DANA App Sentiment Analysis: Comparison of XGBoost, SVM, and Extra Trees

Muhamad Jodi Setiawan[1]*, Vinna Rahmayanti Setyaning Nastiti [2]
Department of Informatics, Faculty of Engineering [1], [2]
University of Muhammadiyah Malang
Malang, Jawa Timur, Indonesia
jodisetiawan15@webmail.umm.ac.id [1], vinastiti@umm.ac.id [2]

This research aims to analyze sentiment on DANA application reviews to find out user perceptions by comparing Extreme Gradient Boosting (XGBoost), Support Vector Machine (SVM), and Extra Trees Classifier classification methods. DANA application review data is obtained from the Kaggle site which consists of 50,000 Indonesian-language reviews labeled with positive and negative sentiments. The research stages include data preprocessing to clean and prepare the review text, applying word weighting using Word2Vec to give weight to words based on their context, balancing sentiment classes using SMOTE to address the imbalance of positive and negative review classes. It should be noted that the initial proportion of data before applying SMOTE may affect the results. The data is then divided into training and testing sets, then the models are trained and evaluated using Confusion Matrix and K-Fold Cross-Validation. The results of the three classification methods are measured by the accuracy matrix and F1-Score to assess model performance, the SVM and XGBoost methods obtained an accuracy of 93% and the ETC method achieved an F1-Score value of 96% at K=6, the three models proved to be very accurate in predicting the sentiment of DANA application reviews both positive and negative. The practical implications of this research can identify areas for application improvement, develop popular features, personalize services based on user preferences, and manage application reputation.

Keywords— Sentiment Analysis, XGBoost, Support Vector Machine, Extra Trees Classifier, Word2Vec, SMOTE.

I. INTRODUCTION

Technological advances continue to have a significant impact, causing more and more innovations that make daily activities easier. One of these innovations is the change in payments from cash to non-cash payments or digital payments (e-wallets) [1]. In Indonesia itself, there are various digital payment services (e-wallets) such as Go-Pay, OVO, DANA, Link Aja, and ShopeePay and so on. As the adoption of digital payments increases, user reviews are becoming increasingly important as a key indicator of the platform's performance [1]. Reviews not only reflect the level of user satisfaction, but also provide valuable insights into features of interest, areas for improvement, and the overall public perception of the platform. In this context, sentiment analysis of user reviews becomes a very useful tool for understanding more deeply about user experiences and expectations.

In previous research, machine learning or deep learning models were used to analyze sentiment on a variety of digital payments. Research conducted by [5] using the Support Vector Machine (SVM) method was able to achieve an accuracy of 82.33%. Other research's conducted by [6], [7], and [8] compared several machine learning models for sentiment analysis of digital payments in Indonesia. According to the findings, the SVM approach had the best accuracy, with 91.30%, 74.29%, and 89.0%. respectively. In research [9] used a dataset regarding (OVO, GO-PAY, and LinkAja) taken from Twitter. This study compared two machine learning models (Naïve Bayes and K-Nearest Neighbor or KNN). With a 20-fold cross validation of 91.00%, the KNN model had the highest accuracy, according to the data. In research [10] used 3,878 datasets from Twitter. A lexical dictionary was used to label the data, which was then split between 70% training data and 30% test data. The study findings demonstrated that the accuracy of the Naïve Bayes model was 88.56%. Research by [11] used 5,000 datasets that had been labeled with sentiment. Using Word2Vec as a text representation, this study compared two deep learning models: CNN (Convolutional Neural Network) and LSTM (Long Short Term Memory). The CNN model, with an accuracy of 86.00%, produced the best results. Research [12] used 2,000 OVO and DANA application review datasets that already had sentiment labels and compared two machine learning models (Naïve Bayes and Support Vector Machine or SVM). With an accuracy of 91.00% and an AUC of 98.60%, this study demonstrated that the SVM model with 10-fold cross-validation produced the best results.

Research [13] indicates that XGBoost is particularly excellent at managing unbalanced data, as seen by the 96.24% accuracy of the XGBoost approach applied in sentiment analysis of telemedicine application evaluations using a dataset of 10,353 positive and 1,197 negative reviews. Other studies also show that the XGBoost model provides the best results on natural language processing (NLP) tasks [14], [15], and [16]. Research [14] shows that the XGBoost model using Word2vec and Word2vec and Doc2vec feature extraction have average F1-Score values of 93.42% and 93.44%, respectively, demonstrating their ability to categorize unbalanced datasets. Research by [15] compares several models (XGBoost, LSTM, and SVM) by combining CNN and Word2Vec models as feature extraction. The results of the study show that XGBoost excels in classification with an F1-Score value of 93.16%. Other studies conducted by [13], [14], [15], and [16] compared two feature extractions (Bag-Of-Words and Word2Vec) and compared several machine learning models (Naïve Bayes, Random Forest, and XGBoost) the XGBoost model as a classifier and Word2Vec as a feature extraction in

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i3.2239, Copyright ©2024
Submitted : August 1, 2024, Revised : September 11, 2024, Accepted : September 16, 2024, Published : November 20, 2024

sentiment analysis, were able to provide the best results.

Based on previous research, although it has explored sentiment analysis in digital payment application reviews using various models, there has been no study that specifically compares the performance Using Support Vector Machine (SVM), Extra Trees Classifier (ETC), and Extreme Gradient Boosting (XGBoost) in evaluating e-wallet application. So in this research will analyze sentiment in e-wallet applications by comparing three classification methods as a comparison, namely Extreme Gradient Boosting (XGBoost), Extra Trees Classifier (ETC) and Support Vector Machine (SVM). The reseacher used a case study, namely the DANA e-wallet product because based on the results of research conducted by DailySocial, the most widely used digital wallet service in Indonesia is Go-Pay, This number represents 87% of all users. With 80.4% of users, OVO is in second place, while DANA is in third place with 75.6% [2]. Therefore, it may be concluded that improvements are still needed. Because a high-quality application may increase user happiness, which in turn increases the number of users, one method to evaluate DANA's performance and quality is to learn about the user experience via reviews and comments on the Google Play Store [3]. Reviews from users might reveal their opinions about the application's usability, service quality, and other aspects [4]. Therefore, a sentiment analysis of DANA application user reviews is needed.

Three classification methods used, while having advantages in handling text data, have limitations such as sensitivity to hyperparameters and potential overfitting. Hyperparameters need to be set carefully, choosing the wrong hyperparameters can affect the model's performance. This study did not use techniques such as grid search or random search for the optimal model. Overfitting may also occur in the three classification models used when the model is overfitted to the training data so that it cannot be generalised well to new data. By testing the model on many data sets, this research employs K-Fold Cross-Validation to lower the chance of overfitting.

II. RESEARCH METHODOLOGY

The system in this research is build using a sentiment analysis model derived from Google Play Store review of DANA applications. In the first stage, preprocessing is carried out to clean data noise and simplify the classification process. The Word2Vec technique is then used to weight the words, and the SMOTE method is used to balance the dataset's sentiment class. Then, divide the dataset into two categories: testing and training. Additionally, the Extreme Gradient Boosting, Support Vector Machine, and Extra Trees Classifier models are used for sentiment classification. The Confusion Matrix displays the evaluation model following sentiment classification. Figure 1 depicts the system build in this research.

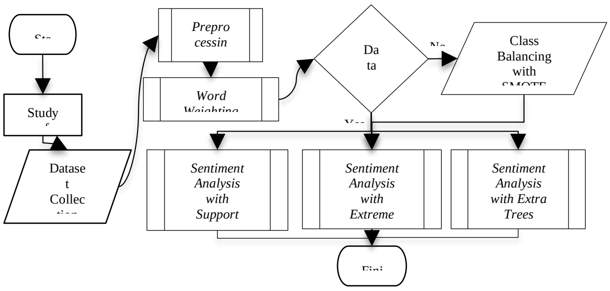
Study
Datase t Collec
Prepro cessin
Word
Da ta
Class Balancing with
Sentiment Analysis with Support
Sentiment Analysis with Extreme
Sentiment Analysis with Extra Trees

Figure 1. Research Workflow

Figure 1 shows the research workflow and each stage carried out starting from conducting a literature study, obtaining a dataset that matches the research topic from the Kaggle site, then conducting data preprocessing aimed at cleaning and preparing the raw text so that it is ready to be processed by the model, after the data is clean, word weighting is carried out by applying word weighting using Word2Vec to give weight to words based on their context, continued by seeing whether the data is balanced or not, if the data is not balanced, data balancing is carried out using the SMOTE method to overcome situations where one class has a much larger number of samples than the other class. The results of the data are trained with the Extreme Gradient Boosting, Support Vector Machine, and Extra Trees Classifier classification models to determine the classification of negative and positive sentiments. Furthermore, the classification results are evaluated using the Confusion Matrix and K-Fold Cross-Validation. The results of the three classification methods are measured by the accuracy matrix and F1-Score to assess model performance.

A. Dataset

In this research using dataset of reviews for the Dana application, sourced from Kaggle. The dataset contains 50,000 reviews written in Indonesian. While it represents only a portion of the approximately 6 million reviews available on the Google Play Store, it has been labeled into three categories: positive, neutral, and negative.

TABLE I. RESEARCH DATASET

Label	Sentence
Positive	“Terimakasih apl bagus,sangat berguna”.
Negative	"Saya kecewa dengan dana saldo saya tiba2 ada transaki ke akun tidak di kenal saya buat laporan pun tidak di tangangi dengan baik. Tolong lah tutup aja aplikasi nya ngerugiin banyak orang".

Table 1 is an example of positive and negative sentences from a dataset obtained from the Kaggle website about Dana application reviews. Data preparation was done by removing the distribution of neutral comments, leaving 43,628 data from the positive and negative classes.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i3.2239, Copyright ©2024
Submitted : August 1, 2024, Revised : September 11, 2024, Accepted : September 16, 2024, Published : November 20, 2024

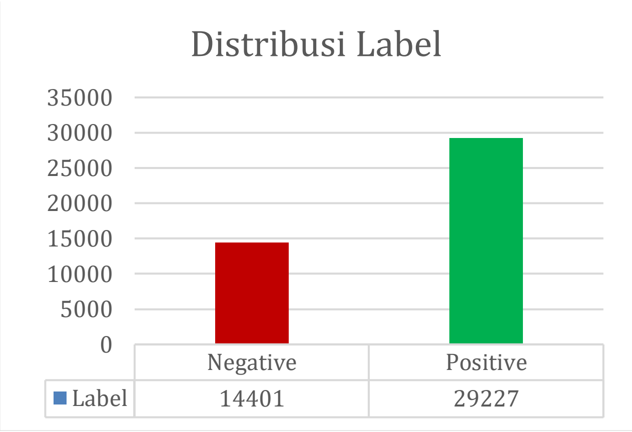

Figure 2. Label Distribution

Figure 2 presents data on positive and negative comments, where the number of positive comments was 29,227 and the number of negative comments was 14,401.

B. Preprocessing

Following the collection and preparation of all data, the preprocessing step begins. The purpose of preprocessing is to address issues with data processing. Figure 3 shows the preprocessing steps used in this research.

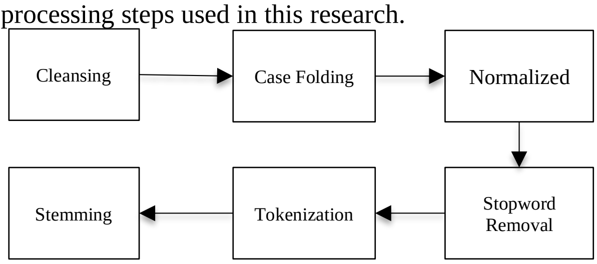

Figure 3. Preprocessing Stage

In this research, preprocessing is divides into six steps: Cleaning, Case Folding, Normalization, Stopword Removal, Tokenization, and Stemming. During preprocessing, the data in the review column is also modified. The data type is adjusted to assign the positive class a value of 1 and the negative class a value of 0.

- *Cleaning*

Cleaning is the process of removing extraneous elements from input data, such as punctuation and symbols [16]. Cleaning in this research data uses regular expression techniques to remove certain patterns in text such as symbols, numbers, emojis, words containing numbers, words that are less than or equal to 3 letters.

TABLE 2. CLEANING RESULTS

Sentence	Cleansing Results
“Terimakasih apl bagus,sangat berguna”	“Terimakasih apl bagus sangat berguna”

In Table 2, the comma after the word "bagus" was removed as part of the cleaning process. This is because punctuation marks like commas, periods, and question marks generally hold little significance in text analysis. By removing them, it will helps simplify the text and emphasizes the main words.

- *Case Folding*

On the case folding process, every character in the data is transformed into lowercase letters [17]. The outcome of the case folding is shown in Table 3.

TABLE 3. CASE FOLDING RESULTS

Cleaning Results	Case Folding Results
“Terimakasih apl bagus sangat berguna”	“terimakasih apl bagus sangat berguna”

Case folding aims to standardize text by converting all letters to lowercase. This ensures consistency during data processing and eliminates variations caused by capitalization differences. As a result, it helps reduce the number of features required for training.

- *Normalization*

With this normalization, words are converted into standard form without prefixes or suffixes using lemmatization technique to enable identification of similar words with the same meaning [19]. The outcome of the normalization is shown in Table 4.

TABLE 4. NORMALIZATION RESULTS

Case Folding Results	Normalization Results
“terimakasih apl bagus sangat berguna”	“terimakasih aplikasi bagus sangat berguna”

During this stage, normalization converts the word “apl” into “aplikasi” The process follows the standard form of the Indonesian word.

- *Stopword Removal*

Stopword removal removes meaningless and unimportant words in Indonesian [19]. The outcome of the stopword removal is shown in Table 5.

TABLE 5. STOPWORD REMOVAL RESULTS

Normalization Results	Stopword Results
“terimakasih aplikasi bagus sangat berguna”	“terimakasih aplikasi bagus berguna”

Table 5 presents the word that was removed at the stopwords stage, namely “sangat” because it tends not to carry specific information about the text itself.

- *Tokenization*

Tokenization is divided by technique (whitespace) or separated by spaces and functions as a sentence breaker based on each word that makes it up [19]. The outcome of the tokenization is shown in Table 6.

TABLE 6. TOKENIZATION RESULTS

Stopword Results	Tokenization Results
“terimakasih aplikasi bagus berguna”	“['terimakasih', 'aplikasi', 'bagus', 'berguna']”

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i3.2239, Copyright ©2024
Submitted : August 1, 2024, Revised : September 11, 2024, Accepted : September 16, 2024, Published : November 20, 2024

Table 6, the tokenization stage simplifies the text by breaking the words “terimakasih aplikasi bagus berguna” into token-units “['terimakasih', 'aplikasi', 'bagus', 'berguna']” which are considered semantically helpful.

- *Stemming*

Stemming is the process of changing words into essential words. Word affixes, including as suffixes, prefixes, and combinations, are throughout eliminated this procedure [18]. The outcome of the steaming is shown in Table 7.

TABLE 7. STEMMING RESULTS

Tokenization Results	Stemming Results
“['terimakasih', 'aplikasi', 'bagus', 'berguna']”	“terimakasih aplikasi bagus guna”

Table 7 is the result of stemming changing words in the text into basic forms or basic words, namely, the word "berguna" is changed to "guna".

After completing all preprocessing stages, the clean sentence is shown in table 8.

TABLE 8. RESULTS OF PREPROCESSING STAGE

Sentence	Clean Sentence
“Terimakasih apl bagus,sangat berguna”	“terimakasih aplikasi bagus guna”

At this stage, the resulting sentence is free from prefixes, words that are not important or have no meaning, symbols and punctuation, so that the next process can be carried out.

C. Split Data

After the preprocessing stage is complete, the next step is to split the dataset into training and testing data. In this research, 80% of the data is used for training, while the remaining 20% is used for testing. Table 9 shows the result of the split data.

TABLE 9. DATA SPLIT RESULTS

Split Data	Data Train	Data Test
Data Total	34.902	8.726

A total of 34,902 training samples will trained through oversampling and be processed using machine learning models. The resulting model will then be validated using the testing data to evaluate how effectively the proposed architecture addresses sentiment analysis problems.

D. Word2Vec

Weight the words using Word2Vec comes next, following the completion of the preprocessing and split data phases. Word2Vec is a deep learning-based method designed to represent words within a context as vectors in an N-dimensional space [20]. In this research, Word2Vec is employed for feature extraction. There are two types of Word2Vec models available: Continuous Bag of Words (CBOW) and Skip-Gram. CBOW predicts the context of a word based on previous words, while Skip-Gram predicts the context of nearby words by focusing on the middle word [21]. This research adopts the Skip-Gram model because it is effective for learning word vector representations from unstructured text [20]. The formula used in the Skip-Gram model are shown in Number 1 :

$$\frac{1}{T} = \sum_{t=1}^{T} \sum_{-c\leq j\leq c, j\neq 0} log\, p\left(w_{t+j}\middle|w_t\right) \qquad (1)$$

Description :

c	= Steps in the context of training
w_{t+j}	= Words after the middle word
w_t	= Middle word
$p(w_{t+j}\|w_t)$	= Probability of words in the middle word

In this research, Word2Vec is drilled specifically on tokenized training data with the configuration w2v_model = Word2Vec (sentences=X_train_tokenized, vector_size=300, window=5, sg=1, hs=0, min_count=1, negative=5, epochs=1000). The vector dimension used is 300, the context window size is 5, using the Skip-Gram model where sg=1, using the negative sampling training method or hs=0, ignoring all words with a frequency of less than 1, with min_count to help reduce the size of the area and focus on more important words. negative=5 which means the number of negative samples used in negative sampling. epochs=1000 is the number of training iterations on all training data.

E. Synthetic Minority Oversampling Technique (SMOTE)

The imbalance of each class in the dataset determines the level of validity and accuracy of a model. Good dataset quality can be obtained with good consistency and level of confidence. The imbalance of dataset classes can be overcome by using SMOTE [22]. This method makes use of the K-Nearest Neighbor idea, after which SMOTE contributes to the creation of synthetic data from the minority class [23]. Using this technique, two minority samples are linearly interpolated to produce new minority samples [24]. In the case of multi-class classification with neutral classes, SMOTE can be applied with two approaches: ignoring the neutral class or oversampling the neutral class with a lower factor than the minority class. The implementation of SMOTE can be seen in Figure 4.

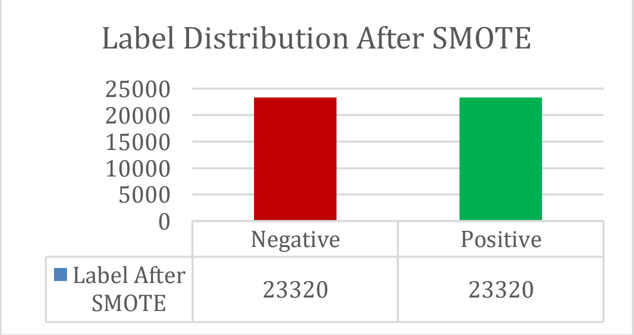

Figure 4. SMOTE Implementation Results

Figure 4 shows that by performing SMOTE, researcher can balance the dataset using the train values from Word2Vec. Because there is a data gap between the positive and negative classes, it is imperative that the dataset classes be balanced. In order to match the number of samples in the majority class, or the positive class, SMOTE will generate synthetic data from the negative class that is included in the minority class. This will enhance the model's ability to identify patterns in the minority class and less bias towards the majority class.

F. Support Vector Machine

Support Vector Machine is a machine learning method used for data classification. Supervised learning includes in this technique. Labeling the data is therefore necessary [25,26]. Support Vector Machine works to separate data by finding the best hyperplane and maximum margin [27]. Margin is a distance between the class's outermost samples, also known as the Support Vector. While a hyperplane is a plane that separates or distinguishes two classes. Number 2 shows the Support Vector Machine formula [26].

$$w.x - b = 0 \tag{2}$$

Where w is the weight vector, x is the input vector, and b is the bias. The selection of this SVM method also tends to be relatively strong against overfitting. The main hyperparameters used in SVM include regularization parameters, kernel functions, and the influence of training samples.

G. XGBoost Classifier

The gradient-boosting framework is used by the XGBoost algorithm, a machine learning method for regression analysis and decision tree-based classification [13]. Several learning methods are used in the ensemble classifier technique to improve performance [28]. The formula of XGBoost is shown in Number 3 :

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_{i,}\hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \tag{3}$$

Where n is the number of models to be used, l l is a function to measure the difference between the target predictions y_i AND $\hat{y}_i$, $f_t(x_i)$ is a new model built. While the function Ω is to prevent the model from overfitting [28].

The choice of the XGBoost method is because this model is a sophisticated implementation of the gradient boosting algorithm which is known for its high speed and performance. XGBoost is often the main choice because of its ability to handle high-dimensional data, data that has many features, and data that has a non-linear relationship between features and targets. The main hyperparameters used in XGBoost include the number of trees, tree depth, learning rate, proportion of samples and features used, and the minimum value of loss reduction.

H. Extra Trees Classifier

The use of the Extra Trees Classifier model is also based on the fact that this algorithm is a variation of Random Forest that adds more randomness to the tree building process. This can help reduce overfitting and improve model generalization, especially when the dataset has a lot of noise or irrelevant features. Extra Trees Classifier has hyperparameters such as the number of trees, tree depth, minimum number of samples for node division, and maximum number of features considered.

The method of Extra Trees, an ensemble version of the fundamental Decision Tree algorithm, is comparable to Random Forest in that it generates a Decision Tree based on several bootstraps. Classification decisions are determined based on the majority of decisions from the Decision Tree [29]. The usage of bootstraps is where Extra Trees and Random Forest diverge. Unlike Extra Trees, Random Forest generates bootstraps by randomly selecting data from the dataset. The way each Decision Tree is constructed also varies. While Random Forest uses best-split to select nodes in the Decision Tree, Extra Trees utilizes random-split [29].

I. Evaluation

Evaluation procedure in this research is carried out using the Confusion Matrix. The value of numerous points required for this step is recall, precision, accuracy, and F1-Score, is determined using the Confusion Matrix [30]. The Confusion Matrix can be seen in Table 10.

TABLE 10. CONFUSION MATRIX

Confusion Matrix		Factual Value	
		Positif	Negatif
Prediction Value	Positive	TP	FP
	Negative	FN	TN

Description :
TP = Cases where the prediction is positive, and it is actually positive (true positive)
FN = Cases where the prediction is negative, but it is actually positive (false negative)
FP = Cases where the prediction is positive, but it is actually negative (false positive)
TN = Cases where the prediction is negative, and it is actually negative (true negative)

The performance of the classification method’s is measured using f1-score, precision, and recall. The following is the formula for evaluating performance :

$$F1 - Score = \frac{2 \times Recall \times Precision}{Precision + Recall} \tag{4}$$

The F1-Score is a balanced average that determines a classification method's performance by taking recall and accuracy values [20].

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{5}$$

Precision is the ratio of the number of items correctly identified as positive to those identified as positive [21].

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{6}$$

True Positive Rate (TPR), often known as recall, is the ratio of properly recognized relevant things to correctly identified objects [20].

Confusion matrix provides a detailed picture of the model's correct and incorrect predictions for each class, while evaluation metrics such as accuracy, precision, recall, specificity, and F1-score dig deeper to measure the model's performance from different perspectives. T When dealing with imbalanced datasets, accuracy alone may not fully represent the model's performance. Metrics like recall and F1-score are more reliable for evaluating how well the model identifies the minority class, which is often the primary focus.

J. Cross-Validation

A statistical technique called Cross-Validation, also known as K-Fold Cross-Validation, is used to determine how well a machine learning model predicts unseen data. It is a common evaluation approach in machine learning applications due to its ease of use and useful outcomes [13]. The generic K-Fold Cross Validation process may be written as follows :

1) Dataset will be shuffle randomly.
2) Split the dataset into k folds.
3) Each fold will serve as the validation data for iteration k, while the remaining folds will serve as the training data. The validation data is then used to evaluate the model.
4) Summarize the model quality by using the average score from each iteration.

These results show whether the model has stable overall performance. If there is a spike in high or low scores, there may be other problems. This study sets k fold to k = 10 because this value is commonly used in machine learning applications [12].

III. RESULT AND DISCUSSION

A. Sentiment Analysis

This research analyzes sentiment with a dataset that has been labelled for each sentence. There are 29,227 reviews with positive sentiment and 14,401 reviews with negative sentiment. Then, preprocessing is carried out to clean the data. The results of preprocessing can be seen in Table 8. After the preprocessing is complete, the dataset is split into 80% for training and 20% for testing, resulting in 34,902 samples for training and 8,726 samples for testing. Then, the next step is involves feature extraction using Word2Vec, where each word is represented as a vector. After extracting features, the next step is implementing SMOTE to balance the minority and majority classes. The results of the SMOTE implementation can be seen in Figure 2. Then, classify using the Extreme Gradient Boosting method, Support Vector Machine, and Extra Trees Classifier. Data exploration is done by visualizing the most frequently occurring words. These comments will later be constructive in analyzing what factors are most complained about by users.

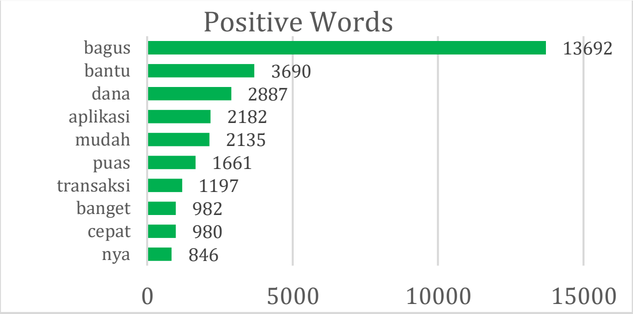

Figure 5. Distribution of Positive Sentiment Words

Figure 5 shows some dominant words that appear in positive reviews, meaning that these words are often written and appear a lot in reviews given by users. The most frequently appearing word is 'bagus' which appears 14,000 times. Other dominant words that also appear in positive reviews are 'bantu', 'dana', 'aplikasi', 'mudah', 'puas', 'transaksi', 'banget', 'cepat', and others. Examples of these words show that users are satisfied with the easy fund application and help with fast transactions. This shows the strength of DANA that must be maintained in providing services to users.

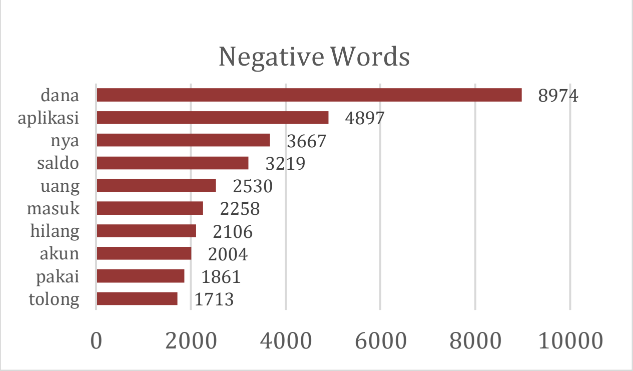

Figure 6. Distribution of Negative Sentiment Words

Figure 6 shows some words that predominantly appear in negative reviews. The most frequently appearing word is 'dana', which appears 8,000 times. Other dominant words that also appear in negative reviews are 'aplikasi', 'saldo', 'uang', 'masuk', 'hilang', 'akun', 'pakai', 'tolong', and others. These examples of words show that users are expressing complaints about the application. This also highlights the shortcomings of DANA that need to be fixed, such as balances suddenly disappearing, money not coming in, and accounts that are sometimes difficult to log in to.

B. Evaluation Model

K-Fold Cross-Validation and Confusion Matrix is used to evaluate the model. Table 11-13 displays the outcomes of each fold Cross-Validation. Figure 7-10 displays the Confusion Matrix, a performance metric for machine learning classification issues.

TABLE 11. XGBOOST METRICS EVALUATION

Sentiment	Precision	Recall	F1-Score	Accuracy
Negative	89%	88%	88%	93%
Positive	94%	95%	95%	

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i3.2239, Copyright ©2024
Submitted : August 1, 2024, Revised : September 11, 2024, Accepted : September 16, 2024, Published : November 20, 2024

Table 11 shows that the XGBoost model performs well in sentiment classification. This model achieves a precision of 89% for negative sentiment and 94% for positive sentiment and a recall of 88% for negative sentiment and 95% for positive sentiment. The f1-score, the mean between precision and recall, describes the class imbalance in the dataset. The f1-score for negative sentiment is 88%, and the f1-score for positive sentiment is 95%. The overall accuracy for this model is 93%.

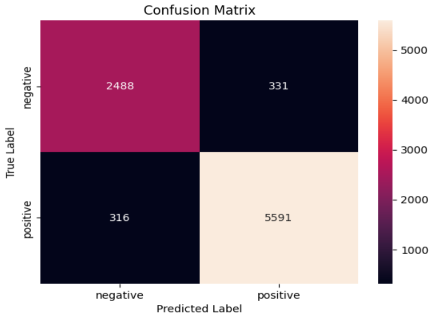

Figure 7. Confusion Matrix XGBoost

Figure 7 shows the confusion matrix results, with 2488 true negatives, 331 false positives, 316 false negatives, and 5591 true positives.

TABLE 12. SUPPORT VECTOR MACHINE METRICS EVALUATION

Sentiment	Precision	Recall	F1-Score	Accuracy
Negative	90%	88%	89%	93%
Positive	94%	95%	95%	

Table 12 shows that the Support Vector Machine model produces 90% precision for negative sentiment and 94% for positive sentiment, 88% recall for negative sentiment and 95% for positive sentiment, and an f1-score of 89% for negative sentiment and 95% for positive sentiment. The overall accuracy for this model is 93%.

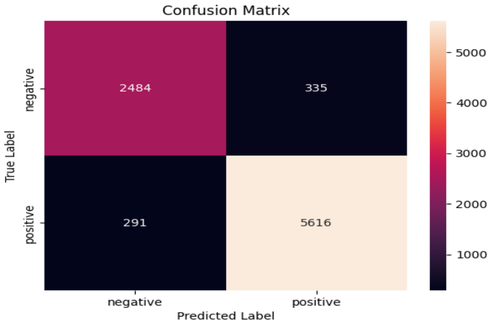

Figure 8. Confusion Matrix SVM

Figure 8 shows the results of the confusion matrix with 2484 true negatives, 335 false positives, 291 false negatives, and 5616 true positives.

TABLE 13. EXTRA TREES CLASSIFIER METRICS EVALUATION

Sentiment	Precision	Recall	F1-Score	Accuracy
Negative	86%	89%	88%	92%
Positive	95%	93%	94%	

Table 13 shows that the Extra Trees Classifier model produces 86% precision for negative sentiment and 95% for positive sentiment, 89% recall for negative sentiment and 93% for positive sentiment, and an f1-score of 88% for negative sentiment and 94% for positive sentiment. The overall accuracy for this model is 92%.

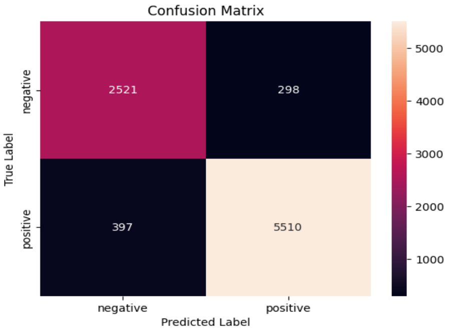

Figure 9. Confusion Matrix Extra Trees

Figure 9 shows the confusion matrix results, with 2521 true negatives, 298 false positives, 397 false negatives, and 5510 true positives.

Confusion Matrix results of each model are shown in Figures 7-9. Then for the performance evaluation of each model shown in Tables 11-13, the results show that the Extreme Gradient Boosting model and the SVM model provide comparable results of 93%. While the Extra Trees model gives a result of 92%.

TABLE 14. RESULT OF K-FOLD CROSS VALIDATION XGBOOST

Iterasi	F1-Score
1	93.44%
2	92.36%
3	93.32%
4	93.17%
5	93.63%
6	95.34%
7	95.32%
8	94.76%
9	95.30%
10	95.02%

Table 14 shows the model performance changes along with iterations in the Extreme Gradient Boosting model, the 6th iteration is the best with an f1-score of 95.34% which means the model achieves its best performance at this iteration. The 2nd iteration is the worst with an f1-score of 92.36% which means the model achieves its lowest performance at this iteration.

TABLE 15. RESULTS OF K-FOLD CROSS VALIDATION SVM

Iterasi	F1-Score
1	92.29%
2	92.33%
3	92.60%
4	92.21%
5	93.04%
6	93.73%

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i3.2239, Copyright ©2024
Submitted : August 1, 2024, Revised : September 11, 2024, Accepted : September 16, 2024, Published : November 20, 2024

7	94.30%
8	93.86%
9	94.19%
10	93.58%

Table 15 shows the model performance changes along with iterations in the Support Vector Machine model; the 7th iteration is the best, with an f1-score of 94.30%, which means the model achieves its best performance at this iteration. The 4th iteration is the worst, with an f1-score of 92.21%, which means the model achieves its lowest performance at this iteration.

TABLE 16. RESULTS OF K-FOLD CROSS VALIDATION EXTRA TREES

Iterasi	F1-Score
1	93.08%
2	92.58%
3	92.73%
4	92.42%
5	93.16%
6	95.46%
7	94.64%
8	94.38%
9	94.77%
10	94.61%

Table 16 shows the model performance changes along with iterations in the Extra Trees Classifier model. The 6th iteration is the best, with an f1-score of 95.46%, which means the model achieves its best performance in this iteration. The 4th iteration is the worst, with an f1-score of 92.42%, which means the model achieves its lowest performance in this iteration.

TABLE 17. COMPARISON OF THREE ALGORITHMS

Algorithm	*Precision*	*Recall*	*F1-Score*	*Accuracy*
Support Vector Machine	92%	92%	92%	93%
Extreme Gradient Boosting	92%	91%	92%	93%
Extra Trees Classifier	91%	91%	91%	92%

Table 17 compares the performance of the Support Vector Machine, Extreme Gradient Boosting and Extra Trees Classifier classification algorithms. The highest precision is obtained from the Support Vector Machine algorithm, and Extreme Gradient Boosting means that the algorithm's accuracy in predicting positive samples has the same ability. In contrast, the highest recall is obtained from the Support Vector Machine algorithm, which indicates that the algorithm's accuracy in predicting negative samples is better than other algorithms.

Extra Trees Classifier algorithm generates the highest f1-score value, so these three models can well predict positive and negative comments.

IV. CONCLUSION

This research implements XGBoost, SVM, and Extra Trees Classifier algorithms to analyze sentiment on DANA app reviews. The dataset consists of 29,227 positive reviews and 14,401 negative reviews. 33% of the reviews fall into the negative class, so it is necessary to improve the features and transaction problems so that the reputation of the application can increase in the future. The SMOTE method is used to overcome the class imbalance because it affects the model's performance. SMOTE can improve the model's performance because the model is trained with the same amount of data from each classification class. Conversely, suppose the model is trained with data with significant class differences. In that case, its performance in determining the minority class will decrease because it will be taught with fewer minority samples. The results obtained after applying the SMOTE method are 23,320 reviews in each class. The evaluation using the configuration matrix shows that the SVM and XGBoost methods achieve an accuracy of 93% with a data split ratio of 80:20, slightly higher than the accuracy of Extra Trees method of 92%. Evaluation with K-Fold Cross Validation showed that Extra Trees Classifier method achieved the best F1-Score value of 96% at K=6.

This research has limitations in the amount of data used, potential bias in the review data caused by the use of the SMOTE method, and the limited number of techniques used only for the XGBoost, SVM, and Extra Trees methods. So, further research is recommended to use a more extensive and diverse dataset and other methods and compare various oversampling techniques. Algorithm optimization can also be done to improve classification performance further. In addition, further research can explore the use of application, including reviews from multiple platforms or other data sources such as social media. Other features such as review metadata such as date, location and application version can also be considered for more in-depth analysis.

REFERENCES

[1] M. Najib and F. Fahma, “Investigating the adoption of digital payment system through an extended technology acceptance model: An insight from the Indonesian small and medium enterprises,” Int. J. Adv. Sci. Eng. Inf. Technol., vol. 10, no. 4, pp. 1702–1708, 2020, doi: 10.18517/ijaseit.10.4.11616.

[2] Y. Dinda Oktaviani Waruwu, “Sistem Pendukung Keputusan Pemilihan E – Wallet Terbaik Dengan Menggunakan Metode Analytical Hierracy Process (AHP),” J. Ilm. Sain dan Teknol., vol. 2, no. 2, pp. 101–116, 2024.

[3] H. Ammar Faris, dkk, “DeLone and McLean Model Analysis of Success Factors of SIDEMANG Application in Palembang City,” Sisfokom., vol. 13, no. 2, p. 160, Juny 2024, doi: 10.32736/sisfokom.v13i2.1894.

[4] P. A. Permatasari, L. Linawati, and L. Jasa, “Survei Tentang Analisis Sentimen Pada Media Sosial,” Maj. Ilm. Teknol. Elektro, vol. 20, no. 2, p. 177, Dec. 2021, doi: 10.24843/MITE.2021.v20i02.P01.

[5] K. Kusnawi, M. Rahardi, and V. D. Pandiangan, “Sentiment Analysis of Neobank Digital Banking using Support Vector Machine Algorithm in Indonesia,” JOIV Int. J. Informatics Vis., vol. 7, no. 2, p. 377, May 2023, doi: 10.30630/joiv.7.2.1652.

[6] D. A. Putri, D. A. Kristiyanti, E. Indrayuni, A. Nurhadi, and D. R. Hadinata, “Comparison of Naive Bayes Algorithm and Support Vector Machine using PSO Feature Selection for Sentiment Analysis on E-Wallet Review,” J. Phys. Conf. Ser., vol. 1641, no. 1, p. 012085,

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i3.2239, Copyright ©2024
Submitted : August 1, 2024, Revised : September 11, 2024, Accepted : September 16, 2024, Published : November 20, 2024

Nov. 2020, doi: 10.1088/1742-6596/1641/1/012085.

[7] B. Andrian, T. Simanungkalit, I. Budi, and A. F. Wicaksono, “Sentiment Analysis on Customer Satisfaction of Digital Banking in Indonesia,” Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 3, pp. 466–473, 2022, doi: 10.14569/IJACSA.2022.0130356.

[8] B. Harnadi and A. D. Widiantoro, “Evaluating the Performance and Accuracy of Supervised Learning Models on Sentiment Analysis of E-Wallet,” in 2023 7th International Conference on Information Technology (InCIT), IEEE, Nov. 2023, pp. 175–180. doi: 10.1109/InCIT60207.2023.10413111.

[9] H. Wisnu, M. Afif, and Y. Ruldevyani, “Sentiment analysis on customer satisfaction of digital payment in Indonesia: A comparative study using KNN and Naïve Bayes,” J. Phys. Conf. Ser., vol. 1444, no. 1, p. 012034, Jan. 2020, doi: 10.1088/1742-6596/1444/1/012034.

[10] F. R. Ananda Dimas Sanjaya, Tacbir Hendro Pudjiantoro, Ade Kania Ningsih, “Sentiment Analysis Of E-Wallets on Twitter social media With Naïve Bayes and Lexicon-Based Methods,” in Proceedings of the International Conference on Industrial Engineering and Operations Management, Michigan, USA: IEOM Society International, 2022, pp. 1002–1010. doi: 10.46254/AP03.20220200.

[11] M. Omarkhan, G. Kissymova, and I. Akhmetov, “Handling data imbalance using CNN and LSTM in financial news sentiment analysis,” Proc. - 2021 16th Int. Conf. Electron. Comput. Comput. ICECCO 2021, pp. 1–8, 2021, doi: 10.1109/ICECCO53203.2021.9663802.

[12] D. A. Kristiyanti, D. A. Putri, E. Indrayuni, A. Nurhadi, and A. H. Umam, “E-Wallet Sentiment Analysis Using Naïve Bayes and Support Vector Machine Algorithm,” J. Phys. Conf. Ser., vol. 1641, no. 1, 2020, doi: 10.1088/1742-6596/1641/1/012079.

[13] K. Afifah, I. N. Yulita, and I. Sarathan, “Sentiment Analysis on Telemedicine App Reviews using XGBoost Classifier,” 2021 Int. Conf. Artif. Intell. Big Data Anal. ICAIBDA 2021, pp. 22–27, 2021, doi: 10.1109/ICAIBDA53487.2021.9689762.

[14] I. R. Hendrawan, E. Utami, and A. D. Hartanto, “Comparison of Word2vec and Doc2vec Methods for Text Classification of Product Reviews,” Proceeding - 6th Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng. Appl. Data Sci. Artif. Intell. Technol. Environ. Sustain. ICITISEE 2022, pp. 530–534, 2022, doi: 10.1109/ICITISEE57756.2022.10057702.

[15] A. N. Azhar, M. L. Khodra, and A. P. Sutiono, “Multi-label Aspect Categorization with Convolutional Neural Networks and Extreme Gradient Boosting,” Proc. Int. Conf. Electr. Eng. Informatics, vol. 2019-July, no. July, pp. 35–40, 2019, doi: 10.1109/ICEEI47359.2019.8988898.

[16] M. Anita, R. Mannava, M. L. Deep, and M. V. R. Durga Prasad, “Using Machine Learning to Analyze Twitter’s Sentiment,” Proc. 2nd IEEE Int. Conf. Adv. Comput. Commun. Appl. Informatics, ACCAI 2023, pp. 1–8, 2023, doi: 10.1109/ACCAI58221.2023.10199230.

[17] M. Syarifuddinn, “Analisis Sentimen Opini Publik Terhadap Efek Psbb Pada Twitter Dengan Algoritma Decision Tree,Knn, Dan Naïve Bayes,” INTI Nusa Mandiri, vol. 15, no. 1, pp. 87–94, 2020, doi: 10.33480/inti.v15i1.1433.

[18] N. D. Kusumawati, S. Al Faraby, and M. Dwifebri, “Analisis Sentimen Komentar Beracun pada Media Sosial Menggunakan Word2Vec dan Support Vectore Machine (SVM),” e-Proceeding Eng., vol. 8, no. 5, pp. 10038–10050, 2021.

[19] V. K. S. Que, A. Iriani, and H. D. Purnomo, “Analisis Sentimen Transportasi Online Menggunakan Support Vector Machine Berbasis Particle Swarm Optimization,” J. Nas. Tek. Elektro dan Teknol. Inf., vol. 9, no. 2, pp. 162–170, 2020, doi: 10.22146/jnteti.v9i2.102.

[20] A. Fahmi Sabani, Adiwijaya, and W. Astuti, “Analisis Sentimen Review Film pada Website Rotten Tomatoes Menggunakan Metode SVM Dengan Mengimplementasikan Fitur Extraction Word2Vec,” e-Proceeding Eng., vol. 9, no. 3, p. 1800, 2022.

[21] D. I. Af’idah, D. Dairoh, S. F. Handayani, and R. W. Pratiwi, “Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen,” J. Inform. J. Pengemb. IT, vol. 6, no. 3, pp. 156–161, 2021, doi: 10.30591/jpit.v6i3.3016.

[22] G. Yang and L. Qicheng, “An Over Sampling Method of Unbalanced Data Based on Ant Colony Clustering,” IEEE Access, vol. 9, pp. 130990–130996, 2021, doi: 10.1109/ACCESS.2021.3114443.

[23] E. D. N. Sari and I. Irhamah, “Analisis Sentimen Nasabah pada Layanan Perbankan Menggunakan Metode Regresi Logistik Biner, Naïve Bayes Classifier (NBC), dan Support Vector Machine (SVM),” J. Sains dan Seni ITS, vol. 8, no. 2, Feb. 2020, doi: 10.12962/j23373520.v8i2.44565.

[24] W. Fangyu, Z. Jianhui, B. Youjun, and C. Bo, “Research on imbalanced data set preprocessing based on deep learning,” in 2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS), IEEE, Jan. 2021, pp. 75–79. doi: 10.1109/ACCTCS52002.2021.00023.

[25] K. R. Kavitha, A. Gopinath, and M. Gopi, “Applying improved svm classifier for leukemia cancer classification using FCBF,” in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, Sep. 2017, pp. 61–66. doi: 10.1109/ICACCI.2017.8125817.

[26] S. Zahoor and R. Rohilla, “Twitter Sentiment Analysis Using Machine Learning Algorithms: A Case Study,” in 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM), IEEE, Aug. 2020, pp. 194–199. doi: 10.1109/ICACCM50413.2020.9213011.

[27] R. M. Simranjot Kaur, “Sentiment Analysis on twitter data using Machine Learning,” J. Xidian Univ., vol. 14, no. 12, Dec. 2020, doi: 10.37896/jxu14.12/039.

[28] M. T. Akter, M. Begum, and R. Mustafa, “Bengali Sentiment Analysis of E-commerce Product Reviews using K-Nearest Neighbors,” in 2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD), IEEE, Feb. 2021, pp. 40–44. doi: 10.1109/ICICT4SD50815.2021.9396910.

[29] M. Syukron, R. Santoso, and T. Widiharih, “PERBANDINGAN METODE SMOTE RANDOM FOREST DAN SMOTE XGBOOST UNTUK KLASIFIKASI TINGKAT PENYAKIT HEPATITIS C PADA IMBALANCE CLASS DATA,” J. Gaussian, vol. 9, no. 3, pp. 227–236, Aug. 2020, doi: 10.14710/j.gauss.v9i3.28915.

[30] A. S. Aribowo, H. T. Jaya, U. Pembangunan, and N. Veteran, “An Evaluation of Preprocessing Steps and Tree-based Ensemble Machine Learning for Analysing Sentiment on Indonesian YouTube Comments,” Int. J. Adv. Trends Comput. Sci. Eng., vol. 9, no. 5, 2020, doi: 10.30534/ijatcse/2020/29952020.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v13i3.2239, Copyright ©2024
Submitted : August 1, 2024, Revised : September 11, 2024, Accepted : September 16, 2024, Published : November 20, 2024