**Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control**
**Journal homepage: http://kinetik.umm.ac.id**
**ISSN: 2503-2267**
**Vol. 9, No. 4, November, Pp. 397-408**

# Aspect-level sentiment analysis on GoPay app reviews using multilayer perceptron and word embeddings

**Henzi Juandri*[1], Hasmawati[1], Bunyamin[1]**
School of Computing, Informatics, Telkom University, Bandung, Indonesia[1]

## Article Info

**Keywords:**
GoPay, Sentiment Analysis, Multilayer Perceptron, Word Embeddings, Natural Language Processing

**Article history:**
Received: June 08, 2024
Accepted: August 01, 2024
Published: November 01, 2024

**Cite:**
H. Juandri, Hasmawati, and Bunyamin, "Aspect-level Sentiment Analysis on GoPay App Reviews Using Multilayer Perceptron and Word Embeddings", *KINETIK*, vol. 9, no. 4, Nov. 2024.
https://doi.org/10.22219/kinetik.v9i4.2041

*Corresponding author.
Henzi Juandri
E-mail address:
henzijuandri.work@gmail.com

## Abstract

The increasing use of smartphone in Indonesia has encouraged the development of digital wallet applications, one of which is GoPay. Nowadays, GoPay has gained significant popularity among the public in Indonesia. Therefore, this research conducts aspect-level sentiment analysis to analyze user reviews of the GoPay application in more detail and depth. The sentiment analysis process in this study utilizes the Multilayer Perceptron (MLP) with fastText and word2vec as word embeddings. The dataset used is GoPay application reviews, which consist of 15,000 reviews collected from Google Play Store. The dataset is categorized into three main aspects: Feature and functionality, App Interface, and User Satisfaction. The stages of the research include data preparation, data preprocessing, word embeddings, model training, and model testing and evaluation. This research explores the effect of fastText and word2vec as word embeddings on model performance. Furthermore, this research examines the application of oversampling techniques, such as SMOTE and Random Oversampling. Based on the experiments conducted, utilizing fastText as word embeddings in MLP with a balanced dataset resulted the best model performance, with an F1-Score of 97%, Recall of 96%, and Precision of 95% for category classification. Then, for sentiment classification, using fastText on MLP with a balanced dataset resulted in a value of 98% for each of the F1-score, Recall, and Precision metrics. This research validates that MLP is effective for aspect-level sentiment analysis, delivering strong evaluation results.

## 1. Introduction

Technological developments have significantly changed smartphone use in Indonesia. In 2023, the number of smartphone users in Indonesia reached over 190 million [1]. This is in line with the increasing number of applications available for smartphones. One of the most downloaded applications on the Google Play Store is GoPay. GoPay is a digital wallet launched by GoTo Indonesia [2]. Initially, GoPay was a digital payment service in the Gojek application, but in July 2023, GoTo released the GoPay application as an independent application. The popularity of GoPay results in many reviews that can be found on Google Play Store [3]. These reviews can be negative or positive and general or specific to certain aspects on GoPay application. Through the sentiment analysis process, the review data can be utilized as an essential source of information.

Sentiment analysis is a subfield of Natural Language Processing (NLP) that focuses on analyzing emotions and sentiments from text [4]. Sentiment analysis aims to classify a text into certain categories, such as positive, negative, or neutral. There are several levels in sentiment analysis, namely document, sentence, and aspect levels [5]. In contrast to sentiment analysis at sentence and document levels, sentiment analysis at aspect level classifies sentiment based on certain components or aspects relevant to the text [6]. Through aspect-level sentiment analysis, the classification of GoPay application reviews can be done in more detail because the classification is carried out based on several aspects that are relevant to GoPay application, such as the application performance, security, and appearance.

One of the classification methods that can be used in the sentiment analysis process is Multilayer Perceptron (MLP). In the previous studies, MLP was used for various sentiment analysis cases [5], [7], [8], [9]. MLP was compared with several other algorithms, such as Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Support Vector Machine (SVM), and Naive Bayes (NB) in sentiment analysis cases. MLP demonstrated prominent performance compared to other machine learning algorithms and can outperform other deep learning algorithms in certain situations [5], [8]. In addition, MLP is significantly less computationally intensive compared to more advance neural network models like RNN and CNN. It has the shortest execution time for model training in most datasets, as proven in research [5].

Other research shows that the use of word2vec and fastText as word embedding can improve the performance of various classification models, including MLP. The use of word2vec in MLP and CNN can produce accuracy of 95% and 92%, respectively [7]. Furthermore, in the sentiment analysis of the new normal in Indonesia, the utilization of fastText on MLP, NB, and SVM resulted in an average F1 score of 91% for MLP, 92% for SVM, and 72% for NB [9]. In addition, the effectiveness of word2vec and fastText as word embeddings was also proven in [10]. This study compared the use of fastText, word2vec, and GloVe on several classification models in sentiment analysis. The best results were obtained by fastText and word2vec, which outperformed GloVe on almost every classification model. While GloVe captures global co-occurrence statistics useful for understanding overall semantic relationships, fastText and word2vec are better suited for sentiment analysis especially at the aspect level, as they capture local context and subword information more effectively.

Further research compares classic word embeddings such as fastText, word2vec, and Glove with contextual word embeddings such as Bidirectional Encoder Representations from Transformers (BERT) and Embeddings from Language Models (ELMo) where the comparison is done on various datasets and classification models. Although the accuracy of contextual embeddings is better than classic embeddings, the difference in accuracy is not very significant. On most datasets, the difference varies around 1-5%, with contextual embeddings showing an improvement over classic embeddings [11], [12]. However, it is important to note that while the accuracy difference is not high, contextual embeddings require significantly more computing time and memory resources [13].

Previous research has proven the effectiveness of MLP in various sentiment analysis cases but it is still at the sentence-level or document-level sentiment analysis [5], [7], [8], [9]. Hence, exploration and research on MLP at aspect-level sentiment analysis, also known as Aspect-Based Sentiment Analysis (ABSA), still needs improvement. Previously, some studies have conducted ABSA on five types of Twitter datasets. Various classification methods have been applied, including MLP, NB, Random Forest (RF), and Support Vector Classifier (SVC). MLP showed significant potential by achieving higher accuracies than other classification methods for every dataset, with average accuracies of 78.99%, 84.09%, 80.38%, 82.37%, and 84.72%. However, the use of word embeddings as feature extraction has not been applied. In addition, further research is still needed to validate the effectiveness of MLP in other ABSA cases [14].

Another significant research contribution is a study focusing on aspect-level sentiment analysis in the context of smartphone application reviews [15], [16]. One of the first to address the aspect-level sentiment analysis, specifically on smartphone application reviews. This research builds two baseline models using MLP and SVM for aspect category classification and aspect sentiment classification, achieving F1 scores of 32%, 31%, and 29% and accuracies of 66%, 67%, and 64% across different aspects. This research also introduces the AWARE dataset, featuring 11,323 reviews across three aspects: Productivity, Social Networking, and Games, and encourages further exploration in this field [15]. Continuing this exploration, subsequent research implemented CNN and several word embeddings on the same dataset, resulting in significant performance improvements, with accuracies of 87.88%, 93.75%, and 31.25% at aspect category classification and improvements of 16.43%, 23.35%, and 3.72% at aspect sentiment classification, demonstrating the effectiveness of using word embeddings to enhance model performance in ABSA task [16].

Based on previous research, MLP provides a balance of strong performance and computational efficiency, outperforming several classic machine learning models while being less computationally intensive than more advanced deep learning models [5], [8]. Additionally, fastText and word2vec have proven to improve the performance of classification models, including MLP. Moreover, fastText and word2vec still offer great performance when compared to contextual embeddings methods like BERT or ELMo, while being less computationally intensive. This makes fastText and word2vec compatible for ABSA, which is a relatively straightforward classification task [7], [9], [10], [11], [12], [13].

This research aims to address this gap by exploring the relatively unexplored area of aspect-level sentiment analysis for smartphone application reviews using the GoPay review dataset. By implementing a Multilayer Perceptron (MLP) for both aspect category and sentiment classification with fastText and word2vec, which are selected for their performance and computational efficiency. The aspects used are Feature and Functionality, App Interface, and User Satisfaction. This research contributes to sentiment analysis at aspect level of the GoPay review dataset, which is still relatively new, considering that GoPay was launched independently in July 2023. In addition, this research also aims to expand the exploration of previous research, particularly in enhancing MLP performance at aspect-based sentiment analysis [15].

## 2. Research Method

This research consists of five main stages: data preparation, data preprocessing, word embeddings, model training, and model testing and evaluation. The research flow is detailed in Figure 1. In the data preparation stage, GoPay application review data is collected through a scrapping process on Google Play Store using the Google Play Scrapper tool. Furthermore, labeling is carried out on the dataset, the labeling process is carried out based on the aspects and sentiments of each review in the dataset. This stage produces two types of datasets, namely datasets for aspect category classification and datasets for sentiment classification. Next, data cleaning, case folding, stopword removal, stemming, and tokenizing are conducted at the data preprocessing stage. After that, in the word embeddings stage, the clean data is represented in vector form using fastText and word2vec word embeddings methods. The dataset represented in vector form is then used in the model training stage. The aspect category dataset is used for aspect

© 2024 The Authors. Published by Universitas Muhammadiyah Malang
This is an open access article under the CC BY NC SA license (https://creativecommons.org/licenses/by-nc-sa/4.0/)

category model training, and the sentiment dataset is used for sentiment model training. In the final stage, the model performance are tested and evaluated using metrics from the confusion matrix.

### 2.1 Data Preparation

The dataset used in this research is the GoPay application review data from the Google Play Store. The dataset is collected using google-play-scraper which is a library of the Python programming language. The dataset includes reviews up to October 30, 2023 with a total of 15,000 reviews. The examples of raw data from the Gopay review dataset scrapping process can be seen in Table 1. The reviews are divided and labeled into two datasets: the aspect classification dataset and sentiment classification dataset. The labeling process for each dataset is conducted by two people. The aspects identified include Feature & Functionality, App Interface, and User Satisfaction. Attributes or keywords that belong to each aspect can be seen in Table 2.

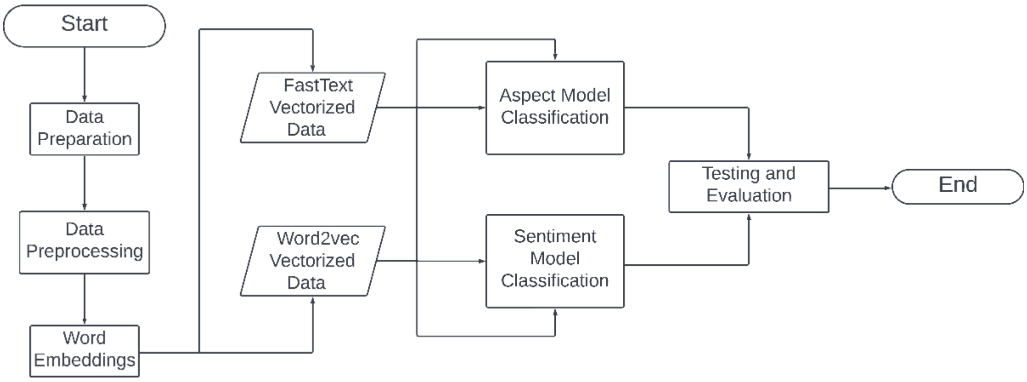

*Figure 1. Research Workflow*

*Table 1. Example of Raw Gopay Review Data*

| Review | Rating | At |
|---|---|---|
| Aplikasi gopay ini bener- bener sangat membantu aku banget. Mau transfer mau bayar listrik atau bayar apapun ga harus ribet pergi² keluar. Lewat menu gopay ini cepat banget dan mudah. Fleksibel banget. Fiturnya simpel juga mudah dipahami. | 5 | 27/10/2023 15:23:15 |
| Gopay tabungan mempersulit TOP UP. Saya pikir sama cepatnya seperti topup biasa, teenyata lebih susah, saldo topup sudag masuk tapi di ambil langsung ke saldo tabungan tapi tabungan sendiri gangguan alhasil percuma top up 200 rb kagak bisa di pakai langsung. | 1 | 19/10/2023 4:49:39 |
| Blm bisa daftar? Karena pakai no luar Negara apakah bisa di daftar melalui email? Biar pengguna lain yg dari Indonesia, bisa pakai apk ini di luar negri. Thank! | 3 | 10/27/2023 5:20:56 |

*Table 2. Keyword of each Aspect*

| Aspect | Keyword |
|---|---|
| Feature & Functionality | login, register, transfer, upload, sign up, top up, refund, bayar, beli, transaksi, pinjam, etc. |
| App Interface | tampilan, halaman, UI, navigasi, visual, desain, gambar, ikon, menu, warna, tombol, mudah digunakan, etc. |
| User Satisfaction | bagus, senang, kesal, ok, puas, kecewa, keren, mantap, nyaman, lancar, berguna, buruk, ribet, gampang, sulit, etc. |

The labeling process for the aspect category dataset is done manually by two people. Review data are labeled '1' or '0'. If the review data contain keywords or attributes of a particular aspect, it is labeled '1'; if not, it is labeled '0'. Review data can be classified into one or more aspects (multi-label). Table 3 provides an example of aspect category dataset labeling.

*Table 3. Aspect Category Labeling*

| Review | Aspect | | |
|---|---|---|---|
| | Feature & Functionality | App Interface | User Satisfacion |
| Keseluruhan bagus, tapi tolong perbaiki UI atau tampilannya. Tampilan menunya malah lebih bagus aplikasi Go-Jek, dibandingkan aplikasi sebelah juga lebih bagus UI nya. Oleh karena itu saya kasih bintang 3 dulu sampai tampilan UI dipercantik lagi. | 0 | 1 | 1 |

Based on Table 3, the review data contains several keywords related to the App Interface aspect, namely the words 'UI', 'menu', and 'tampilan'. Therefore, the App Interface aspect is labeled '1'. In addition, the review data also contains keywords related to the User Satisfaction aspect, namely 'keseluruhan bagus'. Therefore, the User Satisfaction aspect is also labeled '1'. However, the review data does not contain keywords related to Feature & Functionality at all, therefore the Feature & Functionality aspect is labeled '0'.

For the sentiment classification dataset, the labeling process is also done manually by 2 people with the help of the rating attribute from the dataset. Ratings scales are clustered to negative (1-2), neutral (3), and positive (4-5). Labeling is done by first looking at the rating value and then cross-checking it with the text in the review data. The review data are labeled as '1', '0', or '-1'. Review data with a positive overall impression are labeled '1', review data with a negative overall impression are labeled '-1', and neutral reviews are labeled '0'. In other words, the review data are categorized into three possible labels (multi-class). Table 4 illustrates the labeling example for the sentiment classification dataset.

*Table 4. Aspect Sentiment Labeling*

| Review | Aspect | | | |
|---|---|---|---|---|
| | Rating | Feature & Functionality | App Interface | User Satisfacion |
| tentang UI sih, tampilan UI nya jadul bgt kek ebanking tahun 2010/2013. gk selera, lebih bagus yang app gojek. buat resolusi juga saya lihat tadi warna wallpaper dll pecah pixel dan kualitas bit warna seperti dibawah standar. | 2 | 0 | -1 | -1 |

Based on Table 4, the rating value for the review data is 2, which means that the review data is likely to be negative. In addition, there are sentences containing negative sentiment towards the App Interface aspect, one of which is 'warna wallpaper dll pecah'. Therefore, the App Interface aspect is labeled '-1'. After that, there are also sentences containing negative sentiments towards the User Satisfaction aspect, namely 'gk selera'. Therefore, the User Satisfaction aspect is labeled'-1'. Meanwhile, the sentiment for the Feature & Functionality aspect is neutral because there are no keywords related to this aspect.

### 2.2 Data Preprocessing

The review data resulted from the labeling process is still in the form of non-standard text and has a lot of noise, such as emoticons, punctuation marks, symbols, spelling errors, etc. Therefore, the review data is processed through several stages: data cleaning, case folding, stopword removal, stemming, and tokenization. Table 5 shows the examples of the preprocessing stages.

*Table 5. GoPay Review Dataset Preprocessing*

| Preprocessing Stage | GoPay Review |
|---|---|
| Original Text | Sampai sejauh ini g ada kendala. Seneng aplikasi nya. Halaman fiturnya simple tapi tetep modern n menarik. Riwayat lengkap teratur pake nya gampang. Informasi yg dibutuhkan juga mudah nyarinya. catatan keuangan yg keren menurutku 😁. |

© 2024 The Authors. Published by Universitas Muhammadiyah Malang
This is an open access article under the CC BY NC SA license (https://creativecommons.org/licenses/by-nc-sa/4.0/)

| | |
|---|---|
| Data Cleaning | Sampai sejauh ini g ada kendala Seneng aplikasi nya Halaman fiturnya simple tapi tetep modern n menarik Riwayat lengkap teratur pake nya gampang Informasi yg dibutuhkan juga mudah nyarinya Catatan keuangan yg keren menurutku |
| Case Folding | sampai sejauh ini g ada kendala seneng aplikasi nya halaman fiturnya simple tapi tetep modern n menarik riwayat lengkap teratur pake nya gampang informasi yg dibutuhkan juga mudah nyarinya catatan keuangan yg keren menurutku |
| Stopword Removal | sampai jauh kendala seneng aplikasi halaman fitur simple modern menarik Riwayat lengkap teratur pake gampang informasi dibutuhkan mudah nyari catatan keuangan keren menurutku |
| Stemming | sampai jauh kendala seneng aplikasi halaman fitur simple modern tarik riwayat lengkap atur pake gampang informasi butuh mudah cari catat uang keren turut |
| Tokenization | ["sampai", "jauh", "kendala", "seneng", "aplikasi", "halaman", "fitur", "simple", "modern", "tarik", "riwayat", "lengkap", "atur", "pake", "gampang", "informasi", "butuh", "mudah", "cari", "catat", "uang", "keren", "turut"] |

### 2.3 Word Embeddings

At the word embedding stage, the preprocessed dataset is represented in vector form before being processed by the model. The word embeddings used in this research are word2vec and fastText. The vector representation results from fastText and word2vec are used as input data for the model training process. The performance of the model is compared based on the vector representations generated by both word2vec and fastText embeddings.

#### 2.3.1 fastText Word Embeddings

FastText is a word embedding developed by Bojanowski et al. at Facebook AI Research (FAIR), fastText was developed based on the continuous skip-gram architecture from word2vec. Like word2vec, fastText uses two main architectures, Continuous Bag of Words (CBOW) and Skip-Gram, which are based on shallow neural networks. The uniqueness of fastText lies in the way it represents text in vector form through a subword approach. For example, the word “where” with three n-grams will be represented as <wh, we, her, here, re>. Each character of the n-gram text will be represented in vector form [17].

In this research, the fastText model used is a pre-trained model developed by Facebook. The model, built with a CBOW architecture and an n-gram length of 5, it was initially trained using Wikipedia and Common Crawl corpus with various languages. However, to ensure its relevance, the model is retrained using the GoPay app review corpus. The representation result of the model is a word vector with a dimension size of 256. To obtain sentence vectors, each word vector is calculated using the L2-Normalized Average Sum of Word Vectors defined in Equation 1 and Equation 2 [18].

$$sentence\_vector = \frac{1}{n}\sum_{i=1}^{n}\frac{\vec{w}_i}{\|\vec{w}_i\|_2} \tag{1}$$

$$\|\vec{w}_i\|_2 = \sqrt{\sum_{i=1}^{n}\vec{w}_i^{\,2}} \tag{2}$$

Based on Equation 1, $n$ is the number of words, $\vec{w}_i$ is the vector of the $i$-th word, and $\|\vec{w}_i\|_2$ is the L2-normalization value of the $i$-th word. The L2-normalization is defined in Equation 2.

#### 2.3.2 Word2vec Word Embeddings

Word2vec is a word embedding developed by Mikolov et al. at Google. word2vec has two architectures: Continuous Bag of Words (CBOW) and Skip-Gram. Both architectures are based on shallow neural networks that consist of an input layer, one hidden layer, and an output layer. The uniqueness of word2vec lies in its ability to capture the semantic value between words. A word will still have semantic value even though it has been represented in vector form. For example, the words 'France' and 'Paris' will have similar vector values because the two words have a semantic relationship. This can be an advantage of word2vec compared to frequency-based vector representation methods such as Bag of Words (BoW) or TF-IDF [19].

In this study, the word2vec model was built and trained using Wiki corpus with a CBOW architecture. Similar to the fastText model, the word2vec model was retrained using GoPay app review corpus. The word2vec model generates word vectors with a dimension size of 256. To obtain the sentence vector, each word vector is calculated using the L1-Normalized Sum of Word Vectors, as defined in Equation 3 [20].

$$sentence\_vector = \frac{\sum_{i=1}^{n} \vec{w}_i}{\sum_{i=1}^{n} \|\vec{w}_i\|} \tag{3}$$

Based on the Equation 3, $\vec{\boldsymbol{w}}_{\boldsymbol{i}}$ is the vector of the $\boldsymbol{i}$-th word, and $\|\vec{\boldsymbol{w}}_{\boldsymbol{i}}\|$ is the L1-normalization value the $\boldsymbol{i}$-th word. L1-normalization is equivalent to absolute value.

### 2.4 Model Training

In the model training stage, the dataset result from the word embeddings is used to train two models: the aspect category classification model and the sentiment classification model. The general model training flow is shown in Figure 2.

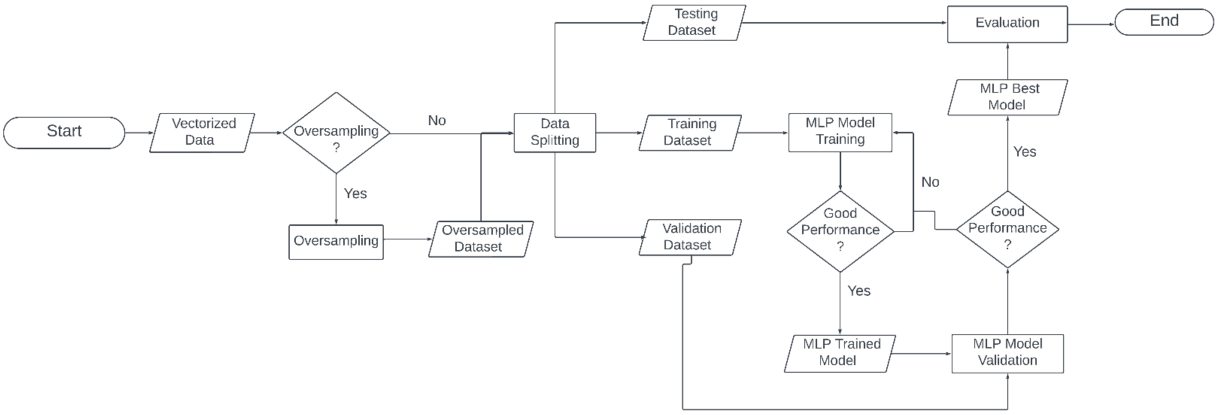

*Figure 2. Model Training*

In addition, this stage explores oversampled data and original data. Oversampling is necessary because the dataset used has an imbalanced data distribution. Oversampling is a technique to overcome data imbalance by adding replica samples (data) from minority classes in the dataset [21]. Based on Figure 2, if oversampling is performed, the process is done before the dataset is split into three data types (training, validation, and testing). Meanwhile, the embedded review data is split directly into three data types if oversampling is not performed.

In this research, both types of classification models are built using Multilayer Perceptron (MLP). Each model is trained with dataset from embeddings. The training process continues until the best-performing model is achieved. Model performance is evaluated using the evaluation matrix described in section 2.5. Model with the best performance is saved and re-evaluated using the testing dataset.

### 2.4.1 Aspect Category Model Training

The category aspect classification model identifies aspect categories in GoPay review data. A review data can be classified into more than one aspect. Therefore, the MLP model built is a multi-label classification model. The training process on the aspect category model uses two datasets from fastText and word2vec embeddings. The training flow for the aspect category model is shown in Figure 3. Based on Figure 3, Model 1 is trained using the fastText embeddings dataset, and Model 2 is trained using the word2vec embeddings dataset.

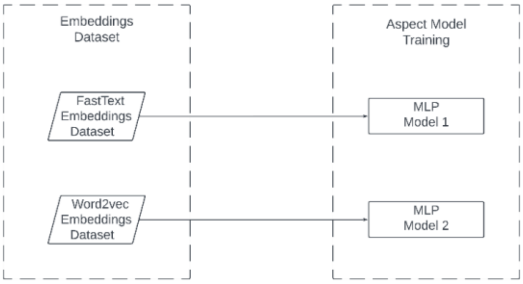

*Figure 3. Aspect Category Model Training*

© 2024 The Authors. Published by Universitas Muhammadiyah Malang
This is an open access article under the CC BY NC SA license (https://creativecommons.org/licenses/by-nc-sa/4.0/)

### 2.4.2 Sentiment Model Training

The sentiment classification model classifies the sentiment of GoPay review data. Review data can be classified into three possible sentiments, namely negative, positive, or neutral (multi-class classification). In this research, the number of the MLP models is adjusted to the number of the aspects. The training process on the sentiment model uses two datasets from the embeddings. The training flow of the sentiment models is shown in Figure 4. Based on Figure 4, Model 1 is the sentiment model for Feature & Functionality aspect, Model 2 is the sentiment model for the App Interface aspect, and Model 3 is the sentiment model for User Satisfaction Aspect. Meanwhile, Model 4, Model 5, and Model 6 are sentiment models for the same aspects.

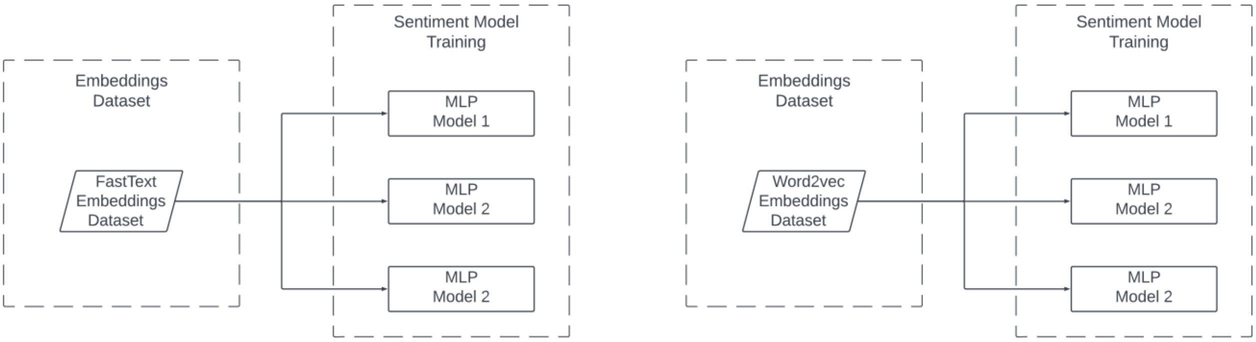

*Figure 4. Sentiment Model Training*

### 2.4.3 Multilayer Perceptron (MLP) Classifier

Multilayer Perceptron (MLP) is one type of Artificial Neural Network (ANN) algorithm that consists of several perceptron layers: input, hidden, and output. Each perceptron has input, weight, and bias values optimized through backpropagation [7]. Furthermore, the backpropagation and feedforward processes are carried out from the input layer to the output layer; this process will be carried out repeatedly until the best weight and bias values and the lowest error value are obtained [22]. Optimization of input, weight, and bias values can be accelerated by applying activation functions to perceptrons such as ReLU, Sigmoid, or Tanh, which introduce non-linear elements that allow the model to learn more complex patterns [23]. The calculation on each perceptron is defined by Equation 4.

$$y = f\left(b + \sum_{i=1}^{n} x_i w_i\right) \tag{4}$$

In Equation 4, $y$ is the output value, $b$ is the bias value, $x$ is the input value, $w$ is the weight value, and $f$ is the activation function [7]. In this research, both classification models are built using MLP. The aspect classification model is trained using a binary cross entropy loss function to address the multi-label classification problems. Binary cross-entropy is suitable for multi-label problems as it measures the loss value between the binary prediction and the actual label separately for each class [24]. Furthermore, to handle the multi-class problem, the sentiment classification model is trained using the sparse categorical cross-entropy loss function. As in the previous research, sparse categorical cross entropy is effectively used in multi-class problems [25].

### 2.5 Testing and Evaluation

The model's performance is measured using a confusion matrix in the testing and evaluation stage. In measuring the performance, the Confusion Matrix utilizes actual and predicted data classification results. The confusion matrix can be seen in Table 6.

*Table 6. Confusion Matrix Table*

| Actual Value | Predicted Value | |
|---|---|---|
| | Positive | Negative |
| Positive | TP<br>(True Positive) | FN<br>(False Negative) |
| Negative | FP<br>(False Positive) | TN<br>(True Negative) |

The confusion matrix table show the performance of the models in terms of the accuracy, precision, recall, and F1-Score [26], [27]. Equation 5, Equation 6, Equation 7, and Equation 8 define the calculation of accuracy, precision, recall, and F1-Score, respectively.

$$\boldsymbol{accuracy = \frac{TP + TN}{TP + FP + TN + FN}} \tag{5}$$

$$\boldsymbol{precision = \frac{TP}{TP + FP}} \tag{6}$$

$$\boldsymbol{recall = \frac{TP}{TP + FN}} \tag{7}$$

$$\boldsymbol{F1\ Score = \frac{2\ \times (recall\ \times precision)}{(recall + precision)}} \tag{8}$$

## 3. Results and Discussion

This section discusses the research findings, including the results of dataset processing and model evaluation. Model evaluation includes the aspect model and sentiment model. Model evaluation is conducted on balanced and imbalanced datasets, as well as on fastText and word2vec word embeddings. The evaluation metrics used for evaluation are precision, F1-Score, and recall.

### 3.1 Dataset Result

After preprocessing and embeddings, the final dataset has 14,920 reviews. The distribution of data in the review dataset for each aspect classification and sentiment classification can be seen in Table 7 and Table 8.

*Table 7. Aspect Classification Dataset Distribution*

| Aspect | Presence | Amount of Data | Percentage | Data Total |
|---|---|---|---|---|
| Feature & Functionality | Yes | 10,061 | 67.43% | 14,920 |
| | No | 4,859 | 32.57% | |
| App Interface | Yes | 1,522 | 10.2% | 14,920 |
| | No | 13,398 | 89.8% | |
| User Satisfaction | Yes | 13,160 | 88.2% | 14,920 |
| | No | 1,760 | 11.8% | |

Based on Table 7, the aspect classification dataset has a relatively imbalanced distribution. In the Feature and functionality aspect, 10,061 review data contain Feature and functionality aspects, and 4,859 review data do not contain these aspects. The App Interface aspect has a strongly imbalanced distribution, with only 1,522 review data containing the App Interface aspect and 13,398 review data that does not contain this aspect. In the aspect of user satisfaction, the review data is also relatively imbalanced.

*Table 8. Sentiment Classification Dataset Distribution*

| Aspect | Sentiment | Amount of Data | Percentage | Data Total |
|---|---|---|---|---|
| Feature & Functionality | Positive | 7,609 | 51% | 14,920 |
| | Negative | 2,452 | 16.43% | |
| | Neutral | 4,859 | 32.57% | |
| App Interface | Positive | 1,374 | 9% | 14,920 |
| | Negative | 148 | 1% | |
| | Neutral | 13,398 | 90% | |
| User Satisfaction | Postive | 10,195 | 68.33% | 14,920 |
| | Negative | 2,965 | 19.87% | |
| | Neutral | 1,760 | 11.80% | |

© 2024 The Authors. Published by Universitas Muhammadiyah Malang
This is an open access article under the CC BY NC SA license (https://creativecommons.org/licenses/by-nc-sa/4.0/)

The data distribution on the sentiment classification dataset can be seen in Table 8. The Feature & Functionality aspect is dominated by positive review data totaling 7,609 reviews, while negative reviews have the least 2,452 reviews. In the app interface aspect, neutral reviews are the most common, with 13,398 reviews, inversely proportional to negative reviews, which only amount to 148 reviews. Regarding user satisfaction, positive reviews have the most data, namely 10,195 reviews, and neutral data has the least, at 1,760 reviews. An oversampling technique is performed to overcome the imbalance of data in the aspect classification and sentiment classification datasets.

### 3.2 Aspect Category Classification

This section discusses the results of MLP model testing for aspect category classification. Testing is conducted on both balanced dataset and imbalanced dataset. The Random oversampling (ROS) technique is used to address data imbalance in this multilabel classification task. While Oversampling techniques like the Synthetic Minority Over-sampling Technique (SMOTE) is effective for binary or multiclass classification, it is less suitable for the characteristics of multilabel data. SMOTE works by creating synthetic samples of minority data by interpolating between existing examples [28], [21], [29]. Tests are also performed using both fastText and word2vec embedding datasets. This experiment aims to evaluate how dataset imbalance can affect the performance of MLP models on aspect classification tasks. Additionally, we also evaluate how well the MLP model performs with both types of word embeddings. The results of this evaluation are presented in Table 9.

*Table 9. Aspect Category Classification Result*

| Aspect Category | Word Embeddings | Imbalanced Dataset | | | Balanced Dataset | | |
|---|---|---|---|---|---|---|---|
| | | Precision | F1-Score | Recall | Precision | F1-Score | Recall |
| Feature & Functionality | fastText | **87%** | **93%** | **90%** | **89% (+2%)** | **90%** | **90%** |
| | Word2vec | 86% | 92% | 89% | 88% **(+2%)** | 89% | 88% |
| App Interface | fastText | 84% | **50%** | **63%** | **95% (+11%)** | **97% (+47%)** | **96% (+33%)** |
| | Word2vec | **86%** | 25% | 39% | 91% (+5%) | 91% (+66%) | 91% (+52%) |
| User Satisfaction | fastText | 91% | 98% | 95% | **92% (+1%)** | 89% | 90% |
| | Word2vec | **92%** | 98% | 95% | 91% | 89% | 90% |

#### 3.2.1 Word Embeddings Comparison Result

Based on the imbalanced dataset results in Table 9, it can be seen that the use of word embeddings in the MLP model shows different performance in each aspect category. In the Feature and functionality aspect, MLP+fastText produces slightly superior performance than MLP+word2vec. MLP+fastText sequentially obtains precision, F1-score, and recall of 87%, 93%, and 90%, which indicates that the performance of this model is 1% higher than MLP+word2vec.

In the App Interface aspect, both models performed poorly. This is due to the extremely imbalanced data distribution in the App Interface aspect. However, the performance of MLP+fastText still outperforms MLP+word2vec, where MLP+fastText obtains F1-Score and Recall of 50% and 63% respectively, while MLP + word2vec obtains F1-Score and Recall of 25% and 39% respectively. In terms of user satisfaction, both models show good and consistent performance, and there is no significant difference between them. The only difference is the precision, where MLP+word2vec obtains a precision of 92%, while MLP+fastText obtains a precision of 91%.

Based on the evaluation results on the three aspects, the use of fastText and word2vec word embeddings in MLP for classifying aspect categories shows a slight difference in performance. The performance of MLP+fastText is slightly better than MLP+word2vec in each aspect category, especially in the App Interface aspect. The superiority of fastText is due to the subword approach in fastText, which breaks the text into n-grams. Through the subword approach, fastText is effective when used on complex text [9]. In addition, the subword approach used by fastText enhances its ability to handle out-of-vocabulary (OOV) words, which is particularly useful for GoPay app review dataset that includes many nonstandard or slang sentences [30], [31].

.

#### 3.2.2 Oversampling Result

The results of applying oversampling to MLP can be seen in the balanced dataset results in Table 9. For Feature and Functionality and User Satisfaction aspects, where the distribution of the two datasets is quite balanced, the application of oversampling shows little change, both in the MLP+fastText model and the MLP+word2vec model. The increase is only in the Precision metrics, which range from 1-2%. In the app interface aspect, which has an imbalanced dataset distribution, there is a significant increase in both MLP models. In the MLP+fastText model, there was an

*Cite: H. Juandri, Hasmawati, and Bunyamin, "Aspect-level Sentiment Analysis on GoPay App Reviews Using Multilayer Perceptron and Word Embeddings", KINETIK, vol. 9, no. 4, Nov. 2024.* https://doi.org/10.22219/kinetik.v9i4.2041

increase in precision, F1-score, and recall, i.e., 95% (+11%), 97% (+47%), and 96% (+33%) respectively. Then, for the MLP+word2vec model, there was a sequential increase in the precision, F1-score, and recall, i.e., 91% (+5%), 91% (+66%), and 91% (+52%) respectively.

Based on the evaluation results on the three aspects, it can be concluded that the application of Random Oversampling (ROS) has a fairly diverse impact on MLP for multi-label classification. In datasets with a fairly balanced distribution, the application of ROS does not greatly improve the model performance, such as in the Features and functionality and user satisfaction aspects. However, for datasets with an imbalanced distribution, such as the App Interface aspect, the application of ROS resulted in a significant improvement in both models. This may be due to the way ROS randomly duplicates minority samples [32].

### 3.3 Sentiment Aspect Classfication

This section discusses the results of MLP model testing for sentiment classification. In sentiment classification, the model is built based on the number of aspects, hence three models are created. Each model is tested in the same way as aspect category classification, which is tested on balanced and imbalanced datasets, as well as on datasets generated from fastText and word2vec embeddings. The oversampling technique used is SMOTE for testing on balanced data because sentiment classification is a multi-class classification task, unlike aspect classification, which is a Multilabel classification task [21], [29]. This experiment aims to evaluate the performance of each sentiment model across multiple embeddings and dataset distribution. The evaluation results of each sentiment model can be seen in Table 10.

*Table 10. Sentiment Classification Result*

| Sentiment Model | Scenario | F1 Score | Recall | Precision |
|---|---|---|---|---|
| Feature & Functionality | Imbalance Data + fastText | 82% | 82% | 82% |
| | Imbalance Data + Word2vec | 77% | 77% | 78% |
| | Balance Data + fastText | **87%** | **87%** | **87%** |
| | Balance Data + Word2vec | 84% | 74% | 74% |
| App Interface | Imbalance Data + fastText | 55% | 56% | 89% |
| | Imbalance Data + Word2vec | 54% | 53% | 87% |
| | Balance Data + fastText | **98%** | **98%** | **98%** |
| | Balance Data + Word2vec | 96% | 96% | 96% |
| User Satisfaction | Imbalance Data + fastText | 87% | 74% | 77% |
| | Imbalance Data + Word2vec | 66% | 65% | 69% |
| | Balance Data + fastText | **89%** | **89%** | **89%** |
| | Balance Data + Word2vec | 84% | 84% | 84% |

Based on the evaluation results in Table 10, the scenario that produces the best performance is when using fastText as word embeddings and oversampling the dataset (balanced data + fastText). For the fastText and balanced data scenarios, the Feature and Functionality sentiment model scored 87% for f1-score, recall, and precision. The App Interface sentiment model scored 98% for f1-score, recall, and precision. Meanwhile, the User Satisfaction sentiment model scored 89% for f1-score, recall, and precision. This shows that the oversampling technique highly affects the performance of the resulting model, especially on datasets with an imbalanced data distribution. In addition, this experiment proves that using fastText as word embeddings for MLP is more effective than using word2vec. As explained in section 3.2.1, fastText is more effective when used on complex text [9], [30], [31]. This causes the evaluation results with fastText to perform better than word2vec for each sentiment model in this experiment.

## 4. Conclusion

This research implemented an aspect-level sentiment analysis model on GoPay application review dataset using Multilayer Perceptron (MLP) with fastText and word2vec word embeddings. Sentiment classification is carried out in three aspects: feature functionality, app interface, and user satisfaction. Experimental results show that word embeddings play an essential role in enhancing the performance of MLP models for aspect-level sentiment classification. Word embeddings such as fastText and word2vec have been proven to capture semantic relationships between words that positively impact the model's ability to understand the context in user reviews of GoPay application. In the experiments that have been carried out, utilizing fastText as word embeddings provides better performance than word2vec, with the best performance observed in the sentiment classification on balanced data with fastText, resulting in 87% F1-score, recall, and precision for Feature and Functionality, 98% for App Interface, and 89% for User Satisfaction. In addition, data distribution is also an essential factor in obtaining better performance results. This is

© 2024 The Authors. Published by Universitas Muhammadiyah Malang
This is an open access article under the CC BY NC SA license (https://creativecommons.org/licenses/by-nc-sa/4.0/)

because balanced data can ensure that the trained model is not biased toward certain classes and can provide more accurate results for each class.

The aspect-level sentiment analysis that has been carried out on GoPay application allows developers to identify areas that require further improvement and development, such as improving certain features, improving the app interface, or increasing overall user satisfaction. This research contributes to the development of MLP methods with fastText and word2ec embeddings for aspect-level sentiment analysis on Gopay app review dataset, which show better performance than the previous studies [15]. In future research, further exploration can be conducted into advanced models and hybrid techniques, such as integrating transformers or contextual embeddings like BERT with MLP, to assess if they offer significant improvements over the current MLP with fastText and word2vec. In addition, future research may utilize oversampling techniques that are more suitable than random oversampling for the aspect category classification (multi-label classification).

## References

[1] Statista, “1 Number of smartphone users in Indonesia from 2018 to 2028 (in millions) [Graph].

[2] Rakuten Insight, “2 Major e-payment services used among respondents in Indonesia as of October 2022 [Graph].”

[3] GoPay, “3 GoPay: Transfer & Payment," Google Play Store.”

[4] K. L. Tan, C. P. Lee, and K. M. Lim, “4 A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research,” Apr. 01, 2023, *MDPI*. https://doi.org/10.3390/app13074550

[5] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, “5 Sentiment analysis based on deep learning: A comparative study,” *Electronics (Switzerland)*, vol. 9, no. 3, Mar. 2020. https://doi.org/10.3390/electronics9030483

[6] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, “6 A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges,” Mar. 2022. https://doi.org/10.48550/arXiv.2203.01054

[7] S. Behl, A. Rao, S. Aggarwal, S. Chadha, and H. S. Pannu, “7 Twitter for disaster relief through sentiment analysis for COVID-19 and natural hazard crises,” *International Journal of Disaster Risk Reduction*, vol. 55, Mar. 2021. https://doi.org/10.1016/j.ijdrr.2021.102101

[8] A. E. O. Carosia, G. P. Coelho, and A. E. A. Silva, “8 Analyzing the Brazilian Financial Market through Portuguese Sentiment Analysis in Social Media,” *Applied Artificial Intelligence*, vol. 34, no. 1, pp. 1–19, Jan. 2020. https://doi.org/10.1080/08839514.2019.1673037

[9] R. P. Aluna, I. N. Yulita, and R. Sudrajat, “9 Electronic News Sentiment Analysis Application to New Normal Policy During The Covid-19 Pandemic Using Fasttext And Machine Learning,” in *2021 International Conference on Artificial Intelligence and Big Data Analytics*, 2021, pp. 236–241. https://doi.org/10.1109/ICAIBDA53487.2021.9689756

[10] I. Kaibi, E. H. Nfaoui, and H. Satori, “10 A Comparative Evaluation of Word Embeddings Techniques for Twitter Sentiment Analysis,” in *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 2019, pp. 1–4. https://doi.org/10.1109/WITS.2019.8723864

[11] S. F. Sabbeh and H. A. Fasihuddin, “11 A Comparative Analysis of Word Embedding and Deep Learning for Arabic Sentiment Classification,” *Electronics (Basel)*, vol. 12, no. 6, 2023. https://doi.org/10.3390/electronics12061425

[12] C. Wang, P. Nulty, and D. Lillis, “12 A Comparative Study on Word Embeddings in Deep Learning for Text Classification,” in *Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval*, in NLPIR ’20. New York, NY, USA: Association for Computing Machinery, 2021, pp. 37–46. https://doi.org/10.1145/3443279.3443304

[13] D. S. Asudani, N. K. Nagwani, and P. Singh, “13 Impact of word embedding models on text analytics in deep learning environment: a review,” *Artif Intell Rev*, vol. 56, no. 9, pp. 10345–10425, 2023. https://doi.org/10.1007/s10462-023-10419-1

[14] S. H. Janjua, G. F. Siddiqui, M. A. Sindhu, and U. Rashid, “14 Multi-level aspect based sentiment classification of Twitter data: using hybrid approach in deep learning,” *PeerJ Comput Sci*, vol. 7, pp. 1–25, Apr. 2021. https://doi.org/10.7717/peerj-cs.433

[15] N. Alturaief, H. Aljamaan, and M. Baslyman, “15 AWARE: Aspect-Based Sentiment Analysis Dataset of Apps Reviews for Requirements Elicitation,” in *Proceedings - 2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops, ASEW 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 211–218. https://doi.org/10.1109/ASEW52652.2021.00049

[16] S. Gunathilaka and N. De Silva, “16 Aspect-based Sentiment Analysis on Mobile Application Reviews,” in *22nd International Conference on Advances in ICT for Emerging Regions, ICTer 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 183–188. https://doi.org/10.1109/ICTer58063.2022.10024070

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “17 Efficient Estimation of Word Representations in Vector Space,” Jan. 2013. https://doi.org/10.48550/arXiv.1301.3781

[18] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, “18 Learning Word Vectors for 157 Languages,” *CoRR*, vol. abs/1802.06893, 2018. https://doi.org/10.48550/arXiv.1802.06893

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “19 Efficient Estimation of Word Representations in Vector Space,” Jan. 2013. https://doi.org/10.48550/arXiv.1301.3781

[20] N. Dilawar *et al.*, “20 Understanding citizen issues through reviews: A step towards data informed planning in Smart Cities,” *Applied Sciences (Switzerland)*, vol. 8, no. 9, Sep. 2018. https://doi.org/10.3390/app8091589

[21] A. Hafeez *et al.*, “21 Addressing Imbalance Problem for Multi Label Classification of Scholarly Articles,” *IEEE Access*, vol. PP, p. 1, Jun. 2023. https://doi.org/10.1109/ACCESS.2023.3293852

[22] R. Ali, J. Hussain, and S. W. Lee, “22 Multilayer perceptron-based self-care early prediction of children with disabilities,” *Digit Health*, vol. 9, Jan. 2023. https://doi.org/10.1177/20552076231184054

[23] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, “23 Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark,” Sep. 2021. https://doi.org/10.48550/arXiv.2109.14545

[24]. Usha Ruby Dr.A, “24 Binary cross entropy with deep learning technique for Image classification,” *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 4, pp. 5393–5397, Aug. 2020. https://doi.org/10.30534/ijatcse/2020/175942020

[25] S. Chatterjee and A. Keprate, “25 Predicting Remaining Fatigue Life of Topside Piping Using Deep Learning,” in *2021 International Conference on Applied Artificial Intelligence, ICAPAI 2021*, Institute of Electrical and Electronics Engineers Inc., May 2021. https://doi.org/10.1109/ICAPAI49758.2021.9462055

[26] A. I. Ramadhan and E. B. Setiawan, “26 Aspect-based Sentiment Analysis on Social Media Using Convolutional Neural Network (CNN) Method,” *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 4, Mar. 2023. https://doi.org/10.47065/bits.v4i4.3103

[27] S. Riyanto, I. S. Sitanggang, T. Djatna, and T. D. Atikah, “27 Comparative Analysis using Various Performance Metrics in Imbalanced Data for Multi-class Text Classification.”. https://dx.doi.org/10.14569/IJACSA.2023.01406116

[28] C. Padurariu and M. E. Breaban, “28 Dealing with data imbalance in text classification,” in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 736–745. https://doi.org/10.1016/j.procs.2019.09.229

[29] M. Hayaty, S. Muthmainah, and S. M. Ghufran, “29 Random and Synthetic Over-Sampling Approach to Resolve Data Imbalance in Classification,” *International Journal of Artificial Intelligence Research*, vol. 4, no. 2, p. 86, Jan. 2021. https://doi.org/10.29099/ijair.v4i2.152

[30] M. R. Ilham and A. D. Laksito, “30 Comparative Analysis of Using Word Embedding in Deep Learning for Text Classification,” *Jurnal Riset Informatika*, vol. 5, no. 2, pp. 195–202, Mar. 2023.

[31] P. Mojumder, M. Hasan, M. F. Hossain, and K. M. A. Hasan, “31 A study of fasttext word embedding effects in document classification in bangla language,” in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, Springer, 2020, pp. 441–453. https://doi.org/10.1007/978-3-030-52856-0_35

[32] C. Yang, E. A. Fridgeirsson, J. A. Kors, J. M. Reps, and P. R. Rijnbeek, “32 Impact of random oversampling and random undersampling on the performance of prediction models developed using observational health data,” *J Big Data*, vol. 11, no. 1, Dec. 2024. https://doi.org/10.1186/s40537-023-00857-7

© 2024 The Authors. Published by Universitas Muhammadiyah Malang
This is an open access article under the CC BY NC SA license (https://creativecommons.org/licenses/by-nc-sa/4.0/)