Received: May 21, 2024. Revised: June 18, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Model Analytic in Fintech User Comment Features Using LDA-CNN on Imbalanced Data

Albertus Dwiyoga Widiantoro[1,2]* Mustafid Mustafid[1] Ridwan Sanjaya[2]

[1]*Department of Doctoral Program of Information System, Diponegoro University, Semarang, Indonesia*
[2]*Information Systems Department, Soegijapranata Catholic University, Semarang, Indonesia*
* Corresponding author's Email: yoga@unika.ac.id

Abstract: Peer-to-peer (P2P) lending platforms are growing significantly, and users always leave comments on the application to provide ratings. User comments are important to analyze to see the needs and constraints of fintech users. The Purpose of the research is to create an analytical model that effectively addresses the problem of limited accuracy in classification due to data imbalance in P2P Lending platforms. The research aims to improve feature detection and overall model quality by effectively managing imbalanced data. The design involves a combination of techniques. First, the Latent Dirichlet Allocation (LDA) method is used to organize topics and label data. To address the data imbalance, the study employs Random Over Sampling (ROS) and Neighborhood Cleaning Rule (NCL). The final classification is performed using Convolutional Neural Networks (CNN). Additionally, a comparative analysis with other algorithms like LSTM and CNN-LSTM is carried out to validate the effectiveness of the proposed approach. The Findings reveal that the CNN-ROS-NCL model is capable of managing imbalanced data, which improves class distribution and enhances the model's quality by reducing noise and misleading samples. The CNN model achieved a classification accuracy of 94.66% on 10 feature classes, suggesting a significant improvement in feature detection and classification performance on the P2P Lending platform. The Originality of this research lies in the innovative integration of LDA for topic analysis with CNN for classification a novel approach in the context of fintech feature development. This combination has not previously been used in fintech and offers a new way to automatically detect features in Fintech P2P Lending user comment datasets by identifying key topics. The research contributes to the enhancement of fintech applications and services by providing a model that improves the understanding and processing of user comments on P2P platforms.

Keywords: Model analytic, Feature, Topic model, Fintech, P2P lending, LDA-CNN.

1. Introduction

Topic modeling on user comments helps influencing of business decisions and software development significantly. As seen from research, analysis of user comments on social media can reveal public sentiment towards new topics and events, which is important for adjusting marketing strategies or product features [1]. Companies can identify themes that are important to customers, such as product quality and features, to improve product design on Huawei phones, where user experience aspects such as smoothness and cost performance are analyzed based on sentiment [2].

P2P lending platforms provide alternative credit options for individuals and small businesses, which have many competitive advantages that have resulted in substantial growth in loan volume and the number of platforms [3]. P2P lending offers lenders better rates of return and greater access to affordable credit for borrowers who may have limited access to banks, so that this type of loan can outperform conventional loans in the retail sector [4]. P2P lending also offers technological expertise and a flexible financial system. P2P lending can eliminate inefficiencies and overhead lending costs by removing barriers for borrowers who have limited access to credit due to low creditworthiness [5], which complements traditional lending channels by acquiring high-

quality, low-risk customers who are underserved by traditional funding. Financial technology (Fintech) can reduce loan costs by using fully automatic algorithms to determine prices and guarantee loans through appropriate systems [6].

There are two important quality attributes for evaluating software applications: usability and user experience. Usability is a task-oriented attribute that measures the extent to which a system, product, or service enables users to achieve their goals with efficiency and effectiveness [7]. In addition to the pragmatic aspects related to completing the tasks addressed by usability, user experience also considers every part of the user interaction, from the consequences of the user's internal states, namely affect, sensations, and emotions, to the characteristics of the system being designed and the context in which the interaction occurs [8].

Designing intelligent systems capable of identifying user needs by translating user feedback to attribute-level design is a key prerequisite for the success of a user-centered design process. Studies show that 49% of design firms lack systems and tools to monitor external platforms, and only 8% have adopted a data-driven digital approach to new product development despite recognizing it as a high priority [9].

Rapidly developing Fintech platforms provide an untapped source of knowledge about a much larger and more diverse set of user preferences and in many instances, for example, the growth of Fintech as seen from the prevalence of Android applications on the PlayStore, is very significant [10]. Problems in this sector include security, privacy, data, infrastructure, application, management, and service models [11], user complaints, lack of consumer protection [12], inadequate online support, and inadequate error detection [13].

Latent Dirichlet Allocation (LDA) is a probabilistic generative model that can be used to estimate multinomial observations with an unsupervised learning approach [14]. With LDA, each document is represented as a multinomial topic distribution where topics can be seen as high-level concepts in the document, seeing that documents are collections of topics where each topic is presented with a mixture of words. The LDA topic modeling approach is used in extracting product aspects in aspect-based sentiment analysis [15], identifying key topics for consideration by companies in marketing campaigns [16], and automatically separating opinions in multi-language hotel, restaurant, and electronic device comment data [17]. However, there has been no use of LDA to identify Fintech features based on user comments that can be utilized for service development.

The problem of unlabeled data poses a significant innovation challenge but also motivates the development and use of machine learning strategies. The development of the Topic2labels framework, which offers an automatic method for tagging data using LDA techniques, aims to identify topics from data sets automatically. The integration of label data into the LDA process produces a variety of supervised and semi-supervised topic models, with the aim of improving the topic modeling process as well as document classification [18]. LDA shows significant improvements in performance in multi-label classification [19]. This approach has succeeded in improving LDA's capabilities in multi-label document classification, especially through combining label frequency data and relationships between labels. This research proposes to use the LDA topic clustering results for the feature-based labeling process.

A feature is a cohesive set of system functionality that is part of the system requirements, part of the system implementation, and an artifact. A system requirements subset is a precise specification that encompasses all the significant behavioral attributes of a system [20]. Features refer to a collection of predetermined words or phrases taken from written reviews that can convey specific characteristics of an item, such as the quality of cuisine, service, and atmosphere, in a restaurant recommendation system [21]. User reviews that have been identified are transformed into features, which then enable the creation of system artifacts and services.

Coherence is crucial for improving the readability and applicability of topic modeling outcomes in many areas. The coherence score evaluates the degree of semantic correlation among the high-probability terms inside a topic, hence serving as a metric for assessing the quality of the topic. High coherence indicates that the words make sense as a whole, making the topic easy to interpret [22]. A high coherence score correlates with human interpretations of topic quality, making coherence a useful metric for ensuring that LDA-generated topics are relevant and understandable to users. It is important to make informed decisions based on the topics identified through LDA [23].

Imbalanced data sets are a frequently encountered and studied problem in the financial industry [24], and data imbalance is a major problem faced when applying machine learning concepts to real-life problems. Imbalanced data sets pose severe problems for many supervised learning problems [25].

Likewise, the results of the LDA process in this research produced unbalanced data.

The issues in this research can be summarized as follows: the challenge of identifying overlapping collections of user comments in Fintech P2P lending, the limited accuracy of classification algorithms in handling imbalanced data, the difficulties in identifying fintech features, the labeling of features, and the automated detection of user comments on fintech P2P lending platforms. We need an automated analysis approach for the Fintech feature to solve this problem.

The objective of this research is to develop an analytical model that can effectively tackle this problem by employing the LDA approach for topic generation and data labeling, then subsequently applying a Random Over Sampler (ROS), the neighborhood cleaning rule (NCL), Adaptive Synthetic (ADASYN), and Synthetic Minority Over-sampling (SMOTE) to handle imbalanced data. The subsequent step involves the utilization of Convolutional Neural Networks (CNN) for the purpose of classification. Model assessment use many criteria, including accuracy, precision, recall, F1-score, and support, to gauge the performance of a model.

The main innovation of this research is a set of Fintech feature analysis methods that combine LDA topic analysis with CNN classification on imbalanced data. This technique is not commonly used in existing literature, especially in the fintech P2P lending field. An automatic method for annotating textual data is used by utilizing LDA distribution. This strategy reduces the need for manual labeling and automates the conversion of subjects to features by creating feature datasets. By using ROS and NCL data processing techniques, the class distribution is improved and the model quality is improved through noise reduction and elimination of misleading samples. Therefore, a Fintech model was created that can accurately recognize Fintech characteristics, handle imbalanced data effectively, and achieve high classification performance.

In this study, a comparison was also carried out with other algorithms, namely LSTM and CNN-LSTM, to validate the accuracy of the proposed approach, because LSTM is a good algorithm for multi-label classification.

This paper's structure is as follows: In Section 1, we present the research background; in Section 2, we present related work; and in Section 3, we present the research methodology and explanation of the architecture of LDA, CNN, ROS, and NCL. Section 4, Results and Discussion, includes the construction of the data set; in Section 5, we conclude the paper with a summary and discuss potential future work.

2. Related work

LDA was designed as topic modeling to find groups of words that often appear together in documents in a data set [26]. Topic modeling is text mining to process, organize, manage, and extract knowledge based on probabilistic modeling used to discover hidden structures in large archives of documents based on similar word usage patterns in each document. It is used to determine the underlying "topic" of a text document. A topic represents a broader concept belonging to a corpus of documents [27].

2.1 Labeling topic

The results of the LDA process that produces topics will be processed to label the initial dataset using an automatic method.

Incorporating label information into LDA results for document classification purposes offers a variety of supervised and semi-supervised topic models. LDA models that utilize label information by formulating relationships between labels produce significant improvements in the classification of both single labels and multi-labels [30]. At this stage, the labeling process will utilize the number of words and the weight of each word in the document.

2.2 Coherence

This study emphasizes the importance of score coherence in evaluating and selecting LDA topic models. Coherence scores not only facilitate the comparison of different models or configurations but also guide researchers in optimizing models to ensure that the extracted topics are meaningful and allow for interpretation. Studies using LDA demonstrate the importance of selecting techniques, modeling topics, and measures with appropriate coherence for specific corpora. This study emphasizes role weighting terms and suitability methods to model various topics for analyzing non-mainstream domains [31]. A study to investigate the impact of using abstracts versus full text on LDA-generated topic coherence showed that LDA coherence and topic rankings were influenced by the type of textual data, with document frequency, word length, and vocabulary size showing that the influence varied on topic coherence [32]. Topic modeling in reviewing e-commerce products, using score coherence as a quality topic to reveal that LDA outperforms other techniques on their dataset, score

coherence as an important factor measure in the effectiveness of topic models [33].

2.3. CNN classification

Text classification automates text categorization into predefined classes, making data management more efficient. The benefits of classification include text retrieval systems that respond to user questions with the aim of understanding the text, such as summarization or question answering systems [34], increasing the accuracy of information retrieval systems by ensuring only relevant information is retrieved according to user needs [35], spam filtering, identification language, and sentiment analysis [36]. Modern text classification systems are designed to be able to adapt to various data types and be accurate enough to meet user expectations, thus having benefits for a variety of domains, including web content management and enterprise data handling [37].

This research proposes a CNN algorithm for classification because, in several studies using CNN, it produces excellent accuracy, namely classifying online short messages in Mandarin with effective results in label evaluation [38]. CNN-based short text classification improves the results of short text classification with a combination of language models. N-gram and concentration mechanisms for feature selection [39], Chinese text classification based on hybrid CNN and LSTM models achieved good results on news datasets [40], and the text classification DEP-CNN model for multiclass and multilabel text classification showed good performance. outperforms on four benchmark datasets when compared to its peers [41]. A CNN model for text classification improves robustness against overfitting and improves performance on five benchmark datasets [42].

2.4 Imbalanced data

In handling text data that is not balanced, using ROS and NCL methods is one of the internal strategies to solve the problem. This hybrid method of sampling and oversampling is proposed to overcome the data imbalance. A number of studies that use ROS and NCL are collectively separated, including the NCL method as a method of sampling capable of increasing mark accuracy [43-44]. The ROS method has good performance and emphasizes efficiency, computing, and accuracy; it is robust as a good option for addressing the data that does not balance [45]. The use of ROS on sentiment analysis product e-commerce services on text data, posts, and comments, with the use of the BERT algorithm empirically, gives more results OK and is capable of reducing problem imbalance [46].

3. Methods

This research employs an experimental method and an empirical approach in a case study of P2P lending user comment data, obtained through the scraping method of an Android-based application. Fig. 1 is a flow diagram that illustrates a series of analytical procedures, starting with data processing, machine learning in fintech, and progressing to performance evaluation. The Google Play Store service serves as the data source, after which the data undergoes cleaning. These processes include tokenization, stop word removal, normalization, and lemmatization. The next process employs the LDA algorithm to pinpoint topics within a collection of user comment texts, thereby identifying themes within the dataset. Once we identify the topic, we proceed with the document labeling and feature identification processes. The next stage of the classification process uses deep learning CNN. We carried out sampling because the unbalanced dataset for each topic results in a low level of accuracy. We use the ROS-NCL combination as the sampling method. After that, we begin the deep learning classification process, using CNN to classify data and make predictions. We evaluate the performance of the tested models using accuracy, precision, recall, and F1-score metrics.

3.1 Data collection

Data Source in the context of Fintech services: The data source comes from Google Play. Automatic data collection uses “Scaping API (Application Programming Interface) to retrieve P2P lending data in Indonesia which has the most comments, namely PinjamDuit, KreditGo, Finplus, and KoinWork. Data was taken from P2P lending services from January 2022 to January 2023. After cleaning unimportant words and symbols, removing double sentences and deleting more than 4 words per line, which resulted in 16,401 clean data.

3.2 LDA algorithm

This LDA algorithm is used to find topics that appear in the P2P lending text dataset. In the context of P2P lending, it will identify the main topics discussed by users. When data is processed using LDA, data preprocessing must be done, followed by tokenization, stop-words, normalization, and lemmatization, then the process of selecting features and creating a dictionary, creating a bag of words,

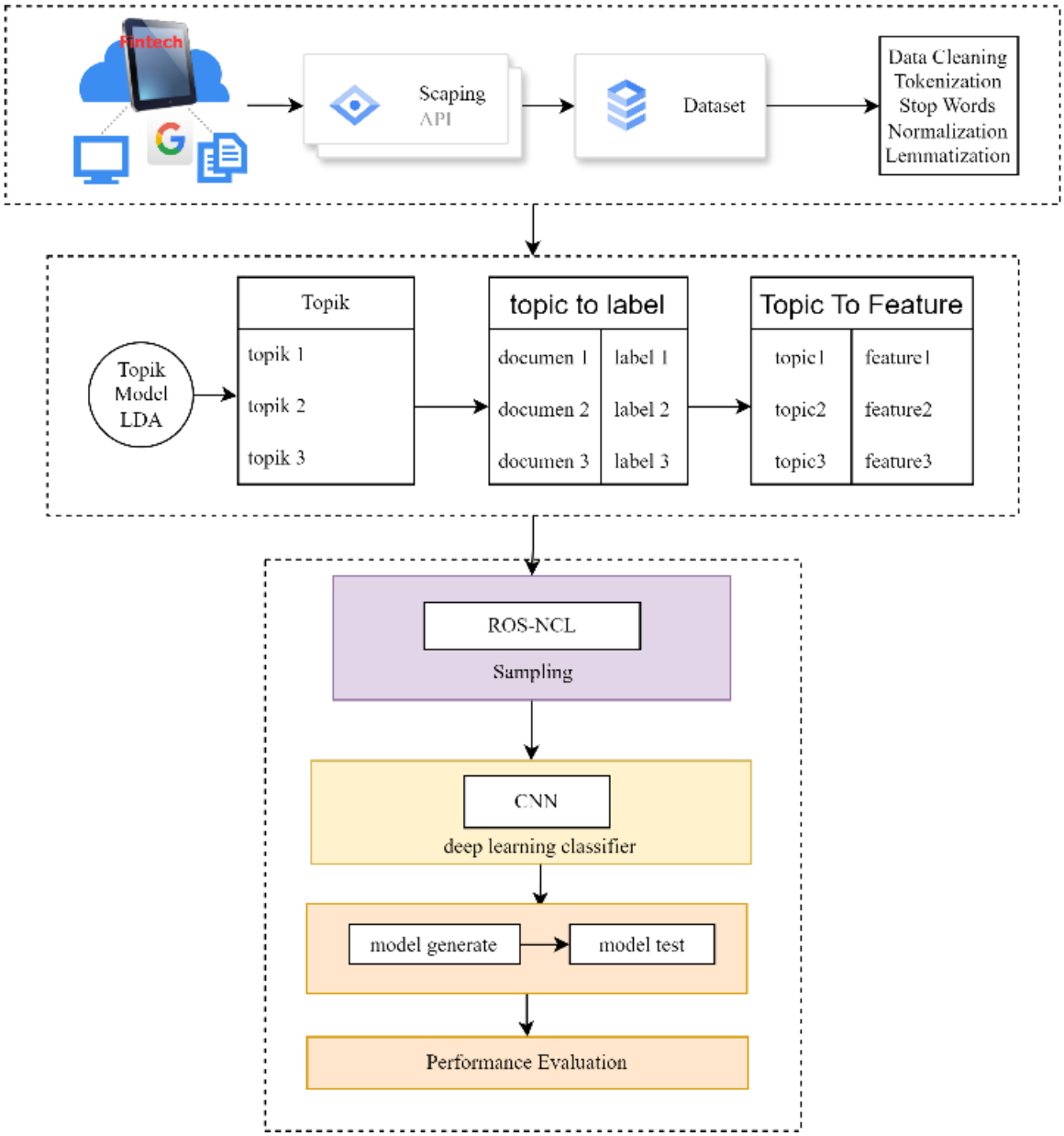

Figure. 1 LDA-CNN Analytical Data Flow

setting LDA parameters, and training the LDA model. Fig. 2 displays a plate diagram of the LDA model [28], which is a topic model in machine learning for processing Fintech user comments in text form.

Variables and Parameters

K: Amount topic in models.

M: Amount document.

N: Number of words in document.

α and β: Dirichlet hyperparameters for distribution topic -per- document and word-per- topic distribution.

θ_d : Topic distribution vector for document d where $\theta_{d,k}$ is the probability of topic k in the document d.

ϕ_k: Word distribution vector for topic k, where $\phi_k$is the probability of word w appear in topic k.

$Z_{d,n}$: Variable topic for the nth word in document d.

$W_{d,n}$: nth word in document d.

Generative Process of LDA model is assumed produce document through the following process for every document d:

Choose $\theta_d \sim$ Dirichlet(α).

For every topic k, choose $\phi_k \sim$ Dirichlet(β).

For every word n in document d:

Choose topic $Z_{d,n} \sim$ Multinomial(θd).

Select the word $W_{d,n} \sim$ Multinomial ($\phi_{Z_{d,n}}$).

Equality from LDA method can defined as following [29] :

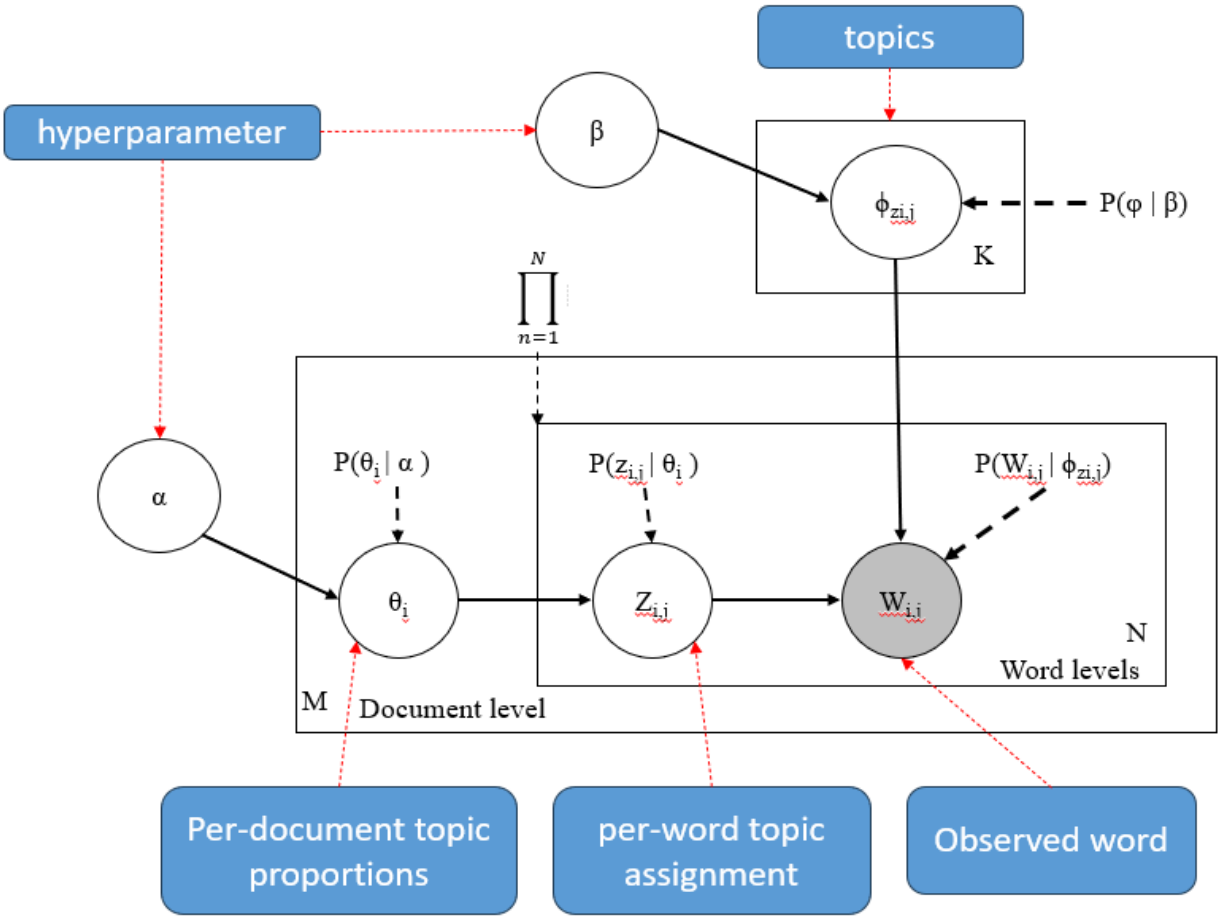

Figure. 2 LDA Model

$$p(w, z|\alpha, \beta) = p(w|\alpha, \beta)\, p(z|\alpha) \quad (1)$$

$p(w, z|\alpha, \beta)$: joint probability of observing word w and topic z with see exists hyperparameters α and β.$p(w|\alpha, \beta)$ is the probability of the word w with see topic z and hyperparameters β, which is related with distribution of words in topic. $p(z|\alpha)$ is the probability of topic z remember hyperparameters α, which influences distribution topic in document.

3.3. CNN algorithm

Fig. 3 represents the CNN algorithm for text. The explanation will focus on notation. The layer that appears most frequently on CNN.

Components main on CNN for classification text use a number of layers, namely embedding layers, convolutional layers, pooling layers (max pooling), fully connected layers, and operations special. For processing, existing text data is changed into a format that can be processed by the network.

Here's a detailed explanation of the CNN layer stages for text classification:

1. Embedding Layer: the layer changes the words inside text become rich vector information semantics. If we own composed text from tokens $t_1, t_2, \ldots.. t_n$ then each token t_i is converted into a vector vi using embedding matrix E. This vector represents the word in room more semantics tall.

$$v_i = \boldsymbol{E}[t_i] \quad (2)$$

E is an embedding matrix where each row is a vector representing one word. Usually, the model training process learns this matrix from the data. t_i: represents the ith token or word within a given sentence or document. This could be an index in NLP that refers to a word in a predefined vocabulary. v_i: embedding vector for word v_i. A numerical representation of the word t_i, taken from matrix $\boldsymbol{E}$. We use this vector to carry semantic information from words, which then serves as input for subsequent layers.

2. Convolutional Layer: convolution layer applies a filter or kernel on the embedding vector for catch feature local from the data. For example, F is a filter with size k and $V_{:i:i+k-1}$ is gathering embedding

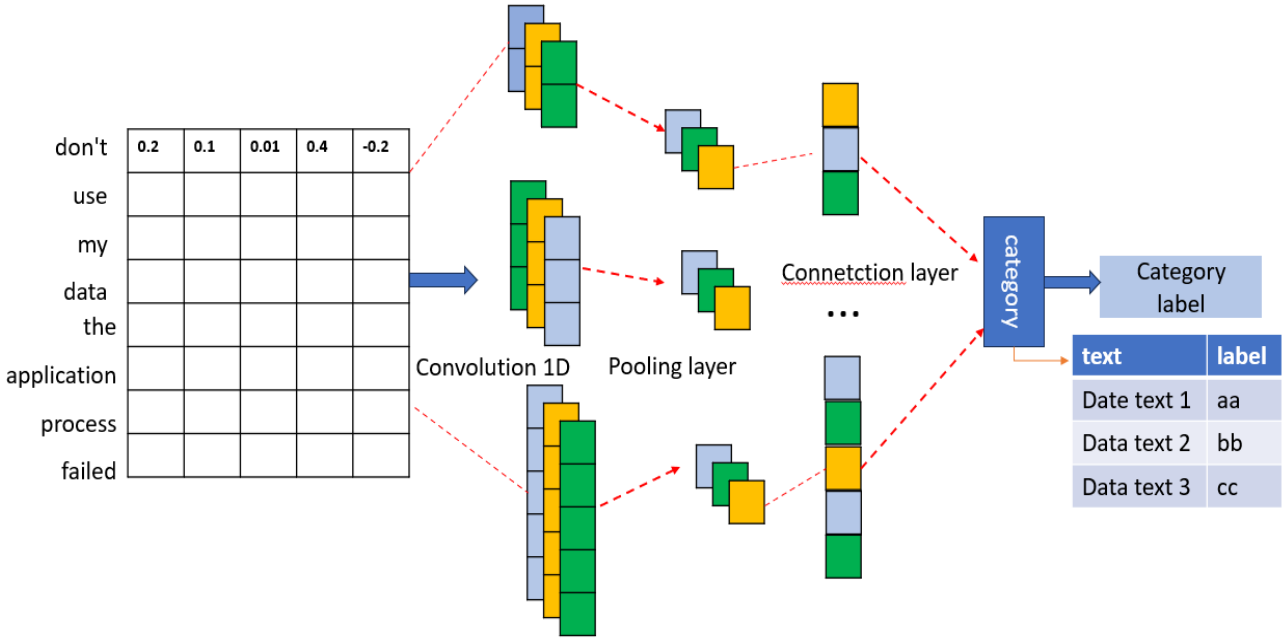

Figure. 3 Classification CNN text

vector of the word i until $i + k - 1$. Operation convolution done as following:

$$\boldsymbol{C}_i = \boldsymbol{f}[F.\ V_{:i:i+k-1} + b] \quad (3)$$

With b is the bias and f is function Activation ReLU, that is $f(x)$=max(0,x)

3. Pooling Layer (Max Pooling): after operation convolution, pooling layers are used for reduce data dimensions with keep information important. Max pooling value maximum taken from every feature that have generated by convolution:

$$p = \boldsymbol{max}(\boldsymbol{c}_1, \boldsymbol{c}_{2,\ldots,}\boldsymbol{c}_m) \quad (4)$$

4. Fully Connected Layer: This layer connects every entry from output pooling to all neurons in the layer, and followed by function Activation like softmax for classification. If $\boldsymbol{p}$ is the output of the pooling layer and W is the weight of the fully connected layer, then the output is:

$$y = softmax(\boldsymbol{W}.\boldsymbol{p} + \boldsymbol{b}') \quad (5)$$

with b' is the vector bias for this layer, and softmax calculated as:

$$softmax(\boldsymbol{Z}_j) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (6)$$

For every component j from logit vector $\boldsymbol{z}$.

3.4 ROS and NCL sampling methods

Random Over Sampler (ROS) is an oversampling technique that handles class imbalance by increasing the number of samples in the minority class. This technique works by taking random samples from the minority class and copying them until they reach equilibrium with the majority class. ROS creates duplicates of existing samples in a minority class to increase the representation of that class in the dataset.

The Neighborhood Cleaning Rule (NCL) is a technique for handling unbalanced data, combining the concepts of undersampling and data cleaning. NCL works by examining each sample in the majority class and removing those that are too close (in terms of characteristics or features) to the minority class. The goal is to clean the "neighborhood" of the minority class so as to minimize classification errors caused by sampling the majority class adjacent to the minority class.

Dealing with datasets that aren't balanced involves various techniques to reduce the number of instances in the class majority (under sampling) or increase the number of instances in the class minority (oversampling). When using a combination of ROS [45, 47] and NCL [48] for data that does not balance, this process aims to balance the distribution class with duplicate class instances and minority instances in a way that shuffles and deletes certain instances from the class majority based on the impact of the nearest neighbor rule.

ROS works with the method of increasing the amount, for example, the minority, in a way random until the amount approaches or is the same with the majority, so the distribution class in the data set becomes more balanced. The ROS technique is required. Because imbalanced classes can have a negative impact on the performance of the learning model machine, Models tend to be biased towards the class majority and produce underpredictions that are inaccurate for the class minority. With the distribution class becoming more balanced, ROS helps increase the accuracy and reliability of the model in classifying minority classes.

Here are the steps to consistently and sequentially implement ROS and NCL using notational mathematics or formulas. For handling data that does not balance with the combination of Random Over Sampler (ROS) and Neighborhood Cleaning Rule (NCL), here is a multi-step process that balances the dataset by adding more lot instances from class minorities and cleaning instances from class the probable majority is noise.

We explain the ROS process in the Fintech P2P Lending feature dataset, which has 10 classes, by adjusting the samples from all minority classes so that the number is proportional to the class that has the maximum number of samples. Let D be a dataset consisting of K classes, with K value=10. Each class k has n_k samples, where k=1,2,..,10.

This process balances the number of samples in all classes so that all classes have a number of samples that is close to the same as the class that has the largest number of samples. Determining the majority class: Identify the class with the maximum number of samples, $n_{max} = \max(n1, n2, \dots n10)$.

For each class, calculate the multiplier factor: For each class k, calculate the multiplier factor f_k which will determine how many times each sample in that class must be duplicated:

$$f_k = \left\lceil \frac{n_{max}}{n_k} \right\rceil \tag{7}$$

The rounding up function is denoted by $\lceil . \rceil$. Sample duplication: For each sample $x_{i,k}$ from class k, duplicate that sample f_k times in the balanced dataset D'.

Resulting dataset: after this process, the number of samples for each class k in dataset D' will be $n'_k = n_k \times f_k$and ideally for k, $n'_k \approx n_{max}$.

Notation D is the initial dataset with 10 feature classes, D' is the dataset after ROS application, n_k is the number of samples in class k before ROS, n'_k: Number of samples in class k after ROS, n_{max} is the maximum number of samples from any class in the dataset D, f_k are multiplier factors for class k.

NCL removes instances from the majority class that are considered to be noise or likely to cause misclassification of minority class samples. This technique utilizes the Nearest Neighbors approach to identify such samples. A Tomek Link is a pair of samples (x_i, x_j) from two different classes where the two samples are nearest neighbors to each other and there are no other samples that are closer to either of them. In other words, there is no third sample x_k such that $d(x_i, x_k) < d(x_i, x_j)$ or $d(x_j, x_k) < d(x_i, x_j)$ where d denotes a distance function (euclidean distance).

Once the ROS process completes, it transitions to the NCL process, which utilizes a dataset consisting of 10 classes. The following are the systematic steps for applying NCL to the 10-class P2P Lending feature dataset.

Step 1: identify the majority and minority classes. The first process identifies which classes are the majority (which have the largest number of samples) and which classes are the minority (which have a smaller number of samples). We can determine this by calculating the number of samples per class.

Given a dataset D with samples $\{x_1, x_2 \dots . x_n\}$ and class labels $\{y_1, y_2 \dots . y_n\}$ with $y_i \in \{1,2, \dots ,10\}$.

Step 2: Applying ENN (Edited Nearest Neighbors) to the majority class, for each sample x_i from the majority class, calculate k-nearest neighbors (using euclidean distance). If the majority of k nearest neighbors have a different label than x_i, remove x_i from the dataset.

Euclidean distance formula:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{m} (x_{ik} - x_{jk})^2} \tag{8}$$

Where m is the number of dimensions.

Step 3 involves identifying and removing Tomek Links for each pair of samples (x_i, x_j), where x_i and x_j are nearest neighbors and belong to different classes.

If there are no samples closer to x_i or x_j than each other, then it forms a Tomek link. To help clarify boundaries between classes, remove samples from the majority of Tomek Link classes.

Step 4: Repeat steps 2 and 3 until it converges (no more samples can be removed), indicating that the boundaries between classes have been optimized.

3.5 Data separation

The data separation process is used to achieve accuracy in performance classification, as well as to reduce problem data sets that are not balanced with ratio separation by 90% for training and 10% for validation and testing. We also apply a validation cross, where we use one data section for developing predictive models and the other for assessing their performance [49].

3.6. Performance evaluation

Model value accuracy shows level accuracy highest during the training process. Prediction the obtained through utilization matrix confusion that categorizes 10 dimensions. Value accuracy, precision, recall, and F-measure then lowered from matrix the same confusion. Accuracy reflects predicted input proportion in a way accurate by the CNN model and shown with decline mark loss. Precision focuses on ratios identified input in a way accurate by the system, whereas acquisition count proportion input that way accurate recognized as Correct. F-measure is the average of precision and recall. Formula For count accuracy, precision, recall, and F-measure are presented in Eqs. (9)-(12), respectively.

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FP+FN+TN)} \quad (9)$$

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

$$\text{Recall} = \frac{TP}{TP+FN} \quad (11)$$

$$\text{F-Measure} = 2\ x\ \frac{Precision\ x\ Recall}{Precision+\ Recall} \quad (12)$$

TP (true positive) indicates predictions accurate from positive data real, TN (true negative) represents predictions accurate from real data negative, FP (false positive) indicates wrong predictions from positive data as positive, and FN (false negative) represents wrong predictions about positive data as negative data.

4. Results and discussion

4.1 Topic construction

Determine amount Optimal topics are pre-processed with cleaning text with stages delete sign read and number, delete character specifically, tokenization with break text into words, removal of stop words i.e., eliminating common words, normalization namely stemming reduces words to form Basically, creating a dictionary, namely a mapping dictionary where each word is unique in the corpus to a unique id, bag-of-words creation, and extensive filtering that deletes the word too often or seldom,

The LDA method for identifying dimension topics in P2P lending This algorithm models the majority of topics used in the analysis [50]. However, the LDA does not specify the number of existing topics in the corpus. If the number of selected topics is too small, it could lead to a significant loss of information on that topic. However, if too many topics are extracted, there may be no useful topics with high similarities [51]. The amount of topics to be extracted must be decided; moreover, formerly, with the use of coherence score analysis, which is also carried out by several researchers, the highest coherence score will be used to determine the number of topics.

Table 1. Results Coherence

Number of topics	Coherence score
1	0.177204
2	0.438963
3	0.442477
4	0.493276
5	0.500895
6	0.444528
7	0.498369
8	0.497913
9	0.467494
10	0.534758

Table 2. LDA Results

Topic1	easy, acc, hopefully, help, application, borrow, can, complicated, enough, recommendation
Topic2	fill, true, not yet, one, difficult, pass, far, thank you, mere, team
Topic3	borrow, goal, increasingly, first, new, smooth, money, worse, million, still
Topic4	can, profit, more, much, get, funds, pay, online, permission, borrow
Topic5	safe, please, good, borrow, start, definitely, delete, answer, continue, name
Topic6	process, time, clear, pay, ID card, borrow, enter, now, wait, where
Topic7	borrow, reject, sell, advance, all, medium, response, period, email, comfortable
Topic8	fast, easy, need, funds, liquid, register, borrow, successful, method, once
Topic9	help, good, capital, business, can, interest, easy, application, believe, money
Topic10	use, results, tenor, community, only, CitizenID, conditions, wrong, complete, account

Table 1 displays the optimal coherence score results for a total of 10 topics, which the research uses to examine the dimensions of P2P lending.

The coherence scores in Table 1 determine the quality and separateness of the topics generated by the topic models. Ten topics achieved the highest coherence scores (0.534758), suggesting that this is the ideal number of topics for the P2P lending dataset.

The coherence scores in Table 1 determine the quality and separateness of the topics generated by the topic models. Ten topics achieved the highest coherence scores (0.534758), suggesting that this is the ideal number of topics for the P2P lending dataset. The coherence score provides insight into the clarity of a topic, with a high score indicating a topic with deep words that are logically related. This score aids in determining the optimal number of topics, preventing too many fragmented or uninformative topics.

It is also useful for comparing and evaluating models, and for selecting the best topic model to optimize insights from comments from P2P lending users. The LDA results produce 10 topics with 10 words each, with the process of determining the topic shown in Table 2.

4.2 Labeling the process document according to the topic

The first step in labeling documents based on topics is to count each word that has been constructed using LDA. Taking topics from the LDA model means taking a list of topics along with words and their weights from the LDA model. Combine all the words from a cleaned document into a single list containing all the words from the document. The next step involves calculating the frequency of each word in the list to determine the total number of occurrences of each word as seen in Table 3.

The Psuedocode 1 in Fig. 4. are shown below. In stage 1, import the library; in stage 2, take the topics generated by the LDA model. The formatted=False parameter directs the return of the topic as a tuple of (word, weight), rather than a formatted string.

Table 3. Count Weight and Number of Words

No	word	topic_id	importance	Word count
0	fast	0	0.2888337	2766
1	process	0	0.076935835	1172
2	help	0	0.06755528	3457
3	borrow	0	0.050982427	10826
4	Enough	0	0.042810813	455

1. Import necessary libraries
a. from collections import Counter
b. import pandas as pd
2. Retrieve topics from the LDA model, topics = lda_ model.show _topics (formatted=False)
3. Flatten the document data
a. Initialize data_flat as an empty list
b. For each word list in doc_clean :
i. For each word in the word list:
- Append the word to data_flat
4. Count word occurrences
a. counter = Counter(data_flat)
5. Initialize output list
a. out = []
6. Iterate through topics and words
a. For each topic index i and topic in topics:
i. For each word and weight in topic:
- Append [word, i, weight, counter[word]] to out
7. Create DataFrame
a. df_imp_wcount = pd.DataFrame (out, columns=['word', ' topic_id ', 'importance', ' word_count '])
8. Print DataFrame
a. print(df_imp_wcount)

Figure. 4 Calculation Pseudocode 1 Amount Emergence

Table 4. Results Labeling Based on Topic

Doc	DT	PC	Topic_Keywords	Text
0	4.0	0.5829	complicated, full, interest, fast, direct, here, most, indeed, borrow, light	[' many ', ' procedure ', ' installment ', ……]
1	1.0	0.3076	borrow, once, a lot, good, online, sell, need, download, withdraw, front	['install', ' story ', ' wa ', 'email', ' now ', …..]

Stage 3 initializes data_flat as an empty list to store all words from all documents, then repeats for each list of words cleaned from documents (doc_clean): for each word in the word list, add a word to data_flat. This step changes from a list to a single list containing all the words. Stage 4 creates a counter object that counts the occurrences of each unique word in the data_flat flattened list. Stage 5 out=[] initializes an empty list to store topics, words, weights, and the number of occurrences. Stage 6 iterates through the topics and words: for each topic (indexed by i) and its corresponding words (contained in the topic): For each word and its associated weight within the topic, append the list [word, i, weight, counter [word]] to the output. It captures the word, its topic ID, its weight in the topic, and the total number of occurrences of the word across all documents.

```
1. Define function format_topics_sentences
( ldamodel =None, corpus= doc_term_matrix,
texts=document)
a. Initialize output list: rows = []
2. Iterate over each document in the corpus
a. For each document index i and topic distribution
row_list in enumerated ldamodel [corpus]:
i. If ldamodel.per_word_topics is True, set
row = row_list [0], else set row = row_list
ii. Sort row by second item of each tuple in
descending order
3. Extract dominant topic information
a. For each topic index j, topic number topic_num,
and topic proportion prop_topic in enumerated sorted
row:
i. If j is 0:
- Retrieve topic words and probabilities using
ldamodel.show _topic ( topic_num )
- Join topic words into a string topic_keywords
- Append [int(topic_num), round(prop_topic, 4),
topic_keywords ] to rows
- Break the loop
4. Create DataFrame
a. sent_topics_df = pd.DataFrame (rows,
columns=[' Dominant_Topic ', ' Perc_Contribution ', '
Topic_Keywords '])
5. Add original text
a. contents = pd.Series (texts, name="Text")
b. sent_topics_df = pd.concat ([ sent_topics_df,
contents], axis=1)
6. Return the DataFrame
a. Return sent_topics_df
7. Call the function and format the DataFrame
a. df_topic_sents_keywords = format_topics_
sentences (ldamodel = lda_model, corpus=
doc_term_matrix, texts= doc_clean)
b. df_dominant_topic = df_topic_sents_
keywords.reset _index ()
c. df_dominant_ topic.columns = [' Document_No
', ' Dominant_Topic ', ' Perc_Contribution ', '
Topic_Keywords ', 'Text']
8. Print the DataFrame
a. printdf_dominant_ topic. head (20))
```

Figure. 5 Pseudocode 2 labeling in documents

Stage 7 converts the output list to a Pandas DataFrame. The DataFrame has columns labeled ‘word’, ‘topic_id’, ‘importance’ (weight of words in the topic), and ‘word_count’ (total word occurrences). Stage 8 prints a data frame, which allows for visually examining the words associated with each topic, their significance within the topic, and their frequency in the overall corpus. Table 3 provides an example of the calculation results.

Stage furthermore is combined with document original. Stages in the labeling process that is initialize data and model, topic dataframe and word count, prepare output list, create dataframe of output list, formatting function topic and sentence, extract topic main for every document, create dataframe for topic and sentence, apply function formatting, seen in pseudocode 2. In Fig. 5 Pseudocode explanation is as follows: Stage 1 establishes the role of the trained LDA model; the corpus serves as the term-document matrix for LDA modeling; the texts are: Original document (text data). In the Output List Initialization stage, rows=[] prepares an initialized empty list to store the output of each document’s dominant topic, along with its contributions and keywords. Stage 2 iterates through each document in the ldamodel[corpus] enumeration corpus; this process produces pairs (index i and topic distribution row_list) for each document in the corpus. We perform a conditional true or false check when handling documents, and then sort the topics by contribution (the second item in each tuple) in descending order to prioritize the most significant topics first. Stage 3: extract dominant topic information iteratively through ordered topics; for each topic distribution, only the first topic (the dominant topic) is considered for output. Fetch and format topic keywords to get the top words and their probability for dominant topics, then format this into a string. In the output, include dominant topic numbers, rounded contributions, and keywords in rows. Stage 4 generates a dataframe, while Stage 5 incorporates the original text into the dataframe by converting it into a Pandas series. This series is then added as a new column to the dataframe, providing a comprehensive view. Stage 6 enriches the dataframe with data on dominant topics and the original text. Stage 7 invokes the DataFrame Function and Format, while Stage 8 outputs the dataframe. Table 4 provides an illustration of the outcomes.With the explanation: Doc: document number, DT: dominant topic, namely the tendency to fall into a certain topic group, PC: Percent Contribution, namely the percentage of contributions to the topic, Topic_Keywords, namely the collection of words that have the highest number, Text is the original sentence from the user’s comment.

4.3 Labeling topic into features

In the feature labeling process, we carry out the process of identifying the features in Fintech by conducting a detailed literature study. After the features are formed, we identify the words that support the features. We choose verbs, nouns, and adjectives that support each feature. After the features are formed, we identify the words that support the features. We select supporting verbs, nouns, and adjectives in each feature. Each feature has an unequal number of supporting words. In table 5 only features are displayed that have their own

```
load new topic_df
load dataset_feature_df
function get_highest_match_ count (topic, dataset_feature_df):
initialize highest_count to 0
initialize best_features to empty string
initialize best_matching_words to empty string
initialize best_feature_name to empty string
split topic into set of words (topic_words)
for each row in dataset_feature_df:
extract kata_pengusun_feature from row
extract feature_name from row
if word_composer_feature is not null:
split kata_pengusun_feature into set of words (kata_pengusun_words)
find matching words between topic_words and kata_pengusun_words
count number of matching words (match_count)

if match_count > highest_count :
update highest_count to match_count
update best_feature to kata_pengusun_feature
update best_matching_words to joined matching words
update best_feature_name to feature_name
return highest_count, best_features, best_matching_words, best_features_name
initialize results as empty list
for each topic in new topic_df ['Topic']:
call get_highest_match_count with topic and dataset_feature_df
append (topic, highest_count, best_feature, best_matching_words, best_feature_name) to results
convert results to DataFrame (results_df)
print results_df
```

Figure. 6 Pseudocode 3. Matching Topic into Features

identification based on topic. The definition feature is supported by a good journal paper. The characteristics of supporting words can be seen in table 5.

After the constituent words topic and organizer feature are obtained, the next process is to enter the identification topic into the feature. Therefore, this process generates documents with labels derived from the features. The topic-based feature detection process involves converting words into vectors and then verifying whether the word is present in the feature. The same goes for words within a feature, which are stored in a particular feature and repeated until the last word in the topic.

The topic-based feature detection steps are shown in pseudocode 3. In Fig. 6, with the following steps: Stage 1 loads the dataset and library. Stage 2 defines the get_highest_match_count function. This function finds the best matching words between the topics from new topic_df and the words that form the features from dataset_feature_df.

Table 5. Words That Make Up Features

Feature	**Feature Word Order**
Convenience use and recommend	easy, advice, optimal, submission, assess, intuitive, acc, efficient, practical, sufficient, reliable, borrow, recommendation, reliability, helpful
Requirements Fulfilled	fill, verify, complete, overcome, achieve, complete, accurate, detailed, complicated, requirements, form, pass, team, process, correct, difficult, selection
Transaction security and information safety	encryption, severe, monitor, borrow, validate, analyze, smooth, secure, objective, secret, vulnerable, protected, encryption, monitor, validate, risk
Profit in a way economy	borrow, facility, save, fast, investment, profit, efficient, can, easy, fast, safe, pay, can, access, efficiency, payment, borrow, online, permission
Data Privacy, Data Recovery and Authorization	clear, help, improve, definitely, support, satisfy, borrow, help, please, safe, answer
response time	fast, minimal, measure, optimal, analyze, pay, enter, efficient, slow, precise, time, immediate, duration, optimization, clear, enter, ID card
caution determine loans to customers	respond, guard, reject, verify, protect, monitor, trust, safe, thorough, borrow, transparent, believe
Convenience service loan	easy, fast, simple, integration, optimal, fluid, access, integration, efficiency, guarantee, loan, consistency, response, reliable, satisfied
Benefits obtained Help business	Increase, easy, connect, simple, application, trust, safe, efficiency, benefit, help, access, connect, secure, support, develop, strong, easy, encourage, protect, support, grow, strong, visibility, protection
Convenience registration	prevent, detect, audit, condition, supervise, prevent, detect, complete, wrong, monitor, vulnerable, account, ID card

Table 6. Topic To Feature

Topics	Highest Match Count	Match Words from topics to features	Best Feature Name
easy, acc, hopefully, help, application, borrow, can, complicated, enough, recommendation	5	borrow, acc, enough, easy, application	Feature 1
fill, true, not yet, one, difficult, pass, far, thank you, mere, team	3	correct, pass, fill	Feature 2
borrow, goal, increasingly, first, new, smooth, money, worse, million, still	4	smooth, borrow, aim, severe	Feature 3
can, profit, more, much, get, funds, pay, online, permission, borrow	4	profit, can, can, online	Feature 4
safe, please, good, borrow, start, definitely, delete, answer, continue, name	5	borrow, please, safe, delete, sure	Feature 5
process, time, clear, pay, ID card, borrow, enter, now, wait, where	4	clear, pay, enter, time	Feature 6
borrow, reject, sell, advance, all, medium, response, period, email, comfortable	3	respond, borrow, reject	Feature 7
fast, easy, need, funds, liquid, register, borrow, successful, method, once	4	fast, loan, easy, liquid	Feature 8
help, good, capital, business, can, interest, easy, application, believe, money	4	application, easy, trust, help	Feature 9
use, results, tenor, community, only, CitizenID, conditions, wrong, complete, account	4	account, terms, complete	Feature 10

Stage 3 iterates over the topics and records the results. The following code iterates over each topic in new topic_df and applies the get_highest_match_count function, then stores the results.

Pseudocode 3 processes topics to features from two CSV files, namely table 2 and table 5, to identify the most relevant word matches and presents the results in a structured format. Pseudocode 3 matches words from topics with words in the feature table. This pseudocode bridges topics to features, so that the dataset can be labeled as features. The Fig. 6 illustrates the integration of feature labeling results into document labeling.

4.4 Experiment and imbalanced data handling

Result of processing the topic into features, resulting in 10 features. The main one has an amount that varies, consisting of feature 1 totaling 4472, feature 2 totaling 233, feature 3 totaling 969, feature 4 totaling 1940, feature 5 totaling 1476, feature 6 totaling 776, feature 7 totaling 1737, feature 8 totaling 2561, feature 9 totaling 1848, and feature 10 totaling 389.

4.5 ROS and NCL

ROS performs over-sampling to add a sample in the minority class with the method of making duplicates in a way random until it reaches balance. NCL reduces the sample in class to the majority; NCL under-sampling considers information neighboring the sample; eliminates the sample from class; the majority is considered "noisy" or close to it with the class minority. ROS stage creates an instance by applying oversampling, generating a new dataset with a more balanced distribution. After the ROS process stages, furthermore, create an instance of NCL. Perform under-sampling on a dataset that has been over-sampled. This matter aims to remove "noisy" or potentially "noisy" samples that cause confusion on models and produce more clean and balanced datasets.

Furthermore, divide the dataset into training data and test data. Divide the resampled dataset into two sets: a training set and a test set. train_size =0.9 shows that 90% of the data will be used for training and the rest for testing. Dimensions from the training data set as following

(33431, 250) (33431, 10)
(3715, 250) (3715, 10)

First output show that x_train has 33,431 samples with each sample has 250 features. Meanwhile, y_train also has 33,431 samples, each sample represented as vector with 10 elements or target class for classification. Second output show that x_test has 3,715 samples, with each having 250 features, similar with x_train. y_test has 3,715 samples with same target representation like y_train, i.e vector with 10 elements per sample.

4.6 CNN model implementation

After the resampling process, the next step is using CNN for the classification feature. The classification results feature from the CNN model in

Table 7. Results Accuracy

Feature	Precision	Recalls	F1-score	Support
0-0	0.87	0.74	0.80	461
1-0	0.98	1.00	0.99	475
2-0	0.96	0.99	0.98	430
3-0	0.94	0.94	0.94	284
4-0	0.98	0.98	0.98	393
5-0	0.98	1.00	0.99	418
6-0	0.97	0.98	0.97	327
7-0	0.79	0.86	0.83	200
8-0	0.92	0.94	0.93	299
9-0	0.98	1.00	0.99	439
accuracy			0.95	3726
avg macros	0.94	0.94	0.94	3726
weighted avg	0.95	0.95	0.95	3726

Table 7 produces 10 different classes that include metric precision (precision), recall, and F1 score for each class, as well as accuracy overall, macro average, and weighted average.

Class 0-0: shows that the model is sufficient for identifying class this, but there is room for improvement, especially in recognizing all positive examples (recall). Classes 1-0 to 9-0: Most classes have very high precision and recall, close to 1.00, which shows excellent performance from the inner model. Class 7-0 shows that this model more often makes mistakes predict class This as another class (precision lower), but still enough good in identifying cases positive (more recalls tall).

The model accuracy reached 0.95, meaning the model was successful in classifying 95% of sample testing as correct. Macro Average (Macro Avg) shows strong performance throughout the class. Weighted Average (Weighted Avg) confirms that the model does a very good job in a way that considers size for every class.

The performance of the CNN model has been trained with good and capable generalization from training data to test data with high performance. That model shows a good balance between the ability to identify class positives and minimizing errors in classification.

4.7 Testing with other algorithm

Table 8 is a comparison of the performance of various sampling models. The CNN-ROS-NCL algorithm has the best performance among CNN-based models, with the highest precision, recall, and F1-score (0.94), as well as the highest accuracy (94.66%) and the lowest level error classification (5.34%).

Table 8. Results CNN Testing

Algorithm	**Precision**	**Recall**	**F1-score**	**Accuracy**
CNN-ROS-NCL-SMOTE	0.87	0.87	0.87	87.52%
CNN-ROS-NCL	0.94	0.94	0.94	94.66%
CNN-ROS-SMOTE-NCL	0.92	0.94	0.93	93.43%
CNN-ROS-ADASYN-NCL	0.93	0.93	0.93	93.81%
CNN-ROS	0.88	0.88	0.88	88.39%
CNN	0.38	0.38	0.38	44.25%

Table 9. Results LSTM Testing

Algorithm	**Precision**	**Recall**	**F1-Score**	**Support**	**Accuracy**
LSTM-ROS-NCL-SMOTE	0,88	0,87	0,87	3719	88,53
LSTM-ROS-NCL	0,88	0,88	0,88	3722	89,71
LSTM-ROS-SMOTE-NCL	0,86	0,86	0,86	3718	87,57
LSTM-ROS-ADASYN-NCL	0,86	0,87	0,86	3724	87,65
LSTM-ROS	0,83	0,83	0,83	4472	82,09
LSTM	0,54	0,45	0,48	1641	51,49

Table 10. Results CNN-LSTM Testing

Algorithm	**Precision**	**Recall**	**F1-Score**	**Support**	**Accuracy**
CNN-LSTMROS-NCL-SMOTE	0.83	0.84	0.83	4472	83,56
CNN-LSTM-ROS-NCL	0.88	0.88	0.88	3721	89,60
CNN-LSTM-ROS-SMOTE-NCL	0.89	0.89	0.89	3722	90,11
CNN-LSTM-ROS-ADASYN-NCL	0.89	0.88	0.88	3722	89,68
CNN-LSTM-ROS	0.84	0.84	0.84	4472	84,44
CNN-LSTM	0.48	0.41	0.42	1641	49,36

Table 11. Accuracy Feature

Feature	description	Information feature	Accuracy
Feature 1	Focus on easy application process, accessibility, and recommendations, in context service loan finance.	Convenience use and recommend [52].	74.19%
Feature 2	Related with charging form or application with right, trouble in process, and aspirations to pass the selection.	Requirements Fulfilled [53].	100%
Feature 3	Relating to goals loan increasingly critical millions of moneys in trouble, experience borrow For First smoothly, and the amount of money.	Transaction security and information safety [54].	99.30%
Feature 4	Discuss about profit from loans, convenience access funds, and payments by online.	Profit in a way economy [55].	94.37%
Feature 5	Refers to aspect security Difficulty Deletion of data by user deletion name, help, and service Good in the loan process.	Data Privacy, Data Recovery and Authorization [56-57].	98.22%
Feature 6	Focus on administrative processes loans, incl time required and documents required (such as ID card).	Response time [53].	99.76%
Feature 7	Covers rejection application loan, application process, and response from giver loan.	Caution determines loans [58].	97.86%
Feature 8	Focuses on speed and convenience in get loan funds.	Service quality [59].	86.50%
Feature 9	Related with help in get business capital, applications used, and interest loan.	Benefits obtained Help business /Benefits [60-61]	93.98%
Feature 10	Explain convenience condition loan for community, and the data verification process	Convenience registration [62]	99.77%

Based on Table 8, it is evident that the CNN-ROS-NCL algorithm is the most effective for further stages. Algorithm This algorithm achieves the highest score in precision, recall, and F1, specifically 0.94, indicating its superior performance. This is very good at classifying data systematically and correctly. Support shows the amount of sample used for count metrics, and for CNN-ROS-NCL, the amount is 3726. Algorithm This level has the highest accuracy (94.66%) and the lowest rate of misclassification (5.34%). High accuracy means an algorithm This capable algorithm can identify class labels correctly in large proportions from the overall data, while the low misclassification rate shows that this algorithm often makes mistakes when applying class labels. Based on this data, the CNN-ROS-NCL algorithm is capable of showing the best balance between accuracy classification and error classification, which is an indicator of excellent model performance in practice.

In this research, the results of the CNN-ROS-NCL algorithm were also compared with the LSTM and CNN-LSTM algorithms with various techniques for handling unbalanced data by looking at the precision, recall, F1-Score, and accuracy metrics. The comparison of the LSTM algorithm can be seen in Table 9. The results of the LSTM test, LSTM-ROS-NCL, gave the best results, with an accuracy of 89.71% and a precision, recall, and F1-score value of 0.88. This shows a significant improvement compared to LSTM without balancing techniques, which only achieved an accuracy of 51.49% with a precision value of 0.54 and a recall of 0.45. This shows that the ROS-NCL technique effectively improves the performance of LSTM in dealing with imbalanced data. A comparison of the CNN-LSTM algorithm combination can be seen in Table 10. The test results with the CNN-LSTM combination show that CNN-LSTM-ROS-SMOTE-NCL provides the best performance with an accuracy of 90.11% and a precision, recall, and F1-score of 0.89. This shows good synergy between CNN and LSTM in capturing temporal and spatial features of data, with improved performance through the use of ROS, SMOTE, and NCL techniques. CNN-ROS-NCL shows the best results among the three algorithms, with the highest accuracy (94.66%). This algorithm not only excels in precision and recall but also offers high consistency across metrics, making it the best choice for situations that require a very high level of accuracy. The combination of CNN-LSTM with the use of data balancing techniques shows very good and consistent results, but is still slightly below the best performance shown by CNN with ROS-NCL.LSTM also shows good improvement with balancing techniques but is not as strong as the performance of CNN or the CNN-

LSTM combination. Thus, CNN-ROS-NCL overall offers the best performance for unbalanced data processing compared to LSTM and even the CNN-LSTM combination. This shows that, for certain tasks and unbalanced data conditions, the use of CNNs with appropriate balancing strategies (ROS-NCL) can be more effective than other techniques. Such as research that uses the NCL method as a sampling method which can increase accuracy [43-44]. The ROS method has good performance for dealing with imbalanced data [45] in sentiment analysis of e-commerce service products [46].

In table 11 we can see that every feature has a high level of accuracy. Topic 1, which pertains to convenience usage and recommendations, exhibits a level of accuracy of 74.19%, indicating that these aspects are significant in P2P lending. Can be predicted with sufficient accuracy. Topic 2 is about how the can model fulfills requirements. Conduct a thorough and productive study with a high accuracy rate of 100%, ensuring correct charging based on a close relationship with the quality of the input data. Perfect accuracy This demonstrates that the data is clear and consistent when utilized. Can produce very accurate predictions. Topic 3 about transaction security and information safety has a high accuracy (99.30%), indicating that the model learns important characteristic features from related security and usage data, which helps the model identify patterns of fraud or fraud from minority data. Secure transactions and information are critical in P2P lending. Topic 4 Profit The model studies the profit economy and online access to funds in P2P lending, as evidenced by its high accuracy of 94.37%.

Topic 5, which focuses on Authorization and Data Recovery, highlights the importance of these issues in P2P lending. This model is capable of identifying and predicting its 98.22% accuracy issues. The difficulty of data deletion and aspects of security show the importance of the data. Topic 6: Time Wait: Demonstrate that the model is capable of predicting and dealing with time-related issues. The model should be able to wait for P2P lending with an accuracy rate of 99.76%. Topic 7 on Trust and Caution shows that trust is a main component of P2P lending. The model can accurately predict rejection applications, demonstrating success in understanding factors and trust, with an accuracy rate of 97.86%. Topic 8 on service quality demonstrates that the model can analyze variants with a high level of service experience, resulting in an accuracy level of 86.50%. Topic 9 about the benefits obtained shows that the benefits obtained from P2P loans are clear and specific, so help in model learning with 93.98% accuracy. Topic 10 on data misuse concerns shows issues of misuse of data, verification, and privacy. Data security is very critical and specific, making it easier to model studies with high accuracy and a value of 99.77%.

LDA is able to provide good representation by generating comment topics from Fintech P2P Lending users which can be used for the next process, namely feature-based labeling. LDA is able to build topics from documents as a mixture of topics [15-17]. By utilizing LDA it can be used to label documents based on the resulting topics by measuring the accuracy of the number of topics using coherence [32-33]. The resulting topics are used to synchronize features, so that documents can be labeled based on features. Documents that have been labeled as features are used to carry out classification using CNN-ROS-NCL, producing high accuracy, so that this model is able to carry out feature classification well.

Overall, the level of accuracy produced by CNN-ROS-NCL reflects the algorithm's excellent ability to understand and classify various aspects of P2P lending. Data that is clear, consistent, and highly relevant to aspects of user comments on financial technology results in high accuracy.

Automatic feature labeling produces 10 features that are relevant for P2P lending users, namely ease of use and recommendation, requirements fulfilled, transaction security and information safety, economic benefits, data privacy, data recovery and authorization, response time, prudence in determining loans, service quality, help and benefits in business, easy registration. Each feature has its own level of accuracy. The high accuracy shows that the model is able to effectively understand and classify various P2P lending features using CNN-ROS-NCL, with accuracy, precision, recall of 94.66%, and the highest F1 score among the various algorithms tested.

5. Conclusion

This research shows that the analytical model in Fintech P2P lending using LDA, with classification using CNN, is an effective approach in terms of coherence and accuracy values in identifying and evaluating important features from user comments on the Fintech P2P lending platform, such as ease of use, security and benefits, and the and the economics of user comments. Handling imbalanced data using the ROS and NCL methods has also been proven to improve the performance of the classification model. Implementing the features identified in this research can help observe features that need attention so that they can improve P2P lending fintech services,

reduce risks, and increase user satisfaction. The CNN model achieved 95% accuracy, with precision and recall scores above 0.94 for most classes. This research makes an important contribution to the topic modeling of fintech P2P lending user comments, offering a new and effective analytical approach to address imbalanced data and improve feature classification.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, first, second authors; methodology, first, second authors; software first, second authors, validation, first, second, and third authors; formal analysis, first, second, and third authors; investigation, second and third authors; resources, first authors; data curation, first authors; writing original draft preparation, first, second and third authors; writing review and editing, first, second and third authors; visualization, first, second and third authors; supervision, second and third authors; project administration, first, second and third authors.

Acknowledgments

Thank you to Diponegoro University, especially the Doctor Information Systems Program, which provided facilities and resources for this research. Thank you to Soegijapranata Catholic University for supporting this publication.

References

[1] D. Ramamonjisoa, "Topic modeling on users' comments", In: *Proc. of 2014 Third ICT International Student Project Conference (ICT-ISPC)*, IEEE, pp. 177-180, 2014, doi: 10.1109/ICT-ISPC.2014.6923245.

[2] D. Zhang, X. Liu, J. Li, and M. Fan, "Sentiment topic mining based on comment tags", *IOP Conf. Ser. Mater. Sci. Eng.*, Vol. 322, No. 5, 2018, doi: 10.1088/1757-899X/322/5/052048.

[3] S. A. Basha, M. M. Elgammal, and BM Abuzayed, "Online peer-to-peer lending: A review of the literature", *Electron. Commer. Res. Appl.*, Vol. 48, No. 2020, p. 101069, 2021, doi: 10.1016/j.elerap.2021.101069.

[4] A. Milne and P. Parboteeah, "The Business Models and Economics of Peer-to-Peer Lending", *SSRN Electron. J.*, 2016, doi: 10.2139/ssrn.2763682.

[5] Z. Liu, J. Shang, S. Wu, and P. Chen, "Social collateral, soft information and online peer-to-peer lending: A theoretical model", *Eur. J. Opera. Res.*, Vol. 281, No. 2, pp. 428-438, 2020, doi: 10.1016/j.ejor.2019.08.038.

[6] N. Barasinska and D. Schäfer, "Is Crowdfunding Different? Evidence on the Relation between Gender and Funding Success from a German Peer-to-Peer Lending Platform", *Ger. Econ. Rev.*, Vol. 15, No. 4, pp. 436-452, 2014, doi: 10.1111/geer.12052.

[7] W. T. Nakamura, E. Cesar, D. Oliveira, E. H. T. De Oliveira, D. Redmiles, and T. Conte, "What factors affect the UX in mobile apps? A systematic mapping study on the analysis of app store reviews ☆", *J. Syst. Softw.*, Vol. 193, p. 111462, 2022, doi: 10.1016/j.jss.2022.111462.

[8] P. V. Law, L.C. Schaik, "Modelling user experience - An agenda for research and practice", *Interact. Comput.*, Vol. 22, pp. 313-322, 2010, doi: 10.1016/j.intcom.2010.04.006.

[9] Y. Han and M. Moghaddam, "Analysis of sentiment expressions for user-centered design", *Expert Syst. Appl.*, Vol. 171, No. 2019, p. 114604, 2021, doi: 10.1016/j.eswa.2021.114604.

[10] D. Hoang, B. Phan, P. Kumar, R. E. Rahman, and A. R. Hutabarat, "Do financial technology firms influence bank performance?", *Pacific-Basin Financ. J.*, Vol. 62, No. 2018, p. 101210, 2020, doi: 10.1016/j.pacfin.2019.101210.

[11] K. Gai, M. Qiu, and X. Sun, "A survey on FinTech", *J. Netw. Comput. Appl.*, Vol. 103, No. 2017, pp. 262-273, 2018, doi: 10.1016/j.jnca.2017.10.011.

[12] S. Agarwal and Y. H. Chua, "FinTech and household finance: a review of the empirical literature", *China Financ. Rev. Int.*, Vol. 10, No. 4, pp. 361-376, 2020, doi: 10.1108/CFRI-03-2020-0024.

[13] D. Sharma, W. Jerene, and A. Minch, "The Effect of e-Finance Service Quality on Bank Customers' Fintech e-Loyalty : Evidence from Ethiopia", *International Journal of E-Business Research (IJEBR)*, Vol. 16, No. 2, pp. 69-83, 2020, doi: 10.4018/IJEBR.2020040105.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation", *The Journal of Machine Learning Research (JMLR)*, Vol. 3, pp. 993-1022, 2003.

[15] B. Ozyurt and M. A. Akcayol, "A new topic modeling based approach for aspect extraction in aspect based sentiment analysis: SS-LDA", *Expert Syst. Appl.*, No. November, p. 114231, 2020, doi: 10.1016/j.eswa.2020.114231.

[16] A. Reyes-Menendez, J. R. Saura, and F. Filipe, "Marketing challenges in the #MeToo era: gaining business insights using an exploratory

sentiment analysis", *Heliyon*, Vol. 6, No. 3, 2020, doi: 10.1016/j.heliyon.2020.e03626.

[17] A. García-Pablos, M. Cuadros, and G. Rigau, "W2VLDA: Almost unsupervised system for Aspect Based Sentiment Analysis", *Expert Syst. Appl.*, Vol. 91, pp. 127-137, 2018, doi: 10.1016/j.eswa.2017.08.049.

[18] J. A. Wahid *et al.*, "Topic2Labels: A framework to annotate and classify the social media data through LDA topics and deep learning models for crisis response", *Expert Syst. Appl.*, Vol. 195, No. January, p. 116562, 2022, doi: 10.1016/j.eswa.2022.116562.

[19] X. Li, J. Ouyang, and X. Zhou, "Centroid prior topic model for multi-label classification", *Pattern Recognit. Lett.*, Vol. 62, pp. 8-13, 2015, doi: 10.1016/j.patrec.2015.04.012.

[20] C. Reid Turner, A. Fuggetta, L. Lavazza, and A.L. Wolf, "A conceptual basis for feature engineering", *J. Syst. Softw.*, Vol. 49, No. 1, pp. 3-15, 1999, doi: 10.1016/S0164-1212(99)00062-X.

[21] T. Chua, "Attentive Aspect Modeling for Review-Aware", *Tois*, Vol. 37, No. 3, 2019.

[22] D. O'Callaghan, D. Greene, J. Carthy, and P. Cunningham, "An analysis of the coherence of descriptors in topic modeling", *Expert Syst. Appl.*, Vol. 42, No. 13, pp. 5645-5657, 2015, doi: 10.1016/j.eswa.2015.02.055.

[23] S. Syed and M. Spruit, "Full-Text or Abstract? Examining Topic Coherence Scores Using Latent Dirichlet Allocation", In: *Proc. of 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 165-174, 2017, doi: 10.1109/DSAA.2017.61.

[24] L. Yu and N. Zhou, "Survey of Imbalanced Data Methodologies", *arXiv Cornell University*, 2021, [Online]. Available: http://arxiv.org/abs/2104.02240

[25] S. Budania, T. Kumar, H. Kumar, and G. Nikam, "Hybrid Machine Intelligence for Imbalanced Data", *SSRN Electron. J.*, pp. 1-6, 2020, doi: 10.2139/ssrn.3602531.

[26] D. M. Blei, L. Carin, and D. Dunson, "Probabilistic topic models", *IEEE Signal Process. Mag.*, Vol. 27, No. 6, pp. 55-65, 2010, doi: 10.1109/MSP.2010.938079.

[27] M. Lamba and M. Madhusudhan, "Author-topic modeling of DESIDOC Journal of Library and Information Technology (2008-2017), India", *Libr. Philos. Pract.*, Vol. 2019, No. May, 2019.

[28] Y. Luo, T. Tong, "Exploring destination image through online reviews: an augmented mining model using latent Dirichlet allocation combined with probabilistic hesitant fuzzy algorithm", *Kybernetes*, Vol. 52, No. 3, pp. 874-897, 2023, doi: 10.1108/K-07-2021-0584.

[29] Y. Fu, M. Yan, X. Zhang, L. Xu, D. Yang, and J.D. Kymer, "Automated classification of software change messages by semi-supervised Latent Dirichlet Allocation", *Inf. Softw. Technol.*, Vol. 57, No. 1, pp. 369-377, 2015, doi: 10.1016/j.infsof.2014.05.017.

[30] Y. Wang, Y. Tong, and D. Shi, "Federated latent dirichlet allocation: A local differential privacy based framework", In: *Proc. of AAAI 2020 - 34th AAAI Conf. Artif. Intel.*, pp. 6283-6290, 2020, doi: 10.1609/aaai.v34i04.6096.

[31] D. O'Callaghan, D. Greene, J. Carthy, and P. Cunningham, "An analysis of the coherence of descriptors in topic modeling", *Expert Syst. Appl.*, Vol. 42, No. 13, pp. 5645-5657, 2015, doi: 10.1016/j.eswa.2015.02.055.

[32] S. Syed and M. Spruit, "Full-Text or abstract? Examining topic coherence scores using latent dirichlet allocation", In: *Proc. of 2017 Int. Conf. Data Sci. Adv. Anal. DSAA 2017*, Vol. 2018, pp. 165-174, 2017, doi: 10.1109/DSAA.2017.61.

[33] D. Chehal, P. Gupta, and P. Gulati, "Implementation and comparison of topic modeling techniques based on user reviews in e-commerce recommendations", *J. Ambient Intell. Humaniz. Comput.*, Vol. 12, No. 5, pp. 5055-5070, 2021, doi: 10.1007/s12652-020-01956-6.

[34] S. M. Kamruzzaman, F. Haider, and A. R. Hasan, "Text Classification using Data Mining", *arXiv Cornell University*, 2010, [Online]. Available: http://arxiv.org/abs/1009.4987

[35] K. Nithya, P. C. D. Kalaivaani, and R. Thangarajan, "An enhanced data mining model for text classification", In: *Proc. of 2012 International Conference on Computing, Communication and Applications*, IEEE, 2012, pp. 1-4. doi: 10.1109/ICCCA.2012.6179179.

[36] V. K. Vijayan, K. R. Bindu, and L. Parameswaran, "A comprehensive study of text classification algorithms", In: *Proc. of 2017 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2017*, Vol. 2017, pp. 1109-1113, 2017, doi: 10.1109/ICACCI.2017.8125990.

[37] X. Zhou *et al.*, "A survey on text classification and its applications", *Web Intell.*, Vol. 18, No. 3, pp. 205-216, 2020, doi: 10.3233/WEB-200442.

[38] G. Lu, Y. Liu, J. Wang, and H. Wu, "CNN-BiLSTM-Attention: A multi-label neural classifier for short texts with a small set of labels", *Inf. Process. Manag.*, Vol. 60, No. 3, p. 103320, 2023, doi: 10.1016/j.ipm.2023.103320.

[39] H. Wang, J. He, X. Zhang, and S. Liu, "A Short Text Classification Method Based on N -Gram

and CNN", *Chinese J. Electron.*, Vol. 29, No. 2, pp. 248-254, 2020, doi: 10.1049/cje.2020.01.001.

[40] X. Li and H. Ning, "Chinese text classification based on hybrid model of CNN and LSTM", In: *Proc. of ACM Int. Conf. Proceedings Ser.*, 2020, doi: 10.1145/3414274.3414493.

[41] Z. Tan, J. Chen, Q. Kang, M. Zhou, A. Abusorrah, and K. Sedraoui, "Dynamic Embedding Projection-Gated Convolutional Neural Networks for Text Classification", *IEEE Trans. Neural Networks Learn. Syst.*, Vol. 33, No. 3, pp. 973-982, 2022, doi: 10.1109/TNNLS.2020.3036192.

[42] L. Li, Z. Zhu, D. Du, S. Ren, Y. Zheng, and G. Chang, "Adversarial Convolutional Neural Network for Text Classification", In: *Proc. of the 2020 4th International Conference on Electronic Information Technology and Computer Engineering*, New York, NY, USA: ACM, 2020, pp. 692-696. doi: 10.1145/3443467.3443837.

[43] S. Saadah and G. S. Wulandari, "Tackling Class Imbalance Problem in Binary Classification using Augmented Neighborhood Cleaning Algorithm", *Lect. Notes Electr. Eng.*, Vol. 339, No. 2016, pp. 777-784, 2015, doi: 10.1007/978-3-662-46578-3.

[44] K. Agustianto and P. Destarianto, "Imbalance Data Handling using Neighborhood Cleaning Rule (NCL) Sampling Method for Precision Student Modeling", In: *Proc. of 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE)*, IEEE, 2019, pp. 86-89. doi: 10.1109/ICOMITEE.2019.8921159.

[45] F. Kamalov, H. H. Leung, and A. K. Cherukuri, "Keep it simple: random oversampling for imbalanced data", In: *Proc. of 2023 Advances in Science and Engineering Technology International Conferences (ASET)*, IEEE, 2023, pp. 1-4. doi: 10.1109/ASET56582.2023.10180891.

[46] A. K. Jayaraman, A. Murugappan, T. E. Trueman, G. Ananthakrishnan, and A. Ghosh, "Imbalanced aspect categorization using bidirectional encoder representation from transformers", *Procedia Comput. Sci.*, Vol. 218, pp. 757-765, 2022, doi: 10.1016/j.procs.2023.01.056.

[47] J. M. Johnson and T. M. Khoshgoftaar, "The Effects of Data Sampling with Deep Learning and Highly Imbalanced Big Data", *Inf. Syst. Front.*, Vol. 22, No. 5, pp. 1113-1131, 2020, doi: 10.1007/s10796-020-10022-7.

[48] N. O. Al-Abdouli, Z. Aung, W. L. Woon, and D. Svetinovic, "Tackling class imbalance problem in binary classification using augmented Neighborhood cleaning algorithm", *Lect. Notes Electr. Eng.*, Vol. 339, pp. 827-834, 2015, doi: 10.1007/978-3-662-46578-3_98.

[49] E. W. Steyerberg, "Validation in prediction research: the waste by data splitting", *J. Clin. Epidemiol.*, Vol. 103, pp. 131-133, 2018, doi: 10.1016/j.jclinepi.2018.07.010.

[50] F. Barravecchia, L. Mastrogiacomo, and F. Franceschini, "Digital voice-of-customer processing by topic modeling algorithms: insights to validate empirical results", *Int. J. Qual. Reliab. Manag.*, Vol. 39, No. 6, pp. 1453-1470, 2022, doi: 10.1108/IJQRM-07-2021-0217.

[51] J. Gan and Y. Qi, "Selection of the optimal number of topics for LDA topic model—Taking patent policy analysis as an example", *Entropy*, Vol. 23, No. 10, 2021, doi: 10.3390/e23101301.

[52] I. Bagus, R. Suardana, L. Kadek, B. Martini, N. Sri, and M. Setini, "Quality factors in technology system capability decision interest in transactions using mobile banking", *International Journal of Data and Network Science*, Vol. 6, pp. 1-8, 2022, doi: 10.5267/j.ijdns.2021.11.003.

[53] P. K. Aggarwal, P. S. Grover, and L. Ahuja, "Evaluating self-management features for mobile applications", *Int. J. E-Services Mob. Appl.*, Vol. 11, No. 2, pp. 43-55, 2019, doi: 10.4018/IJESMA.2019040103.

[54] M. Jun and S. Palacios, "Examining the key dimensions of mobile banking service quality: an exploratory study", *Int. J. Bank Mark.*, Vol. 34, No. 3, pp. 307-326, 2016, doi: 10.1108/IJBM-01-2015-0015.

[55] J. K. Park, J. Ahn, T. Thavisay, and T. Ren, "Examining the role of anxiety and social influence in multi-benefits of mobile payment services", *J. Retail. Consume. Serv.*, Vol. 47, No. 2018, pp. 140-149, 2019, doi: 10.1016/j.jretconser.2018.11.015.

[56] M. A. Nangin, I. Rasita, G. Barus, and S. Wahyoedi, "The Effects of Perceived Ease of Use, Security, and Promotion on Trust and Its Implications on Fintech Adoption", *Journal of Consumer Sciences*, Vol. 5, No. 2, pp. 124-138, 2020.

[57] B. H. Leem and S. W. Eum, "Using text mining to measure mobile banking service quality", *Ind. Manag. Data Syst.*, Vol. 121, No. 5, pp. 993-1007, 2021, doi: 10.1108/IMDS-09-2020-0545.

[58] H. Jason Huang and A. Dastmalchian, "Implications of trust and distrust for

organizations", *Personnel Review*, Vol. 35, No. 4. pp. 361-377, 2006, doi: 10.1108/00483480610670553.

[59] D. Chen, F. Lai, and Z. Lin, "A trust model for online peer-to-peer lending: a lender's perspective", *Inf. Technol. Manag.*, Vol. 15, No. 4, pp. 239-254, 2014, doi: 10.1007/s10799-014-0187-z.

[60] O. Dospinescu, N. Dospinescu, and D. T. Agheorghiesei, "Fintech services and factors determining the expected benefits of users: Evidence in romania for millennials and generation Z", *E a M Ekon. a Manag.*, Vol. 24, No. 2, pp. 101-118, 2021, doi: 10.15240/tul/001/2021-2-007.

[61] H. S. Ryu, "What makes users willing or hesitant to use Fintech?: The moderating effect of user type", *Ind. Manag. Data Syst.*, Vol. 118, No. 3, 2017.

[62] H. H. H. Aldboush and M. Ferdous, "Building Trust in Fintech: An Analysis of Ethical and Privacy Considerations in the Intersection of Big Data, AI, and Customer Trust", *Int. J. Financ. Stud.*, Vol. 11, No. 3, 2023, doi: 10.3390/ijfs11030090.