data. It showcases the most commonly occurring sequences of actions among users alongside their respective frequencies. This list of sequences identifies recurring patterns in user activity, offering insights into common pathways or workflows that users follow. The observed patterns and clustering reveal meaningful insights into user behavior, highlighting common action sequences and typical workflows. The most frequent sequences, such as "Click ⟶ Move ⟶ Click ⟶ Move ⟶ Click ⟶ Move ⟶ Click ⟶ Move" and "Move ⟶ Click ⟶ Move ⟶ Click ⟶ Move," suggest that users are often alternating between clicking and moving actions. This repetitive behavior implies that users might navigate through the interface multiple times in a sequence and indicates that the user only frequently moves components without making connections. More complex sequences that incorporate the Connect action, such as "Click ⟶ Move ⟶ Connect ⟶ Move ⟶ Click ⟶ Move ⟶ Connect", suggest that users often engage in actions that involve linking or connecting different elements in the system after repeatedly moving components. This may indicate that the user is confident in making a connection after collecting some components. Interestingly, disconnect actions are less frequent in these common patterns, suggesting that users rarely need to undo connections once they have made them. This may imply that users are confident in their actions. This stability in user workflows is positive as it suggests that users can proceed with tasks without frequent reversal, which often points to a well-structured interface. As shown in the scatter plot, the clustering provides additional context by grouping users with similar behaviors. For example, Cluster 0 appears isolated and sparse, suggesting a smaller group of users with unique behavior patterns, possibly representing an alternative workflow or using specific features. In contrast, Clusters 1 and 2 are denser and centrally located, likely capturing the majority of users who follow the most common, repetitive workflows. This central clustering indicates that many users share similar patterns, which could reflect common paths or workflows within the application. The additional clusters may represent groups with moderate variations, possibly users who blend typical and atypical behaviors.

The findings underscore the practical considerations for selecting models in the context of concept map construction applications. The LSTM model is particularly suitable for applications where action prediction accuracy is paramount, and user interactions exhibit clear temporal patterns. The sequential learning capability of LSTM aligns well with prior research that highlights its effectiveness in handling temporal data. This is linear with some previous research that shows that the LSTM is suitable for sequential data [40–42]. Although LSTM achieved the highest accuracy at 83.91%, its training time takes the longest time compared to K-NN and Random Forest. These limitations arise due to the sequential data processing inherent in its architecture, which contrasts with the faster, less resource-intensive alternatives like Random Forest and K-NN. Future research could explore optimized LSTM variants or hybrid models combining LSTM layers with more efficient algorithms to balance accuracy with computational efficiency. K-NN, in contrast, stands out for its simplicity and minimal training time, making it a viable option for situations where interpretability and rapid training are valued, though at the cost of accuracy. Its dependence on distance-based metrics renders it less effective for complex, temporally dependent interactions, yet this drawback could potentially be mitigated through feature scaling or other pre-processing steps. Random Forest, by comparison, provides a balanced approach with good generalization across varied interaction types and a more favorable computational profile than LSTM. However, its inability to naturally incorporate temporal dependencies suggests that feature engineering, such as the inclusion of temporal variables, might improve its efficacy for action prediction tasks where sequence matters.

Analyzing user behavior patterns can significantly enhance the learning experience by providing personalized and adaptive support. Identifying interaction patterns can help in designing a group based on their understanding of making a concept map. Educators could monitor group progress. This provides a more granular understanding of how students engage with the material, allowing for better formative assessments that help guide the learning process. By analyzing the patterns across a wide group of learners, educators, and instructional designers can gain valuable insights into common difficulties or successful strategies. These insights can inform adjustments to the curriculum or learning tools, creating a more tailored and efficient learning experience for future learners. Predictive models could provide learners with visual progress indicators, showing how their understanding is evolving over time. This could help learners stay motivated and develop a deeper understanding of their learning journey. Traditional assessment models often rely on static tests that may not capture the nuances of a learner's understanding. Predicting user behavior patterns can also allow for dynamic, ongoing assessments that evaluate learners in real-time based on their actions. This assessment could be used to generate more holistic, continuous measures of learning progress. For the user interface, predicting the actions of novice learners can help design systems that automatically introduce expert-level strategies. If a novice struggles with the same repetitive actions, a system could recognize this pattern and suggest more advanced or efficient ways to structure their concept map. The patterns of behavior could be used to create personalized learning paths. For instance, if a user consistently follows certain patterns that reflect limited understanding, the system could adapt the content or the structure of tasks to provide additional support, explanations, or scaffolded tasks aimed at addressing knowledge gaps.

4. CONCLUSION

In conclusion, the LSTM model appears most suitable for high-accuracy action prediction and interaction pattern recognition in concept map applications where sequential user interactions are central to the task. LSTM was most effective for action prediction due to its ability to model sequences, making it ideal for environments where the order of user actions is significant. On the other hand, for environments constrained by computational resources or where efficiency is prioritized, Random Forest offers a practical alternative with respectable accuracy and speed. Random Forest also proved invaluable for identifying recurring interaction patterns, enabling a deeper understanding of user behaviors that commonly appear in concept map construction. Although K-NN performed with lower accuracy, its simplicity and speed make it a feasible choice for basic implementations or as a component in ensemble methods. While useful in simpler scenarios, K-NN was less suited to handling complex interaction patterns. Future studies should investigate hybrid models' potential to combine LSTM's temporal learning capabilities with Random Forest's generalization strengths, providing a balanced approach for real-time action prediction. Additionally, employing data augmentation and tailored feature engineering could further enhance the performance of non-sequential models, enabling them to capture complex, interaction-based user behaviors better and support adaptive feedback in concept map construction platforms.

5. ACKNOWLEDGEMENTS

The Acknowledgments section is optional. Research sources can be included in this section.

6. DECLARATIONS

AUTHOR CONTIBUTION

F.ti Ayyu Sayyidul Laily, the first author responsible for conceptualizing the research, designing the methodology, and drafting the manuscript. The second author, Didik Dwi Prasetya focuses on data collection, analysis, and interpretation, ensuring the research aligns with the objectives. Author 3, Anik Nur Handayani, provides technical expertise, develops models or tools, and reviews the statistical or computational methods. Author 4, Senior Advisor, Tsukasa Hirashima, offers guidance on framing the research, reviews the manuscript critically, and ensures it meets academic standards for publication.

FUNDING STATEMENT
This study was conducted without any financial support.

COMPETING INTEREST
The authors declare no conflict of interest regarding the publication of this article.

REFERENCES

[1] S. Chunyan, N. Agarwal, and A. Islam, "Concept Mapping as Problem-Based Learning: Assessing its Effectiveness in Teaching-Learning Processes," *International Journal of Innovative Science and Research Technology (IJISRT)*, vol. 9, no. 5, pp. 3694–3699, jul 2024, https://doi.org/10.38124/ijisrt/IJISRT24MAY2484.

[2] B. K. Dahal, "Concept Mapping: Unleashing the Power of Visual Learning in Science Education," *Ganeshman Darpan*, vol. 8, no. 1, pp. 35–43, 2023, https://doi.org/10.3126/gd.v8i1.57330.

[3] D. D. Prasetya, A. Pinandito, Y. Hayashi, and T. Hirashima, "Analysis of quality of knowledge structure and students' perceptions in extension concept mapping," *Research and Practice in Technology Enhanced Learning*, vol. 17, no. 1, pp. 1–25, 2022, https://doi.org/10.1186/s41039-022-00189-9.

[4] "Editorial: Concept mapping: Improving learning and understanding," *Knowledge Management & E-Learning: An International Journal*, vol. 15, no. 3, pp. 365–368, sep 2023, https://doi.org/10.34105/j.kmel.2023.15.020.

[5] A. Pinandito, C. P. Wulandari, D. D. Prasetya, Y. Hayashi, and T. Hirashima, "Knowledge Reconstruction with Kit-Build Concept Map: A Review from Student Experience," in *7th International Conference on Sustainable Information Engineering and Technology 2022*. New York: ACM, nov 2022, pp. 263–270, https://doi.org/10.1145/3568231.3568274.

[6] D. D. Prasetya and T. Hirashima, "Associated Patterns in Open-Ended Concept Maps within E-Learning," *Knowledge Engineering and Data Science*, vol. 5, no. 2, p. 179, 2022, https://doi.org/10.17977/um018v5i22022p179-187.

[7] K. E. de Ries, H. Schaap, A. M. M. van Loon, M. M. Kral, and P. C. Meijer, "A literature review of open-ended concept maps as a research instrument to study knowledge and learning," vol. 56, no. 1, pp. 73–107, 2022, https://doi.org/10.1007/s11135-021-01113-x.

[8] P. G. Fonteles Furtado, T. Hirashima, and Y. Hayashi, "The effect on new knowledge and reviewed knowledge caused by the positioning task in closed concept maps," *Research and Practice in Technology Enhanced Learning*, vol. 14, no. 1, p. 15, dec 2019, https://doi.org/10.1186/s41039-019-0108-1.

[9] Enitan Shukurat Animashaun, Babajide Tolulope Familoni, and Nneamaka Chisom Onyebuchi, "Advanced machine learning techniques for personalising technology education," *Computer Science & IT Research Journal*, vol. 5, no. 6, pp. 1300–1313, jun 2024, https://doi.org/10.51594/csitrj.v5i6.1198.

[10] B. Alnasyan, M. Basheri, and M. Alassafi, "The power of Deep Learning techniques for predicting student performance in Virtual Lear ning Environments: A systematic literature review," *Computers and Education: Artificial Intelligence*, vol. 6, no. 2024, p. 100231, jun 2024, https://doi.org/10.1016/j.caeai.2024.100231.

[11] J. King, R. Lemmens, and E.-W. Augustijn, "Enhancing Large Educational Node-Link Concept Map Design for Optimal User Learning," *Abstracts of the ICA*, vol. 5, no. September, pp. 1–2, 2022, https://doi.org/10.5194/ica-abs-5-23-2022.

[12] N. Islam, H. Noor, and D. M. Farid, "A Novel Ensemble K-Nearest Neighbours Classifier with Attribute Bagging," *Communications in Computer and Information Science*, vol. 1950 CCIS, no. October, pp. 262–276, 2024, https://doi.org/10.1007/978-981-99-7666-9_22.

[13] P. Srisuradetchai and K. Suksrikran, "Random kernel k-nearest neighbors regression," *Frontiers in Big Data*, vol. 7, no. 2021, pp. 1–14, jul 2024, https://doi.org/10.3389/fdata.2024.1402384.

[14] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest Algorithm Overview," *Babylonian Journal of Machine Learning*, vol. 2024, no. 1, pp. 69–79, jun 2024, https://doi.org/10.58496/BJML/2024/007.

[15] J. Luan, C. Zhang, B. Xu, Y. Xue, and Y. Ren, "The predictive performances of random forest models with limited sample size and different species traits," *Fisheries Research*, vol. 227, no. July 2020, p. 105534, jul 2020, https://doi.org/10.1016/j.fishres.2020.105534.

[16] I. Sonata and Y. Heryadi, "Comparison of LSTM and Transformer for Time Series Data Forecasting," in *2024 7th International Conference on Informatics and Computational Sciences (ICICoS)*. IEEE, jul 2024, pp. 491–495, https://doi.org/10.1109/ICICoS62600.2024.10636892.

[17] A. Isah, H. Shin, S. Oh, S. Oh, I. Aliyu, T. W. Um, and J. Kim, "Digital Twins Temporal Dependencies-Based on Time Series Using Multivariate Long Short-Term Memory," *Electronics (Switzerland)*, vol. 12, no. 19, pp. 1–15, 2023, https://doi.org/10.3390/electronics12194187.

[18] V. B and R. Gangula, "Exploring the Power and Practical Applications of K-Nearest Neighbours (KNN) in Machine Learning," *Journal of Computer Allied Intelligence*, vol. 2, no. 1, pp. 8–15, 2024, https://doi.org/10.69996/jcai.2024002.

[19] E. Cantor, S. Guauque-Olarte, R. León, S. Chabert, and R. Salas, "Knowledge-slanted random forest method for high-dimensional data and small sample size with a feature selection application for gene expression data," *BioData Mining*, vol. 17, no. 1, p. 34, sep 2024, https://doi.org/10.1186/s13040-024-00388-8.

[20] A. S. Aljaloud, D. M. Uliyan, A. Alkhalil, M. A. Elrhman, A. F. M. Alogali, Y. M. Altameemi, M. Altamimi, and P. Kwan, "A Deep Learning Model to Predict Student Learning Outcomes in LMS Using CNN and LSTM," *IEEE Access*, vol. 10, no. July, pp. 85 255–85 265, 2022, https://doi.org/10.1109/ACCESS.2022.3196784.

[21] M. Jawthari and V. Stoffová, "Predicting students' academic performance using a modified kNN algorithm," *Pollack Periodica*, vol. 16, no. 3, pp. 20–26, 2021, https://doi.org/10.1556/606.2021.00374.

[22] L. H. Alamri, R. S. Almuslim, M. S. Alotibi, D. K. Alkadi, I. Ullah Khan, and N. Aslam, "Predicting Student Academic Performance using Support Vector Machine and Random Forest," *ACM International Conference Proceeding Series*, vol. PartF16898, no. July, pp. 100–107, 2020, https://doi.org/10.1145/3446590.3446607.

[23] H. N. Bhandari, B. Rimal, N. R. Pokhrel, R. Rimal, and K. R. Dahal, "LSTM-SDM: An integrated framework of LSTM implementation for sequential data modeling[Formula presented]," *Software Impacts*, vol. 14, no. July, p. 100396, 2022, https://doi.org/10.1016/j.simpa.2022.100396.

[24] S. Odoom and E. O. Osei, "An Optimal Random Forest Model for Enhancing Decision-Making in Improving Students' Performance via Educational Data Mining," *International Journal For Multidisciplinary Research*, vol. 6, no. 1, pp. 1–12, 2024, https://doi.org/10.36948/ijfmr.2024.v06i01.11584.

[25] D. Roy, A. Bhaduri, K. Maji, M. Mal, J. Patra, and S. Gupta, "Improving Alexa's response with sentiment analysis and comparing various machine learning models," in *2024 International Conference on Circuit, Systems and Communication (ICCSC)*. IEEE, jun 2024, pp. 1–6, https://doi.org/10.1109/ICCSC62074.2024.10617303.

[26] F. Knapp and M. Šimon, "Methodology for the Development of Production Systems in the Automotive Industry," *Tehnicki Glasnik*, vol. 18, no. 3, pp. 400–409, 2024, https://doi.org/10.31803/tg-20240502085916.

[27] J. G. B. Derraik, W. Parklak, B. B. Albert, K. Boonyapranai, and K. Rerkasem, "Fundamentals of Data Collection in Clinical Studies: Simple Steps to Avoid "Garbage In, Garbage Out"," *The International Journal of Lower Extremity Wounds*, vol. 20, no. 3, pp. 183–187, sep 2021, https://doi.org/10.1177/1534734620938234.

[28] M. A. Oladipupo, P. C. Obuzor, B. J. Bamgbade, A. E. Adeniyi, K. M. Olagunju, and S. A. Ajagbe, "An Automated Python Script for Data Cleaning and Labeling using Machine Learning Technique," *Informatica*, vol. 47, no. 6, pp. 219–232, jun 2023, https://doi.org/10.31449/inf.v47i6.4474.

[29] K. Mahmud Sujon, R. Binti Hassan, Z. Tusnia Towshi, M. A. Othman, M. Abdus Samad, and K. Choi, "When to Use Standardization and Normalization: Empirical Evidence From Machine Learning Models and XAI," *IEEE Access*, vol. 12, pp. 135 300–135 314, 2024, https://doi.org/10.1109/ACCESS.2024.3462434.

[30] A. Carrera-Rivera, D. Reguera-Bakhache, F. Larrinaga, G. Lasa, and I. Garitano, "Structured dataset of human-machine interactions enabling adaptive user interfaces," *Scientific Data*, vol. 10, no. 1, pp. 1–9, 2023, https://doi.org/10.1038/s41597-023-02741-8.

[31] F. Carrara, P. Elias, J. Sedmidubsky, and P. Zezula, "LSTM-based real-time action detection and prediction in human motion streams," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27 309–27 331, 2019, https://doi.org/10.1007/s11042-019-07827-3.

[32] B. Basnayake and N. Chandrasekara, "Assessing the Performance of Feedforward Neural Network Models with Random Data Split for Time Series Data: A Simulation Study," in *2024 International Research Conference on Smart Computing and Systems Engineering (SCSE)*. IEEE, apr 2024, pp. 1–6, https://doi.org/10.1109/SCSE61872.2024.10550735.

[33] J. Allgaier and R. Pryss, "Practical approaches in evaluating validation and biases of machine learning applied to mobile health studies," *Communications Medicine*, vol. 4, no. 1, pp. 1–11, 2024, https://doi.org/10.1038/s43856-024-00468-0.

[34] H. AbdelAzim, M. Tharwat, and A. Mohammed, "Efficient Computational Cost Reduction in KNN through Maximum Entropy Clustering," in *2024 6th International Conference on Computing and Informatics (ICCI)*. IEEE, mar 2024, pp. 34–38, https://doi.org/10.1109/ICCI61671.2024.10485066.

[35] N. Hidayati and A. Hermawan, "K-Nearest Neighbor (K-NN) algorithm with Euclidean and Manhattan in classification of student graduation," *Journal of Engineering and Applied Technology*, vol. 2, no. 2, pp. 86–91, 2021, https://doi.org/10.21831/jeatech.v2i2.42777.

[36] L. Barreñada, P. Dhiman, D. Timmerman, A.-L. Boulesteix, and B. Van Calster, "Understanding overfitting in random forest for probability estimation: a visualization and simulation study," *Diagnostic and Prognostic Research*, vol. 8, no. 1, p. 14, sep 2024, https://doi.org/10.1186/s41512-024-00177-1.

[37] H. Agarwal, G. Mahajan, A. Shrotriya, and D. Shekhawat, "Predictive Data Analysis: Leveraging RNN and LSTM Techniques for Time Series Dataset," *Procedia Computer Science*, vol. 235, no. 2024, pp. 979–989, 2024, https://doi.org/10.1016/j.procs.2024.04.093.

[38] V. J. Raja, D. M, G. Solaimalai, D. L. Rani, P. Deepa, and R. G. Vidhya, "Machine Learning Revolutionizing Performance Evaluation: Recent Developments and Breakthroughs," in *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*. IEEE, jul 2024, pp. 780–785, https://doi.org/10.1109/ICSCSS60660.2024.10625103.

[39] S. Geng, "Analysis of the Different Statistical Metrics in Machine Learning," *Highlights in Science, Engineering and Technology*, vol. 88, no. 2024, pp. 350–356, 2024, https://doi.org/10.54097/jhq3tv19.

[40] A. Faheem and U. Arshad, "Harnessing Long Short Term Memory Networks for Stock Market Forecasts," in *2024 International Conference on Engineering & Computing Technologies (ICECT)*. IEEE, may 2024, pp. 1–6, https://doi.org/10.1109/ICECT61618.2024.10581047.

[41] K. N. Simani, Y. O. Genga, and Y.-C. J. Yen, "Using LSTM to Perform Load Predictions for Grid-Interactive Buildings," *SAIEE Africa Research Journal*, vol. 115, no. 2, pp. 42–47, 2024, https://doi.org/10.23919/saiee.2024.10520212.

[42] X. Huang, X. Li, and W. Wang, "Temporal data-driven short-term traffic prediction: Application and analysis of LSTM model," *Theoretical and Natural Science*, vol. 14, no. 1, pp. 205–211, 2023, https://doi.org/10.54254/2753-8818/14/20240948.