



An Interpretable Topic-Aware Deep Learning Framework for User Experience Analysis in Electronic Wallet Reviews

Albertus Dwiyoğa Widianoro^{*}, Bernardinus Harnadi^{}, Erdhi Wdiyanto Nugroho^{}

Information System Department, Soegijapranata Catholic University, Semarang 50234, Indonesia

Corresponding Author Email: yoga@unika.ac.id

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.310916>

ABSTRACT

Received: 9 June 2026

Revised: 25 August 2026

Accepted: 4 September 2026

Available online: 30 September 2026

Keywords:

electronic wallet reviews, topic modeling, non-negative matrix factorization, interpretable deep learning, convolutional neural network-long short-term memory, fintech analytics

The rapid adoption of fintech-based electronic wallets (e-wallets) has generated a large volume of user-generated reviews containing valuable information about service quality, usability, and operational issues. However, extracting actionable insights from these unstructured texts remains challenging because conventional topic modeling (TM) methods provide semantic interpretation without predictive capability, whereas deep learning (DL) models often achieve high accuracy with limited explainability. This study proposes an interpretable topic-aware DL framework that integrates non-negative matrix factorization (NMF), generative artificial intelligence (AI)-assisted semantic interpretation, and a convolutional neural network-long short-term memory (CNN-LSTM) classifier with Random Over-Sampling with Neighborhood Cleaning (ROS-NCL)-based imbalance handling for e-wallet review analysis. A dataset containing 42,680 user reviews collected from eight major e-wallet platforms in Indonesia during 2023–2024 was used for evaluation. NMF was first employed to identify latent topics, while generative AI was used to generate topic descriptions that were further validated by domain experts. The interpreted topics were then used for document labeling and classification using the proposed DL model. Experimental results show that the proposed framework achieved 96.26% accuracy, with macro-average precision, recall, and F1-score of 0.96, 0.94, and 0.95, respectively. The framework effectively integrates semantic interpretability and predictive performance while improving classification robustness under imbalanced topic distributions. The findings demonstrate its potential for supporting data-driven analysis of user feedback in fintech applications.

1. INTRODUCTION

Financial technology (Fintech) is currently widely used throughout society. Fintech developers are actively developing their products according to their respective characteristics. Electronic wallets (e-wallets), as a payment method integrated with banks, facilitate their use. The quality of e-wallet service is paramount to its continued use. One way to achieve this is by examining user feedback within the e-wallet application.

Consumer reviews within the application significantly influence decision-making [1]. Therefore, companies utilize online consumer reviews to understand consumer demand, the dynamic state of technology, capitalize on commercial opportunities, and promote product innovation [2]. The importance of reviews in Fintech stems from the growing interest in gaining value from user-generated reviews for businesses. These reviews play a crucial role in increasing intention to adopt mobile banking [3]. Because user reviews are text-based and unstructured, and the sheer volume of comments generated daily is so large, user reviews represent a critical analysis challenge for application sustainability. Without the right analytical tools, the data can be rendered meaningless [4].

Automated analysis is crucial when dealing with large volumes of data. As data volumes grow, it's easy to discover correlations that may not be causally related. The right balance is a collaborative approach, where automated mechanisms assist humans in extracting and interpreting useful information [5]. This study used data from comments from e-wallet users in Indonesia, consisting of eight e-wallet developers with millions of users.

A text processing mechanism transforms a collection of comments into a structured set of information that is then analyzed using various methods. Keyword and topic extraction mechanisms enable intelligent document analysis by classifying documents according to their semantic categories. Topic extraction, a dimensionality reduction mechanism designed to represent data in a smaller space, plays a key role in managing, understanding, and visualizing data.

Non-negative matrix factorization (NMF) differs from other traditional dimensionality reduction algorithms because it uncovers the low-dimensional latent structure inherent in high-dimensional data and provides a non-negativity-based data representation, thus enhancing the meaningful interpretation of the mined information [6].

NMF optimization studies have investigated initialization strategies to improve matrix factorization performance,

although the approaches were evaluated in other domains. The ease of understanding of the results obtained from NMF motivates its success in several areas of social phenomenon analysis [7], omic biology data [8], and model prompting [9].

Various studies have demonstrated the effectiveness of combining convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, including in the context of text analysis. The study [10] developed a character-based convolutional network for text classification and demonstrated impressive results in understanding spatial features of text. Kim [11] used CNNs for sentence classification and found that this model excelled in capturing local features of text. Yin et al. [12] compared CNNs and recurrent neural networks (RNNs) (including LSTMs) for natural language processing and found that the combination of the two yielded better results than either model alone. A hybrid CNN–LSTM model for recognizing Parkinson's disease from linguistic features using Mel spectrograms derived from normalized speech signals and dynamic mode decomposition yielded high accuracy [13]. A CNN–LSTM study for malware classification based on Application Programming Interface (API) calls and opcodes to improve classification accuracy yielded very high accuracy [14]. Deep learning (DL) is accurate, but black-box, so the necessary interpretability vs. accuracy trade-off has not been resolved.

Various studies have demonstrated the effectiveness of combining CNN–LSTM architectures for text classification tasks. In the analysis of user-generated content, CNN–LSTM models have been applied to capture local textual patterns and sequential dependencies within app reviews. Recent research on large-scale app reviews indicates that CNN–LSTM models can effectively classify user sentiment and emotions by learning semantic patterns from textual feedback [15]. Furthermore, studies in the fintech sector show that analyzing mobile banking and digital wallet reviews can reveal critical aspects of the user experience, including usability, service quality, and operational issues [16].

This research develops an integration framework that links topic modeling (TM)-based interpretation with the predictive capabilities of DL. The study's novelty lies in a "topic-aware learning" integration mechanism designed to enhance interpretability and classification performance on imbalanced fintech review data, encompassing three aspects:

1. Methodological contribution: A proposed topic-based DL framework bridges the interpretability of NMF-based TM with the predictive power of a combined CNN–LSTM algorithm on imbalanced data.

2. Empirical contribution: A dataset of user reviews from eight major fintech e-wallets in Indonesia demonstrates that the proposed framework achieves high classification performance with high topic semantics.

3. Practical contribution: The topic extraction and classification results yield in-depth insights that can be optimized for e-wallet improvement. Topic-level semantic interpretability, specifically, the framework's ability to generate easily understandable semantic representations, is achieved through word distributions within each topic derived from NMF, semantic interpretation using generative artificial intelligence (AI), and validation of the interpretation results by domain experts.

In analyzing e-wallet user comments, a methodological gap exists: the NMF approach produces topics that can be interpreted descriptively, but are not optimized in automated processes by integrating with other algorithms. DL

classification alone (CNN–LSTM) can predict with high accuracy, but it behaves like a black box, failing to explain what aspects are important. In e-wallets, interpretation is crucial for action. Imbalanced data is often ignored or handled haphazardly, thus failing to detect important issues. The combined NMF–CNN–LSTM approach produces a more actionable topic-aware view, not just a pipeline.

Thus, the research gap is as follows: TM offers interpretability but lacks predictive capability; DL provides accuracy but lacks semantic transparency; and imbalanced reviews hinder the detection of minority issues. The proposed framework aims to bridge these three issues.

This paper is structured as follows: Section 2 introduces some concepts related to the model used to translate tweets from their unstructured form into tweet-term matrices and NMF. The CNN–LSTM algorithm is used for classification. Section 3 describes the methods used in constructing the proposed framework. Section 4 presents the results and discussion for a detailed presentation of a case study. Section 5 concludes.

2. RELATED WORK

TM, or topic detection, is an automated tool for discovering topics from textual data. This process is one of the important steps in understanding and organizing unstructured textual data. These topics are implicitly represented by a set of words. Given a set of textual data, the topic detection problem is how to find these sets of words [17]. Currently, there are several methods that can be used to extract topics in a set of textual data, for example, NMF [18] and latent Dirichlet allocation (LDA) [19]. NMF is a matrix factorization method by factoring matrix A into two matrices W and H , all of whose elements are non-negative. If A is a word-per-data matrix, then W is interpreted as a word-per-topic matrix. In other words, the columns of W are sets of words interpreted as topics. Each topic is usually represented only by the most frequently occurring words. Several simulations have shown that NMF provides better accuracy than other methods [20]. TM (LDA and NMF) is very effective for topic exploration, but is descriptive and not designed for continuous automatic classification of new data.

Most research in topic detection focuses on its application in real-life applications, for example, detecting topics in Urdu on Twitter [21] and monitoring trending topics on Instagram [22]. Most studies stop at topic interpretation, without bridging the results to an operational predictive system.

Another major application is using topics for news article classification and clustering [23]. Other papers place topics as features. Analysis of fintech features has been carried out using topic LDA models [24, 25]. In the study [26], the focus is on sentiment classification and analyzing accuracy using two-class and three-class sentiment analysis datasets. TM approaches offer interpretability, while DL models offer accuracy, but the literature tends to treat the two separately. This results in fintech analytics systems often facing a dilemma between explainability and performance, without a unified framework that balances the two.

NMF aims to approximate a matrix V (data matrix) by the multiplication of two non-negative matrices, W and H . In the context of topic processing, V is a document-word matrix formed from a set of text documents, where rows are documents and columns are words or features [27]. NMF is

structurally robust, but is not designed for end-to-end classification. The basic formula of NMF is as follows:

$$V \approx W \times H \quad (1)$$

where,

$V \in R^{m \times n}$ is the document-word matrix with dimension $m \times n$ (number of documents m and number of words n).

$W \in R^{m \times k}$ is the basis matrix, where each column represents the distribution of words from one topic. Dimension $m \times k$, with k as the number of topics.

$H \in R^{k \times n}$ is the topic-document representation matrix, where each row shows the weight or importance of a topic in a document. Dimension $k \times n$, with k as the number of topics.

2.1 Non-negative matrix factorization optimization

The NMF optimization process is to find the values of W and H that minimize the error function (cost function), which is formulated as the Frobenius norm of the difference between V and WH (Eq. (2)):

$$\min_{W, H \geq 0} \|V - WH\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n (V_{ij} - (WH)_{ij})^2 \quad (2)$$

where,

$\|\cdot\|_F^2$ is the Frobenius norm, which computes the square root of the sum of all elements in the matrix.

The non-negativity condition $W, H \geq 0$ ensures that all elements in matrices W and H remain non-negative.

2.2 Update rules for non-negative matrix factorization

NMF is solved using iterative multiplicative update rules. The update rules for W and H are as follows:

Updates for the matrix H :

$$H = H \odot \frac{W^T V}{W^T W H} \quad (3)$$

Updates for the matrix W :

$$W = W \odot \frac{V H^T}{W H H^T} \quad (4)$$

where,

$\frac{A}{B}$ is the element-by-element division between two matrices A and B .

2.3 Topic processing with non-negative matrix factorization

The V matrix is generated from the document representation (using Term Frequency–Inverse Document Frequency (TF-IDF)). The V matrix is decomposed into W and H matrices. The W matrix gives the distribution of words in topics, where each column represents one topic, and the H matrix shows the importance of topics in the document. Once the W and H matrices are obtained, we can extract topics by identifying the words with the largest weights in the W columns.

This CNN–LSTM model combines the ability of CNN to capture local features of text and LSTM to capture temporal or

sequence dependencies in text, which are then classified using a Dense layer with softmax. Here is the mathematical representation of CNN–LSTM in the given code:

2.3.1 Embedding layer

The embedding layer maps tokens or words into fixed-dimensional vectors.

X is an input token with sequence length T . $W_e \in R^{v \times d}$ is the embedding matrix, where V is the vocabulary size and $d = 256$ is the embedding dimension. Then the embedding of each word can be written as:

$$E(X) = W_e \cdot X \quad (5)$$

where, X is a sequence of words in one-hot vector form, and the result is a dense representation $E(X) \in R^{T \times d}$.

2.3.2 Conv1D layer (CNN)

The convolutional layer captures local features from the input embedding. The convolution operation can be formulated as:

$$F(t) = \text{ReLU} \left(\sum_{i=1}^k W_{\text{conv}}(i) \cdot E(t + i - 1) + b_{\text{conv}} \right) \quad (6)$$

where,

$W_{\text{conv}} \in R^{k \times d \times f}$ is the convolution kernel with $k = 5$ as the filter size and $f = 128$ as the number of filters. $E(t)$ is the input embedding at timestep t . b_{conv} is the bias. The ReLU activation function is applied elementwise.

2.3.3 MaxPooling layer

MaxPooling layer performs pooling to reduce dimensionality and extract the most important features.

$$P(t) = \max(F(t:t + p - 1))$$

where, p is the pooling window size, in this case 2.

2.3.4 Dropout

Dropout is a regularization technique used to prevent overfitting by ignoring some neurons during training with a certain probability.

$$D(t) = P(t) \cdot m(t)$$

where, $m(t)$ is a binary mask that has a value of 1 with probability 0.3 (dropout rate), and 0 otherwise.

2.3.5 Spatial dropout

Spatial dropout is similar to regular dropout, but dropout is applied spatially to the entire feature vector in a timestep, rather than individually to neurons. Where r_t is a binary mask applied over the entire timestep:

$$S(t) = D(t) \cdot r_t$$

by randomization with a dropout probability of 0.3.

2.3.6 Bidirectional LSTM

Bidirectional LSTM is used to capture temporal information from two directions: forward and backward. For LSTM, the basic operation at each timestep t can be formulated as:

Forget gate: $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$,
Input gate: $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$,
Candidate cell state: $\hat{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$,
Output gate: $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$,
Cell state update: $C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t$,
Hidden state update: $h_t = o_t \odot \tanh(C_t)$.

For Bidirectional LSTM, there are two calculation sequences: one for the forward direction and one for the backward direction. The output is combined:

$$h_t^{bi} = [h_t^{forward}, h_t^{backward}] \quad (7)$$

There are two LSTM layers, where the first LSTM returns the complete sequence, and the second LSTM returns the final output of each sequence.

2.3.7 Fully connected layer (dense layer)

The Dense layer converts the LSTM output into class output by using the softmax function for classification:

$$\begin{aligned} z &= W_d \cdot h + b_d \\ y &= \text{softmax}(z) \end{aligned} \quad (8)$$

where, W_d is the weight, b_d is the bias, and z is the linear score for each class.

2.3.8 Loss function (categorical crossentropy)

The loss function is used to measure the error in predicting the probability of the actual class:

$$\mathcal{L} = \sum_{c=1}^C y_c \log(\hat{y}_c) \quad (9)$$

where,

$\mathcal{L}$ is the loss function, in this case categorical cross-entropy.

$\sum_{c=1}^C y_c \log(\hat{y}_c)$ is the sum for each class c from 1 to C , where

C is the total number of classes.

y_c is the actual label for class c . Its value is 1 if the true class is c , and 0 otherwise.

$\hat{y}_c$ is the predicted probability of the model for class c , which is the output of the softmax function.

This function calculates how much the predicted label y_c differs from the true label y_c for all classes, and then penalizes it based on the logarithm of the predicted probability. If the predicted y_c is close to 1 for the true class, then the value of $\log(\hat{y}_c)$ will be large and negative, smaller (i.e., a smaller loss penalty). Conversely, if the predicted y_c is close to 0 for the true class, then $\log(\hat{y}_c)$ will be large in negative, and a larger loss penalty.

Random Over-Sampling with Neighborhood Cleaning (ROS-NCL) is used to improve robustness and avoid bias towards important minority issues. ROS-NCL is responsible for balancing without creating artificial linguistic patterns and maintaining the integrity of text representation. Although various sampling techniques exist to address data imbalance, ROS is the preferred method for handling imbalanced data [28]. The majority class has been shown to be susceptible to misrepresentation, demonstrating that each dataset is uniquely sensitive to imbalance, a problem that can be solved using ROS [29]. A combination of oversampling and undersampling methods can be one solution in resolving imbalance cases [30]. The undersampling method used is Neighborhood Cleaning

Rules. NCL can improve the identification of minority classes that are difficult for classifiers to recognize [31].

A unified framework that is both interpretable and robust does not yet exist. TM approaches provide an interpretable thematic structure but remain largely descriptive and inadequate for scalable predictive applications. Machine learning (ML) models, while highly accurate, often lack semantic transparency and offer limited insight into the specific dimensions of user experience that drive classification results. Furthermore, the inherent class imbalance in fintech reviews introduces bias toward majority topics, reducing robustness in detecting minority but high-risk issues. Therefore, a unified framework that integrates interpretability, predictive power, and imbalance-aware learning remains underexplored.

3. METHODS

Figure 1 illustrates the workflow of an analytical model that integrates NMF TM techniques, generative AI, and CNN-LSTM-based classification to generate high-accuracy predictions on text. This system also handles imbalanced data using the ROS-NCL method. The main objective of this research is to improve the efficiency of managing imbalanced text data and to improve accuracy through the integration of these methods.

3.1 Non-negative matrix factorization for topic extraction

In the first step, NMF is used, an effective matrix decomposition technique for discovering hidden topics in text documents. NMF aims to factor the document-word matrix into two non-negative matrices representing the topic distribution and the word distribution for each topic. In this context, NMF is used to generate eight topics from a collection of documents. Topic coherence, a measure of how meaningful a topic is, is calculated to ensure the quality of the generated topics.

3.2 Topic interpretation using generative AI

After topics are generated from the NMF, the next step involves using generative AI to assist in interpreting the topics. Generative AI processes the information from the generated topics and provides a deeper definition or understanding of each topic. The output of this process is a semantic representation of eight topics that can be further used in data labeling. Use the following prompt with ChatGPT: Analyze the keywords obtained from NMF TM of user reviews for digital wallets (e-wallets) in Indonesia. Identify the key aspects of comments and user experiences represented by these keywords, and provide a concise topic label along with an explanation.

3.3 Document labeling based on topics

In the third step, the topics generated and interpreted by generative AI are used to label documents. Each document is identified and labeled according to the most relevant topic. This process produces labeled documents based on the defined topic categories, which will be used as input for further model training. Furthermore, more in-depth features are also determined based on the interpretation results from generative AI.

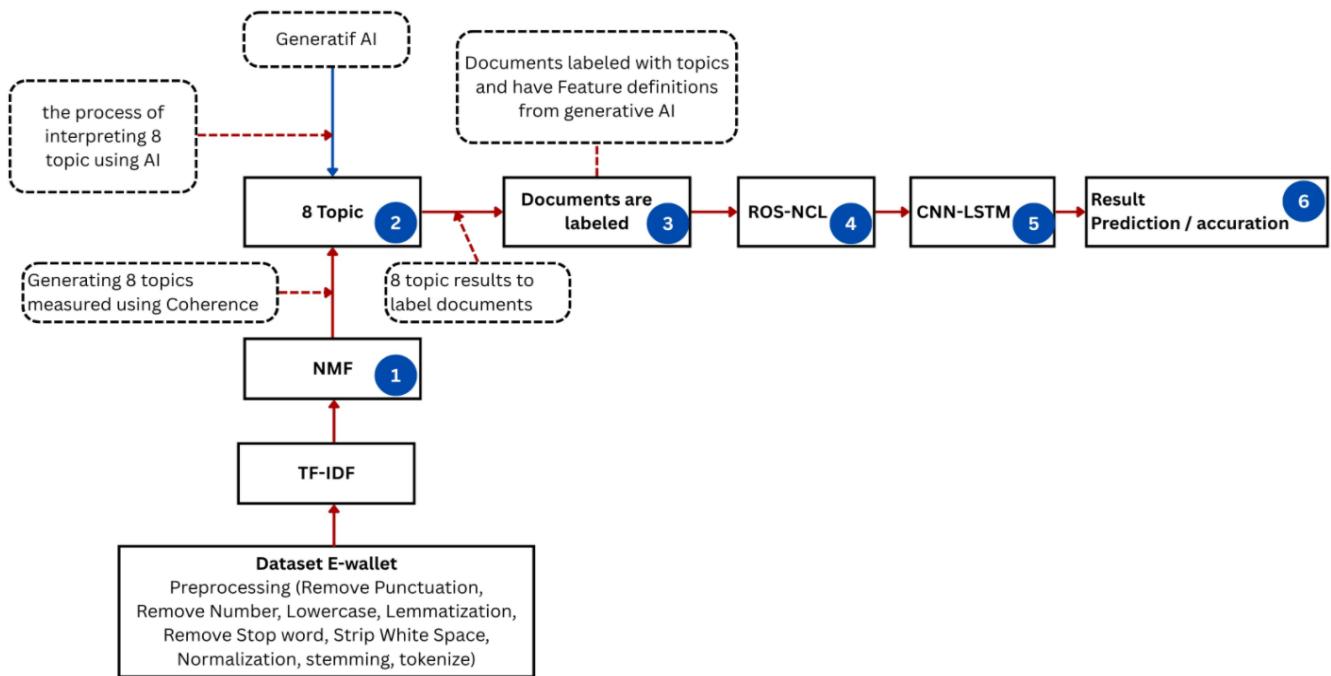


Figure 1. Hybrid topic-deep learning (DL) framework

Note: CNN-LSTM = Convolutional neural network–long short-term memory, ROS-NCL = Random Over-Sampling with Neighborhood Cleaning, TF-IDF = Term Frequency–Inverse Document Frequency, NMF = non-negative matrix factorization.

Each document is assigned a single dominant topic label (single-label classification); the labeling is based on the topic probability distribution derived from NMF, where the topic with the highest contribution value is selected as the document label. For documents involving multiple potential topics, the label is determined by the highest probability value.

Expert assessments are conducted based on semantic relevance, keyword consistency, and domain suitability using a 1–5 Likert scale. The results are averaged to draw conclusions based on these human evaluations. Semantic relevance refers to the degree of alignment in meaning among a set of keywords, the topic interpretation, and the content of documents associated with that topic; keyword consistency refers to the degree of interconnectedness and semantic coherence among keywords within a single topic; and domain suitability refers to the degree to which the topic aligns with the fintech context.

3.4 Random Over-Sampling with Neighborhood Cleaning for imbalanced data

After the documents are labeled, the imbalance problem is addressed using the ROS-NCL methods. The combination of ROS-NCL is a data balancing technique that aims to address imbalanced class distributions. ROS works by oversampling the minority class, while NCL cleans the data by removing samples that could lead to misclassification. This is important to prevent the model from being biased towards the majority class during training.

3.5 Classification with convolutional neural network–long short-term memory

After the data is balanced, CNN-LSTM is used as the primary architecture for classification. A CNN captures local features of the text, while an LSTM captures temporal dependencies or word sequences within the text. The CNN-LSTM enables the model to capture local textual features as

well as sequential dependencies between words in user reviews. CNN extracts local patterns from adjacent words, while the LSTM captures contextual relationships along the sequence of words.

3.6 Prediction and accuracy

The final step is generating predictions from the CNN-LSTM-ROS-NCL model. The model will generate predictions for each labeled document and then be evaluated based on accuracy. Accuracy measures how well the model classifies documents according to the correct topic labels.

Figure 1 shows the complete process, from topic extraction using NMF, topic interpretation using generative AI, to document classification using CNN-LSTM with the ROS-NCL data balancing method. This process aims to improve prediction accuracy by ensuring that the generated topics are relevant and that the trained model is capable of handling imbalanced data. This approach also emphasizes the importance of integrating the data balancing method and neural network architecture to achieve optimal results in topic-based classification.

3.7 Performance evaluation metrics

The performance evaluation metrics used to evaluate the model in the experiment include accuracy, F1-score, precision, and recall. A detailed explanation of these metrics is provided below:

3.7.1 Precision

True positive (TP): The number of data points that are truly positive and are classified as positive by the model. False positive (FP): The number of data points that are truly negative but are incorrectly classified as positive by the model.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (10)$$

Precision measures how accurately the model predicts the positive class. In other words, it shows how much of the model's positive predictions are truly positive.

3.7.2 Recall

False negative (FN): The number of data points that are actually positive but are incorrectly classified as negative by the model.

$$\mathcal{L} = \sum_{c=1}^C y_c \log(\hat{y}_c) \tag{11}$$

Recall measures the ability of the model to detect all true positive samples. It shows how much of the positive data the model actually identified.

3.7.3 F1-score

F1-score is the harmonic mean of precision and recall. F1-score is used when a balance between precision and recall is desired, especially when there is a trade-off between the two. It provides a single metric that takes into account both important aspects of classification performance, namely accuracy and completeness.

$$\text{F1 - score} = \frac{2 \times (\text{precision} \times \text{recall})}{(\text{precision} + \text{recall})} \tag{12}$$

3.7.4 Definition of classification elements

TP: The number of data points that are classified as positive and are actually positive.

True negative (TN): The number of data points that are classified as negative and are actually negative.

FP: The number of data points that are classified as positive but are actually negative.

FN: The number of data points that are classified as negative but are actually positive.

Precision is related to the accuracy of the positive predictions of the model. The higher the precision, the fewer the positive prediction errors. Recall is related to the model's ability to find all positive examples in the data. The higher the recall, the fewer positive samples the model misses. F1-score provides a balance between precision and recall, which is very useful when the dataset has an imbalanced class distribution.

As shown in Algorithm 1, the CNN-LSTM model combines the CNN's ability to extract local patterns from text with the LSTM's capacity to learn sequential relationships and long-term dependencies. The text is first converted into a vector representation via embedding, then processed through Conv1D, batch normalization, max pooling, and dropout layers. The features extracted by the CNN are subsequently processed by two sequential LSTM layers before being classified into eight classes via a dense layer with softmax activation. The model is trained using the Adam optimizer and categorical cross-entropy loss, incorporating Early Stopping and ReduceLROnPlateau to enhance training stability and

mitigate overfitting.

Algorithm 1. Convolutional neural network–long short-term memory (CNN–LSTM)

1. Model Initialization: Create a sequential model.
Embedding Layer: Add an Embedding layer with input 'vocab_size' and output 'embedding_dim'.
2. CNN Block:
Add Conv1D with 128 filters, kernel size 5, padding 'same', and activation 'relu'.
Add Batch Normalization.
Add MaxPooling1D with pool size 2.
Add Dropout with probability 0.3.
3. LSTM Block:
Add spatial dropout with probability 0.3.
Add the first LSTM with 128 units, return_sequences=True, dropout and recurrent dropout 0.3.
Add the second LSTM with 64 units, dropout and recurrent dropout 0.3.
4. Output Layer:
Add Dense with 8 units and activation 'softmax'.
5. Model Compilation:
Use loss 'categorical_crossentropy', optimizer 'adam', and metric 'accuracy'.
6. Initialize Callbacks:
Early Stopping on 'val_loss', patience 3, restore_best_weights = True. ReduceLROnPlateau on 'val_loss', factor 0.2, patience 2, min_lr 0.00001.
7. Model Training:
Train the model with 'x_train' and 'y_train' data, epochs = 15, batch_size = 128, validation_split = 0.1, use callbacks.
8. Visualize Results:
Plot accuracy and loss for training and validation data

4. RESULT AND DISCUSSION

4.1 Dataset analysis

This study uses a dataset of comments from Fintech e-wallet users in Indonesia, collected in the period 2023–2024 from the Play Store using API technology. A total of 42,680 rows in the dataset have been cleaned, with the following breakdown: DANA: 8,834; DOKU: 2,159; GoPay: 6,829; I-Saku: 4,821; LinkAja: 2,951; OVO: 7,942; ShopeePay: 7,204; Tapcash: 1,940. The dataset has been cleaned using a process involving language filtering, duplicate removal, text normalization, and the removal of entries containing fewer than four words.

Table 1 shows the results of the NMF algorithm with 8 topics generated, with words that have weights. The weight of the word shows how important or dominant the word is in each topic. Words with higher weights have stronger relevance to the topic. This table is useful for understanding what users pay most attention to in each topic discussed in the review, helping in further understanding user preferences.

Table 1. Topics with resulting weight

Topic ID	Top Words with Weights
1	easy (2.4277), complicated (1.2580), use (0.9171), pay (0.8650), features (0.8557), credit (0.8481), easy (0.8056), buy (0.7081), application (0.6260), cool (0.5881)
2	transfer (3.9642), free (1.8902), fee (1.3654), admin (1.2867), funds (0.6369), send (0.5778), transfer (0.5681), account (0.4467), ewallet (0.4129), free (0.3653)
3	good (3.2571), application (0.6815), download (0.6544), not bad (0.3298), pity (0.2756), next door (0.2207), uninstall (0.2198),

4	obstacle (0.2114), only (0.1948), hopefully (0.1948) help (3.0007), payment (1.3273), transact (0.6973), useful (0.6618), good (0.5498), digital (0.5424), success (0.4641), online (0.3718), finance (0.3608), needs (0.3263)
5	steadfast (2.9710), hopefully (1.2263), basically (0.9637), best (0.6754), improve (0.5382), best (0.4869), correct (0.4062), victorious (0.3274), basically (0.2967), success (0.2908)
6	balance (1.1732), login (0.9823), account (0.5732), borrow (0.5468), help (0.5441), loan (0.4133), failed (0.4092), difficult (0.4040), upgrade (0.3437), limit (0.3114)
7	transaction (3.9342), easy (1.4149), smooth (1.2361), easy (0.8986), steady (0.3517), all (0.1525), obstacles (0.1208), ber (0.1204), anywhere (0.1107), type (0.1016)
8	fast (2.7422), safe (1.5402), process (1.0770), easy (0.7871), money (0.6033), service (0.4606), response (0.4350), service (0.3991), trusted (0.3210), needed (0.3098)

Table 2. Keyword and user comment

Topic ID	Keywords	Text
1	easy, complicated, use, pay, features, credit, easy, buy, application, cool	Until now, there have been no problems with the appearance; it is simple and easy to use
2	transfer, free, cost, admin, funds, send, transfer, account, ewallet, free	Excellent for paying, free discounts
3	good, application, download, quite, unfortunately, next to, uninstall, obstacle, only, hopefully	Gopay maintenance has never disturbed customers, really great
4	help, payment, transaction, useful, good, digital, success, online, finance, needs	Very good, second to none, no regrets using Gopay
5	steadfast, hopefully, basically, best, improve, best, right, victorious, basically, success	Great for paying, transfers, easy to withdraw cash; it feels like you're holding it in your hand
6	balance, enter, account, borrow, help, loan, fail, difficult, upgrade, limit	If it fails, it takes a long time to return the funds. Not like the one next door; it's only a matter of hours, and the funds are returned. I'm disappointed
7	transaction, easy, smooth, make it easier, steady, all, obstacle, wherever, type	Great, as the name suggests, Gopay makes it easy to transact all digital lines, cashless problems, really awesome
8	fast, safe, process, easy, money, service, response, service, trusted, need	Great, safe, bonuses and others; hopefully they are safe and under control

Table 3. Expert validation results

Topic ID	Topic Interpretation	Expert	Semantic Relevance	Keyword Consistency	Domain Suitability	Average
1	Ease of use	Expert 1 (Fintech)	4.0	4.0	5.0	4.3
		Expert 2 (Fintech)	5.0	4.0	5.0	4.7
		Expert 3 (Language)	5.0	5.0	5.0	5.0
		Average Topic 1	4.7	4.3	5.0	4.7
2	Transfer and fees	Expert 1 (Fintech)	4.0	4.0	5.0	4.3
		Expert 2 (Fintech)	5.0	4.0	5.0	4.7
		Expert 3 (Language)	5.0	5.0	5.0	5.0
		Average Topic 2	4.7	4.3	5.0	4.7
3	Application performance and user sentiment	Expert 1 (Fintech)	4.0	5.0	5.0	4.7
		Expert 2 (Fintech)	4.0	4.0	5.0	4.3
		Expert 3 (Language)	4.0	5.0	5.0	4.7
		Average Topic 3	4.0	4.7	5.0	13.7
4	Digital payments and benefits	Expert 1 (Fintech)	4.0	5.0	5.0	4.7
		Expert 2 (Fintech)	4.0	5.0	5.0	4.7
		Expert 3 (Language)	4.0	4.0	5.0	4.3
		Average Topic 4	4.0	4.7	5.0	4.6
5	Enthusiastic feedback	Expert 1 (Fintech)	4.0	5.0	5.0	4.7
		Expert 2 (Fintech)	5.0	4.0	5.0	4.7
		Expert 3 (Language)	5.0	5.0	5.0	5.0
		Average Topic 5	4.7	4.7	5.0	4.8
6	Account and loan issues	Expert 1 (Fintech)	4.0	4.0	5.0	4.3
		Expert 2 (Fintech)	4.0	4.0	5.0	4.3
		Expert 3 (Language)	4.0	4.0	5.0	4.3
		Average Topic 6	4.0	4.0	5.0	4.3
7	Transactions and convenience	Expert 1 (Fintech)	5.0	5.0	5.0	5.0
		Expert 2 (Fintech)	4.0	5.0	5.0	4.7
		Expert 3 (Language)	5.0	5.0	5.0	5.0
		Average Topic 7	4.7	5.0	5.0	4.9
8	Security and customer service	Expert 1 (Fintech)	4.0	4.0	5.0	4.3
		Expert 2 (Fintech)	5.0	5.0	5.0	5.0
		Expert 3 (Language)	5.0	5.0	5.0	5.0
		Average Topic 8	4.7	4.7	5.0	4.8

Table 4. Meaning of the topics

Topic ID	Topic Name	Meaning
1	Ease of use	Related to the ease or difficulty of using a fintech application, especially in payment and purchase features.
2	Transfers and fees	Focuses on the process of transferring funds and the costs involved, including the expectation of free transfers.
3	App performance and user sentiment	Reflects user sentiment regarding the performance of the application, including positive reviews and complaints.
4	Digital payments and benefits	Users highlight the benefits of digital payments and the ease of transactions provided by the application.
5	Enthusiastic feedback	Positive feedback from enthusiastic users, with the expectation of future feature improvements.
6	Account and loan issues	Describes issues faced by users regarding balances, accounts, loans, and transaction limits.
7	Transactions and convenience	Related to an easy and smooth transaction experience, despite some obstacles.
8	Security and customer service	Focuses on service security and the quality of customer service responses, including trust and speed.

Table 5. Number of topic clusters

Topic ID	Numbers
1	6399
2	4688
3	4952
4	4491
5	3957
6	11116
7	3229
8	3848
total	42680

Table 2 shows examples of keywords arranged from a collection of sentences contained in the content column. Table 2 presents the results of extracting eight topics using NMF from the e-wallet user comment dataset. The topic ID column displays the numerical identity of the NMF extraction results. The Keywords column represents the distribution of dominant words in user comments on each topic, obtained from the H matrix. The Text column represents an example of an e-wallet user comment document with the highest probability weight in the W matrix. This demonstrates a direct relationship between the distribution of words within a topic and the distribution of topics within documents, making the displayed text an empirical representation of the keyword pool within that topic.

Table 3 presents the results of the topic quality validation for the NMF TM output, based on assessments by three experts using three criteria: semantic relevance, keyword consistency, and domain suitability, on a scale of 1–5. The evaluation results indicate that the eight generated topics possess interpretability, keyword consistency, and suitability for the fintech context, making them appropriate for further analysis of e-wallet user reviews. Domain suitability received the highest score, demonstrating that the topics align well with the e-wallet analysis domain. Keyword consistency and semantic relevance also achieved high scores, indicating that the keywords constituting each topic are semantically related and effectively represent the topic's meaning.

Table 4 helps understand the primary focus of each user review based on the most relevant keywords and reflects important aspects such as ease of use, cost, performance, and security of fintech applications. This meaning-making process was carried out by developing prompts. The meaning-making results using OpenAI were validated by two fintech experts and one language expert.

Table 4 continues the explanation from Table 2, adding columns for topic name and meaning, showing the results of

topic analysis conducted using OpenAI to identify the meaning of keyword combinations in user reviews of fintech applications. Each topic contains keywords that frequently appear together in reviews and is assigned meanings based on the context in which they are used.

Table 5 shows the amount of data in each topic cluster, with varying amounts per topic. The data shows that the amount of data in each topic varies significantly. This supports the use of an imbalanced data algorithm.

4.2 Analysis of convolutional neural network–long short-term memory

Table 6 shows that the accuracy rate is above 95%, indicating that fintech user comments on topics such as ease of use of the application, low-cost transfers, and application performance are relatively consistent and easily predicted by the model. Focused comments and frequent repetition of the same pattern make it easier for the model to identify keywords and contexts of each comment. The high error rate for Topic 6: this class consists of user complaints or more complex problems, such as refunds, failed transactions, or account limitations, which can be expressed in various ways by users. High variability in comment content can make it difficult for the model to classify comments correctly. Although the data imbalance handling technique has been carried out, the accuracy rate cannot be significantly improved. However, from several papers on measuring the level of multi-variable accuracy, an accuracy of 75% is acceptable for use.

Table 6. CNN–LSTM–ROS–NCL accuracy

Topic	CNN–LSTM Accuracy	Misclassification Rate
1	93.19%	6.81%
2	98.70%	1.30%
3	97.69%	2.31%
4	98.69%	1.31%
5	98.09%	1.91%
6	77.19%	22.81%
7	95.50%	4.50%
8	98.20%	1.80%

Note: CNN–LSTM = Convolutional neural network–long short-term memory, ROS–NCL = Random Over-Sampling with Neighborhood Cleaning.

Overall, the CNN–LSTM–ROS–NCL model performs very well in most classes except class 6, which has problems with variability in the data. Classes with simpler or more general content, such as those related to transactions and ease of use,

are easier for the model to predict, while classes related to complaints and technical issues, such as class 6, still require more specific handling to improve their classification performance.

Table 7 shows that the total accuracy of the model is 0.96; this shows that overall, the model works very well in classifying the data. Macro average has precision 0.96, recall 0.94, F1-score 0.95: The macro average value calculates the average of each topic individually. Weighted average: precision 0.96, recall 0.96, F1-score 0.96: The weighted average calculates the average by considering the amount of support in each topic. This value is more representative of the overall performance because it takes into account the size of each topic. Since topics with greater support have high performance, the weighted average is very good. The macro average value is lower because Topic 6 affects the overall recall value, while the weighted average remains high because topics with large support have good performance. The overall model is very accurate, with precision, recall, and F1-score close to 1 in most topics. This shows that the model is very effective for classification in most topics, with some improvements likely needed to improve performance in topics with less data or that are more difficult to recognize.

Table 7. Accuracy CNN–LSTM–ROS–NCL

Topic	Precision	Recall	F1-Score	Support
1	0.95	0.93	0.94	1071
2	0.96	0.99	0.97	808
3	0.95	0.97	0.96	695
4	0.97	0.98	0.98	786
5	0.98	0.98	0.98	857
6	0.95	0.74	0.83	170
7	0.96	0.96	0.96	960
8	0.97	0.99	0.98	880
Accuracy	-	-	0.96	6227
Macro average	0.96	0.94	0.95	6227
Weighted average	0.96	0.96	0.96	6227

Note: CNN–LSTM = Convolutional neural network–long short-term memory, ROS–NCL = Random Over-Sampling with Neighborhood Cleaning.

4.3 Comparative analysis

Table 8 shows the comparison of the accuracy levels of three different models, namely CNN–LSTM–ROS–NCL, LSTM–ROS–NCL, AND CNN–ROS–NCL, on eight different classes. Each model combines DL techniques (CNN–LSTM) with methods to handle data imbalance, namely ROS–NCL.

Table 8. Comparison of several algorithms

Class	CNN–LSTM– ROS–NCL Accuracy	LSTM–ROS– NCL Accuracy	CNN–ROS– NCL Accuracy
1	93.19%	89.35%	94.31%
2	98.70%	96.59%	97.43%
3	97.69%	94.12%	97.70%
4	98.69%	94.78%	97.89%
5	98.09%	95.00%	97.29%
6	77.19%	65.93%	60.06%
7	95.50%	92.24%	97.62%
8	98.20%	97.01%	96.99%

Note: CNN–LSTM = Convolutional neural network–long short-term memory, ROS–NCL = Random Over-Sampling with Neighborhood Cleaning.

Table 9. CNN–LSTM without data balancing

Topic	Precision	Recall	F1-Score	Support
1	0.75	0.85	0.80	647
2	0.88	0.72	0.80	507
3	0.71	0.73	0.72	482
4	0.81	0.67	0.74	458
5	0.76	0.72	0.74	403
6	0.84	0.88	0.86	1080
7	0.56	0.67	0.61	304
8	0.73	0.69	0.71	387
Accuracy	-	-	0.77	4268
Macro average	0.76	0.74	0.75	4268
Weighted average	0.78	0.77	0.77	4268

Note: CNN–LSTM = Convolutional neural network–long short-term memory.

Table 10. Classification results of the Random Forest algorithm

Topic	Precision	Recall	F1-Score	Support
1	0.542	0.670	0.599	1280
2	0.642	0.635	0.638	938
3	0.581	0.390	0.467	990
4	0.601	0.526	0.561	898
5	0.599	0.424	0.496	791
6	0.631	0.928	0.752	2223
7	0.564	0.265	0.360	646
8	0.631	0.365	0.463	770
Accuracy	-	-	0.605	8536
Macro average	0.599	0.525	0.542	8536
Weighted average	0.602	0.605	0.584	8536

The CNN–LSTM–ROS–NCL model overall has the highest accuracy in most classes, indicating that the combination of CNN–LSTM enhanced with ROS–NCL provides the best performance on the data faced. CNN–ROS–NCL shows quite competitive performance, with little difference in accuracy compared to CNN–LSTM–ROS–NCL, but still superior to LSTM–ROS–NCL in some classes. LSTM–ROS–NCL generally has the lowest performance among the three models, especially in handling class 6, where this model only achieves an accuracy of 65.93%.

The CNN–LSTM algorithm without a balancing method achieved an accuracy of 77% with a macro F1-score of 0.75, indicating that the model was affected by class distribution imbalance. After applying ROS–NCL, accuracy rose to 96%, while the macro F1-score improved to 0.95. This improvement demonstrates that ROS–NCL effectively reduces classification bias and enhances recognition capabilities across various topic categories, as shown in Tables 7 and 9. A comparison of Table 9 and Table 7 reveals that the F1-score increased with the use of the balancing algorithm. The most significant improvement was observed in Topic 7, where the F1-score rose from 0.61 to 0.96. This indicates that ROS–NCL successfully improved representation learning for previously underrepresented topics. However, Topic 6 remained a challenge, with the recall value dropping from 0.88 to 0.74 following the balancing process.

Table 10 presents an analysis of the Random Forest algorithm, showing a macro-average precision of 0.599. This value indicates that the model is capable of generating topic predictions with a reasonable level of accuracy based on dominant word patterns within the documents; however, it does not outperform the CNN–LSTM and CNN–LSTM–ROS–NCL algorithms.

Table 11. Example of classification error analysis

Review Text	Actual Class	Predicted Class	Confidence Score	Error Pattern
My money just vanished on its own. I tried transferring to another ... account, but the balance didn't go through; it just disappeared.	T6	T2	99.99%	Balance/ account issue overlaps with transfer
It's terrible; whenever I try to make a cash withdrawal, I never get the code...	T6	T4	99.97%	Account/ service issue overlaps with digital payment
I can't log in to my ... account; it says there's a login issue. The GoPay gift feature menu has disappeared from my phone, and my tokens are gone too. Why is this happening? ... is so mean; I had collected so many tokens, you know? That's just awful.	T6	T7	99.84%	Login/ account issue overlaps with transaction convenience

Table 12. Accuracy test results of several algorithms

Algorithm	Test Accuracy
CNN-LSTM-ROS-NCL	96.26%
LSTM-ROS-NCL	93%
CNN-ROS-NCL	96.21%

Note: CNN-LSTM = Convolutional neural network-long short-term memory, ROS-NCL = Random Over-Sampling with Neighborhood Cleaning.

Class 6 in Table 8 is the most difficult to predict by all models, as seen from the significant drop in accuracy compared to other classes. The CNN-LSTM-ROS-NCL model performs best in this class with an accuracy of 77.19%, far above both LSTM-ROS-NCL (65.93%) AND CNN-ROS-NCL (60.06%). This may indicate that class 6 has a highly imbalanced distribution or has features that are difficult for the model to learn, so LSTM in combination with CNN helps capture important temporal patterns in the data. The CNN-LSTM-ROS-NCL combination is the most effective approach in dealing with data imbalance issues and in predicting difficult classes. Although CNN-ROS-NCL performs closely, the use of LSTM provides additional benefits in dealing with more difficult classes, such as class 6. The LSTM-ROS-NCL model performs the worst, especially on difficult classes, indicating that using only LSTM without CNN may not be powerful enough to capture the complexity of the data.

The analysis of misclassifications regarding Topic 6 in Table 11 reveals that the low recall is attributable not only to linguistic variation but also to semantic overlap with several adjacent topics. For instance, reviews concerning missing balances during transfers exhibit characteristics of both account/balance issues and transfer-related problems. Similarly, reviews regarding failed cash withdrawals involve a service-related context yet explicitly feature cash withdrawal activity that strongly points toward the digital payment topic. Reviews concerning login failures and the loss of app features may even encompass multiple service-related intents simultaneously. These findings indicate that a single user review can contain several interconnected intents, whereas the research framework assigns only one dominant topic. Consequently, Topic 6 possesses more complex semantic boundaries compared to some of the other categories.

CNN-LSTM-ROS-NCL gives the best result with 96.26% accuracy, but the difference with CNN-ROS-NCL is only 0.05 percentage points, which means that in this dataset, the temporal pattern captured by LSTM is not very significant in improving the performance, as seen in Table 12. CNN-ROS-NCL shows that CNN with proper data imbalance handling is very effective in predicting the results. LSTM-ROS-NCL shows lower performance, indicating that CNN may be more

suitable for this dataset than using LSTM alone, especially when the data has a more dominant spatial pattern. Overall, the combination of CNN with imbalance handling methods (ROS-NCL) gives the best performance, while LSTM only provides a little additional benefit in some cases.

The integration of ROS-NCL proved effective in reducing bias toward the majority class while maintaining classification stability across topics. The combined CNN-LSTM architecture further improved the model's ability to capture local textual features and sequential dependencies, contributing to an overall accuracy of 96.26%. While the additional contribution of LSTM compared to CNN alone was small on this dataset, the hybrid architecture demonstrated stronger robustness on more complex classes.

The comparison of accuracy between CNN-LSTM-ROS-NCL and CNN-ROS-NCL shows an improvement of only about 0.05 percentage points, indicating a limited contribution; however, CNN-LSTM provides improvements for more difficult classes, particularly class 6, thereby demonstrating its benefits for more complex language patterns.

From a practical standpoint, the proposed topic-based classification framework utilizing NMF algorithms and a CNN-LSTM-ROS-NCL classification model provides structured information on user experience categories that can serve as a basis for analyzing fintech service improvements. By organizing user feedback into easily understandable semantic categories and linking them to predictive analytics models, this framework enables fintech organizations to more systematically understand the distribution patterns and characteristics of issues arising in user reviews.

5. CONCLUSION

This study successfully proves that a hybrid approach combining NMF for topic extraction with CNN-LSTM for classification is effective in analyzing unstructured and imbalanced fintech user reviews. From the results of topic extraction using NMF, eight main topics were identified, covering various important aspects of fintech services such as ease of use, transfer and fees, application performance, benefits of digital payments, positive feedback, account and loan issues, ease of transaction, and security and customer service.

These topics provide in-depth insights into the various factors that influence user perceptions and experiences of fintech services. For example, ease of use and transfer and fees are topics with more consistent reviews, so the CNN-LSTM model is able to classify them with high accuracy. Meanwhile, topics such as account and loan issues are more complex due

to the variation in language used by users to convey complaints, thus requiring more in-depth handling to improve prediction accuracy.

The use of CNN–LSTM has proven effective in capturing spatial and temporal patterns from user reviews, resulting in a classification accuracy of up to 96.26%. The ROS–NCL techniques also successfully handled data imbalance, especially in difficult classes such as topics related to account issues, which generally have fewer reviews but are highly variable. These results are better than the previous study [13] using the CNN–LSTM architecture, which achieved an accuracy of 93.21% for binary classification and 92.9% for multiclass classification in the UNSW-NB15 dataset. This study provides strong evidence that the combination of the NMF method with CNN–LSTM not only improves prediction accuracy but is also able to dig deeper into insights into user perceptions of various fintech service features. With the topics generated from NMF, fintech organizations can better understand which service areas need to be improved, both in terms of ease of use, service costs, and aspects of user security and convenience. This study not only contributes to the development of algorithms in fintech data analytics, but also to improving service quality based on more targeted and in-depth feedback.

Although the proposed framework is capable of providing semantic interpretation through TM and expert validation, this study has not conducted a quantitative evaluation of the model's explainability. Future research could integrate Explainable AI methods, such as SHapley Additive exPlanations (SHAP) or attention visualization, to measure model-level interpretability.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the Ministry of Higher Education, Science, and Technology for the financial support provided through the Regular Fundamental Grant Program for the 2024 Fiscal Year, under Decree Number 0459/E5/PG.02.00/2024 (dated May 30, 2024) and Contract Number 108/E5/PG.02.00.PL/2024-011/LL6/PB/AL.04/2024. Appreciation is also extended to Soegijapranata Catholic University for the facilities and support provided throughout the course of this research.

REFERENCES

- [1] Li, Y., Zhang, L. (2021). Do online reviews truly matter? A study of the characteristics of consumers involved in different online review scenarios. *Behaviour & Information Technology*, 40(13): 1448-1466. <https://doi.org/10.1080/0144929x.2020.1759691>
- [2] Chen, W., Gu, B., Ye, Q., Zhu, K.X. (2019). Measuring and managing the externality of managerial responses to online customer reviews. *Information Systems Research*, 30(1): 81-96. <https://doi.org/10.1287/isre.2018.0781>
- [3] De', R., Pandey, N., Pal, A. (2020). Impact of digital surge during COVID-19 pandemic: A viewpoint on research and practice. *International Journal of Information Management*, 55: 102171. <https://doi.org/10.1016/j.ijinfomgt.2020.102171>
- [4] Liu, H., Motoda, H. (2007). *Computational Methods of Feature Selection*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Taylor & Francis, New York. <https://doi.org/10.1201/9781584888796>
- [5] Casalino, G., Castiello, C., Del Buono, N., Mencar, C. (2018). A framework for intelligent Twitter data analysis with non-negative matrix factorization. *International Journal of Web Information Systems*, 14(3): 334-356. <https://doi.org/10.1108/ijwis-11-2017-0081>
- [6] Alonso, J.M., Castiello, C., Mencar, C. (2015). Interpretability of fuzzy systems: Current research trends and prospects. In *Springer Handbook of Computational Intelligence*, pp. 219-237. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-43505-2_14
- [7] Egger, R., Yu, J. (2022). A topic modeling comparison between LDA, NMF, Top2Vec, and BERTopic to demystify Twitter posts. *Frontiers in Sociology*, 7: 886498. <https://doi.org/10.3389/fsoc.2022.886498>
- [8] Esposito, F. (2021). A review on initialization methods for nonnegative matrix factorization: Towards omics data experiments. *Mathematics*, 9(9): 1006. <https://doi.org/10.3390/math9091006>
- [9] Wang, H., Prakash, N., Hoang, N.K., Hee, M.S., Naseem, U., Lee, R.K.W. (2023). Prompting large language models for topic modeling. In *2023 IEEE International Conference on Big Data (BigData)*, Sorrento, Italy, pp. 1236-1241. <https://doi.org/10.1109/bigdata59044.2023.10386113>
- [10] Zhang, X., Zhao, J., LeCun, Y. (2015). Character-level convolutional networks for text classification. In *NIPS'15: Proceedings of the 29th International Conference on Neural Information Processing Systems*, Montreal, Canada, pp. 649-657.
- [11] Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1746-1751. <https://doi.org/10.3115/v1/d14-1181>
- [12] Yin, W., Kann, K., Yu, M., Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *arXiv preprint arXiv:1702.01923*. <https://doi.org/10.48550/arXiv.1702.01923>
- [13] Hasib, K.M., Azam, S., Karim, A., et al. (2023). MCNN-LSTM: Combining CNN and LSTM to classify multi-class text in imbalanced news data. *IEEE Access*, 11: 93048-93063. <https://doi.org/10.1109/ACCESS.2023.3309697>
- [14] Bensaoud, A., Kalita, J. (2024). CNN-LSTM and transfer learning models for malware classification based on opcodes and API calls. *Knowledge-Based Systems*, 290: 111543. <https://doi.org/10.1016/j.knosys.2024.111543>
- [15] Setiawan, W.F., Amirullah, A., Ariatama, I.P., Anggraini, R.N.E. (2026). CNN-LSTM architecture for multi-task sentiment and emotion classification on large-scale Indonesian TikTok application reviews. *Communication and Information Technology Journal*, 20(1): 77-91. <https://doi.org/10.21512/commit.v20i1.13876>
- [16] Amirkhalili, Y., Wong, H.Y. (2025). Banking on feedback: Text analysis of mobile banking iOS and Google app reviews. *arXiv preprint arXiv:2503.11861*. <https://doi.org/10.48550/arXiv.2503.11861>
- [17] Blei, D., Carin, L., Dunson, D. (2010). Probabilistic topic models. *IEEE Signal Processing Magazine*, 27(6): 55-65. <https://doi.org/10.1109/msp.2010.938079>
- [18] Jia, Y., Kwong, S., Hou, J., Wu, W. (2019). Semi-

- supervised non-negative matrix factorization with dissimilarity and similarity regularization. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7): 2510-2521. <https://doi.org/10.1109/tnnls.2019.2933223>
- [19] Blei, D.M., Ng, A.Y., Jordan, M.I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3: 993-1022.
- [20] O'Callaghan, D., Greene, D., Carthy, J., Cunningham, P. (2015). An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, 42(13): 5645-5657. <https://doi.org/10.1016/j.eswa.2015.02.055>
- [21] Zoya, Latif, S., Shafait, F., Latif, R. (2021). Analyzing LDA and NMF topic models for Urdu tweets via automatic labeling. *IEEE Access*, 9: 127531-127547. <https://doi.org/10.1109/access.2021.3112620>
- [22] Egger, R., Yu, J. (2022). Identifying hidden semantic structures in Instagram data: A topic modelling comparison. *Tourism Review*, 77(4): 1234-1246. <https://doi.org/10.1108/tr-05-2021-0244>
- [23] Jiang, D., Chen, Z., Lian, R., Bao, S., Li, C. (2017). Familia: An open-source toolkit for industrial topic modeling. *arXiv preprint arXiv:1707.09823*. <https://doi.org/10.48550/arXiv.1707.09823>
- [24] Widiatoro, A.D., Mustafid, M., Sanjaya, R. (2023). Analyzing user comments on fintech e-wallet applications in Indonesia using the LDA algorithm. In *2023 7th International Conference on Information Technology (IncIT)*, Chiang Rai, Thailand, pp. 169-174. <https://doi.org/10.1109/incit60207.2023.10412996>
- [25] Widiatoro, A.D., Mustafid, M., Sanjaya, R. (2023). Leveraging latent Dirichlet allocation for feature extraction in user comments: Enhancements to user-centered design in Indonesian financial technology. *Ingénierie des Systèmes d'Information*, 28(5): 1423-1433. <https://doi.org/10.18280/isi.280530>
- [26] Murfi, H., Siagian, F.L., Satria, Y. (2019). Topic features for machine learning-based sentiment analysis in Indonesian tweets. *International Journal of Intelligent Computing and Cybernetics*, 12(1): 70-81. <https://doi.org/10.1108/ijicc-04-2018-0057>
- [27] Pauca, V.P., Shahnaz, F., Berry, M.W., Plemmons, R.J. (2004). Text mining using non-negative matrix factorizations. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, Lake Buena Vista, Florida, USA, pp. 452-456. <https://doi.org/10.1137/1.9781611972740.45>
- [28] Kamalov, F., Leung, H.H., Cherukuri, A.K. (2023). Keep it simple: Random oversampling for imbalanced data. In *2023 Advances in Science and Engineering Technology International Conferences*, Dubai, United Arab Emirates, pp. 1-4. <https://doi.org/10.1109/aset56582.2023.10180891>
- [29] Johnson, J.M., Khoshgoftaar, T.M. (2020). The effects of data sampling with deep learning and highly imbalanced big data. *Information Systems Frontiers*, 22(5): 1113-1131. <https://doi.org/10.1007/s10796-020-10022-7>
- [30] Choirunnisa, S., Lianto, J. (2018). Hybrid method of undersampling and oversampling for handling imbalanced data. In *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Yogyakarta, Indonesia, pp. 276-280. <https://doi.org/10.1109/isriti.2018.8864335>
- [31] Laurikkala, J. (2001). Improving identification of difficult small classes by balancing class distribution. In *Lecture Notes in Computer Science*, pp. 63-66. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-48229-6_9

NOMENCLATURE

b	Bias term in neural network computation
C	Number of output classes
d	Embedding dimension
D	Dropout output vector
E	Word embedding representation
F	Convolutional feature representation
H	Topic-document representation matrix in NMF
h	Hidden-state representation of LSTM
k	Number of topics in NMF
L	Categorical cross-entropy loss
m	Number of documents
n	Number of words/features
P	Max-pooling output representation
r	Spatial-dropout mask
S	Spatial-dropout output representation
T	Input sequence length
V	Document-word matrix
W	NMF basis matrix
W_e	Word-embedding matrix
W_{conv}	Convolution kernel matrix
W_d	Dense-layer weight matrix
X	Input token sequence
x_i	Input representation at position i
y	Output class probability vector
y_c	True label for class c
$\hat{y}_c$	Predicted probability for class c
z	Linear output score before softmax

Greek symbols

$\mathcal{L}$	Categorical cross-entropy loss function
σ	Sigmoid activation function
$\tanh$	Hyperbolic tangent activation function

Subscripts

c	Class index
f	Forget gate
i	Input gate or token position, according to context
o	Output gate
t	Position of a token in the input sequence
$conv$	Convolutional layer
d	Dense layer
e	Embedding layer