2021 International Conference on Artificial Intelligence and Big Data Analytics (ICAIBDA) | 978-1-6654-0890-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICAIBDA53487.2021.9689735

Sentiment Analysis on Telemedicine App Reviews using XGBoost Classifier

Khansa Afifah[ad1], Intan Nurma Yulita[bd2], Indra Sarathan[cd3]
[a]*Department of Mathematics*
[b]*Department of Informatics Engineering*
[c]*Department of Indonesian Literature*
[d]*Research Center for Artificial Intelligence and Big Data*
Universitas Padjadjaran
Sumedang, Indonesia
e-mail: [1]khansa18001@mail.unpad.ac.id, [2]intan.nurma@unpad.ac.id, [3]sarathan@unpad.ac.id

***Abstract*— In recent years, companies have widely used sentiment analysis with machine learning classification algorithms to help business decision-making. Sentiment analysis helps evaluate customer opinions on a product in goods or services. Companies need this opinion or sentiment to improve the performance, quality of their products, and customer satisfaction. Machine learning algorithms widely used for sentiment analysis are Naive Bayes Classifier, Maximum Entropy, Decision Tree, and Support Vector Machine. In this study, we propose an approach of sentiment analysis using a very popular method, Extreme Gradient Boosting or XGBoost. XGBoost combines weak learners into an ensemble classifier to build a strong learner. This study will focus on the reviews data of the most popular telemedicine application in Indonesia, Halodoc. This study aims to examine the people'ssentiment towards telemedicine applications in Indonesia, especially during the COVID-19 pandemic. We also showed a fishbone diagram to analyze the most factors the users complained about. The data we have are imbalanced; however, XGBoost can perform well with 96.24% accuracy without performing techniques for imbalanced data.**

Keywords—sentiment analysis, machine learning, telemedicine, reviews, xgboost.

I. Introduction

Amid the COVID-19 pandemic that has hit Indonesia since early 2020, the government has taken several steps to prevent the spread of COVID-19 community activities. The government is also actively involved in ensuring that the public adheres to health protocols. Even though so many people do not comply with the rules set by the government, daily COVID-19 cases in Indonesia also reached a record high on July 14, 2021, which increased by 54,517 points. As a result, the number of patients admitted to the hospital grew, but the limited capacity of the hospital was not able to accommodate all patients. This impacts medical personnel who feel overwhelmed by the increasing number of COVID-19 patients. Finally, the Ministry of Health and the government urged the public to use telemedicine. Telemedicine applications aim to reduce the number of patients who are not seriously ill in hospitals to increase the capacity of patients who need treatment. COVID-19 patients who are self-isolating at home can use the telemedicine application to see a doctor at any time, either looking for prescription drugs or buying drugs from pharmacies and then sending them to their homes. Halodoc is a telemedicine application that is very popular in Indonesia and has been very successful at the age of 5 years. On August 13, 2020, CB Insights named Halodoc one of the startups on the Digital Health 150 list, including 150 of the most promising digital health companies. This is the second year in a row that Halodoc has entered the Virtual Care Delivery category at this event. Halodoc has various health services, namely Pharmacy Delivery, Contact Doctor, Appointment, and others. During the pandemic of COVID-19, Halodoc also provides vaccination programs and COVID-19 tests to support the government in breaking the chains of transmission of Coronavirus. Halodoc still needs to improve the quality of their product and evaluate their performance by looking at the data of their product sentiment. With reviews on the google play store, companies can use them to see public sentiment regarding the version of their application.

In this study, an analysis of public sentiment will be carried out on the usage of Halodoc telemedicine application in Indonesia. Sentiment analysis, also known as opinion mining, is one of the most critical tasks in natural language processing. Sentiment analysis works by analyzing a text's mood, emotion, or feeling towards products, services, individuals, organizations, and events [8]. The sentiment is then classified as positive, negative, or neutral. Sentiment analysis is often used to improve decision-making and customer satisfaction in a company's business process. Sentiment analysis has become an exciting research topic in various fields such as products reviews [6], [17], services [5], [17], politics [1], [2], and even in gaming chat applications to reveal the existing of cyberbullying among online gamers [11]. Data for sentiment analysis can mainly be obtained from social media such as Twitter and Facebook, application reviews in google play store and app store, blogs, and websites. Sentiment analysis has been widely applied to various languages. In [5], the authors conducted a sentiment analysis on an e-payment service in Jordan which the data were in Arabic taken from social media Facebook and Twitter. The authors revealed that one of the most complex challenges when they had to handle Arabic was Dialectical Arabic, which is an understanding that can help identify the context. Another study [17] has conducted sentiment analysis on online product reviews in Chinese.

In recent years, sentiment analysis has been used with machine learning dan deep learning models. In [7], the authors conducted sentiment analysis research using the proposed method, namely SentiXGBoost. SentiXGBoost is an XGBoost model that combines several models such as Decision Tree, Naïve Bayes, Random Forest, KNN, LR, and SGD to be trained as base classifiers and got an accuracy of 90.8%. In [5], the authors analyzed sentiment using a proposed methodology that combines a neutrality detector model with XGBoost and genetic algorithms. Inspired by [5] and [7], we used XGBoost for this study and applied it in the

978-1-6654-0890-5/21/$31.00 ©2021 IEEE

field of healthcare service reviews in Indonesia. We used the Sastrawi library to handle preprocessing sentiment analysis steps in the Indonesian language.

II. Methodology

In this section, we present the methodology used in this study. There are five stages carried out, as shown in Fig 1. The first stage is data collection. In this stage, we collected the reviews data of Halodoc applications using the google-play-scraper library in Python. After collecting the data, we labeled the sentiment of each review. In the second stage, we conducted data preprocessing to the dataset, consisting of case folding, punctuation removal, stopwords removal, tokenization, normalization, and stemming. Data preprocessing aims to remove noise, so the texts are cleaner and more understandable. In the third stage, we performed feature extraction to transform text data into numeric or vector data. Next, we divided the dataset into training data and test data with a proportion of 75:25. We built a machine learning classification model in the next stage and trained our model to predict positive and negative sentiment. We evaluated the model using k-fold cross-validation and confusion matrix in the last stage.

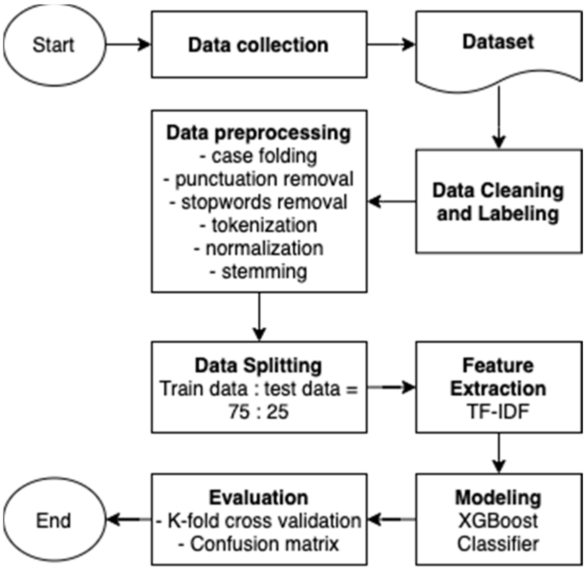

Fig. 1. Proposed Methodology

A. Data Collection

Data were collected from Google Play Store using the google play scraper library in Python. We got 12,969 data reviews from January 1st – September 30th, 2021. The data reviews contain users' names, content, rating scores, dates, and replies. However, we only used the content, rating score columns.

B. Data Cleaning and Labelling

After we collected the data, it was saved into a .csv format file. The reviews contain emojis, which we don't need for analysis, so we removed the emojis using the emoji library in Python. After removing emojis, we continued to label our data. We mapped the rating score using the definition determined by google play store to label the data. Score 1 for negative, 2 for somewhat negative, 3 for neutral, 4 for reasonably positive, and 5 for positive. We only take the negative and positive labels; it yields 11,550 data. We label negative as 0 and positive as 1, as illustrated in Fig. 2.

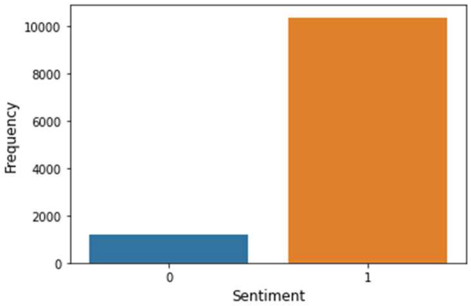

Fig. 2. Number of reviews based on sentiment

More detailed numbers for the figure above can be seen in Table 1.

TABLE I. Number of Reviews Based on Sentiment

Sentiment	Number of Reviews
Positive	10,353
Negative	1,197

The data we have is imbalanced. However, we did not perform any technique to handle imbalanced data in this study. We wanted to determine how good the XGBoost model is on an imbalanced dataset.

C. Data Preprocessing

In natural language processing (NLP), the information used contains unstructured data and a lot of noise. Therefore, it is necessary to convert the form into structured data before further processing. The data used in this study is reviews data in Bahasa Indonesia, therefore in this preprocessing stage, we used the Python Sastrawi library that can be accessed on github.com/har07/PySastrawi, which is a simple library to help text preprocessing such as stopword removal and stemming.

- Case Folding

  Case folding is a preprocessing stage to make all letters lowercase or uppercase. In most NLP cases, it is to convert all letters lowercase. Case folding aims to avoid two or more words with the same meaning but are treated differently by the machine due to writing in different forms; lowercase and uppercase.

- Punctuation Removal

  Punctuation removal aims to remove all the punctuation marks from sentences. A punctuation mark doesn't add extra information to the ruling. By removing punctuation marks, the dimension of our dataset can be reduced.

- Stopword Removal

  Stopword removal is one of the most common preprocessing steps used in NLP applications. The idea is to remove the most common words across all the documents. Stopword does not add much information to the text. Articles, prepositions, pronouns, and conjunction can be classified as stopwords.

- Tokenization

  Tokenization breaks the raw text into small units called tokens. These tokens help to understand the context in developing the model for NLP. Tokenization allows us to interpret the meaning of the text by analyzing the order of the words.

- Normalization

  Normalization is the process of converting a token to its basic form. The normalization process removes the inflected form of a word so that the basic form can be preserved. Normalization also transforms the short words or abbreviations into their complete form. For example, the term "tdk" is changed to its complete form, which is "tidak," and the word "baguuus" is transformed to its base, which is "bagus." For normalization, we used the abbreviation dictionary by meisaputri21 [13].

- Stemming

  Stemming is a step to remove affixes in a word, both affixes that appear before and after the word. Stemming converts each word to its root word without affixes.

D. Feature Extraction

Machines or algorithms cannot understand characters/words, it can only accept numbers as input. However, the inherent nature of textual data is unstructured and noisy, making it impossible to interact with machines. The process of converting raw text data into machine-readable formats (numbers) or features is called feature extraction from text data. There are certain techniques we can use for feature extraction such as Count Vectorizer and TF-IDF. TF-IDF stands for Term Frequency Inverse Document Frequency.

TF is simply a method to calculate the frequency of the occurrence of certain words in a document. The more often the word appears, the greater the TF value. While the IDF calculates the weight of a word against its appearance in the entire document. The more these words appear throughout the document, the smaller the IDF value. TF-IDF is calculated as Eq. 1.

$$IDF = \log\left(\frac{N}{DF}\right) \quad (1)$$

TF(k,d) denotes the number of the word shown in document d, while IDF(k) denotes the inverse document frequency, as shown in Eq. 2.

$$TF - IDF(d,k) = TF(d,k) \times IDF(k) \quad (2)$$

E. XGBoost Classifier

The XGBoost or Extreme Gradient Boosting algorithm was first developed as a research project at the University of Washington by Tianqi Chen and Carlos Guestrin. XGBoost is an implementation of gradient boosted decision trees designed to improve speed and performance. XGBoost is known for the ability to optimize the consumption of time, memory resources, and handle imbalanced data. The XGBoost or Extreme Gradient Boosting algorithm is a decision tree-based ensemble machine learning algorithm that uses a gradient boosting framework. Ensemble learning offers a solution to combine the predictive power of multiple learners. In boosting, trees are built sequentially so that each next tree aims to reduce errors from the previous tree. Each tree learns from its predecessors and updates residual errors. Therefore, the tree that grows next in the sequence learns from the updated residuals. The base learners in boosting are weak learners in which the bias is high. Each of these weak learners contributes to give some information for prediction, enabling the boosting technique to produce a strong learner by effectively combining these weak learners. Suppose we have a training data $x_i$and their labels y_i, XGBoost utilize classifier to predict the final prediction $\widehat{y}_i^{\,t}$

$$\widehat{y}_i^{\,t} = \sum_{k=1}^{t} f_k(x_i) = \widehat{y}_i^{\,t-1} + f_t(x_i) \quad (3)$$

Where $\widehat{y}_i^{\,t-1}$ is previous prediction and $f_t(x_i)$ is new prediction. To get a good model, in XGBoost we need to minimize the following objective function.

$$\mathcal{L}^t = \sum_{i=1}^{n} l(y_i, \widehat{y}_i\,) + \Omega(f_t) \quad (4)$$

The objective function contains loss function $l(y_i, \widehat{y}_i\,)$ and regularization term $\Omega(f_t)$. With the existing of (3), we can rewrite the objective function as follow.

$$\mathcal{L}^t = \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^{\,t-1} + f_t(x_i)\right) + \Omega(f_t) \quad (5)$$

The loss function measures how well the model fits on the training data, while regularization measures the complexity of trees. Optimizing loss function encourages predictive models for higher accuracy while optimizing regularization encourages generalized simpler models. Regularization is also utilized to avoid the model from overfitting. To enhance the XGBoost model, we can tune the hyper-parameters which are similar to decision tree hyper-parameters such as learning rate, max depth, n_estimators, and sub-sample. Learning rate and n_estimators are two critical hyper-parameters for gradient boosting algorithms. The learning rate parameter has a role to control the step weight. In other words, it tells us how fast the model learns. While n_estimators is the number of decision trees in XGBoost. If we set it to 1 then it makes the algorithm generate only a single tree. To yield the best performance of XGBoost, the model needs careful tuning of its parameters

.

F. Cross-Validation

Cross-validation, also known as K-Fold Cross Validation, is a statistical method to estimate the quality of machine learning models to predict unseen data. It is a popular evaluation method in machine learning applications because it is easy to understand and gives beneficial results. The general procedure of K-Fold Cross Validation can be written as follow.

1) shuffle the dataset randomly
2) split the dataset into k folds
3) for k iteration, each fold will become validation data, and the rest will become training data, fit the model on training data, evaluate the model on the validation data
4) summarize the quality of the model using the mean of scores from each iteration.

This allows us to see if our model has stable performance overall folds. There might be other problems if there are spikes in high scores or low scores. In this study, we set k=10 because it is widespread to use in machine learning applications.

III. Result and Discussion

This section explains how we performed the preprocessing, feature extraction, modeling, and evaluation.

A. Data Preprocessing

We have applied the preprocessing steps already mentioned in the previous section to the dataset. We can see the result of each preprocessing step in Table 2. After the data were preprocessed, we explored the data by visualizing the most frequent words in Fig. 3-6. The comments later will be helpful to analyze what factors the users complain about the most.

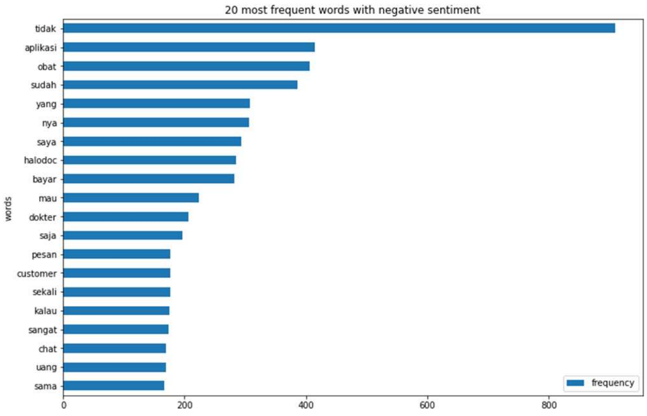

Fig. 3. Most frequent words with negative sentiment

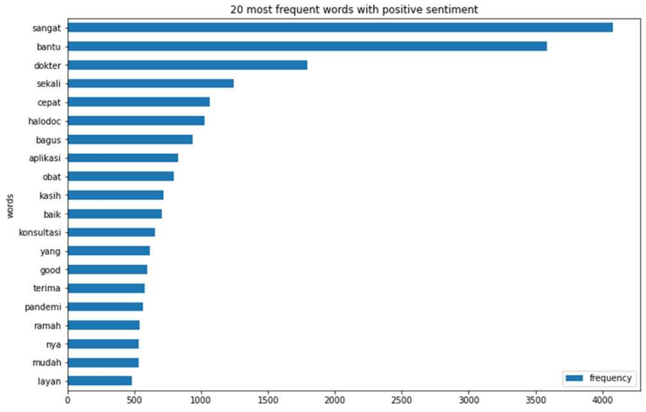

Fig. 4. Most frequent words with positive sentiment

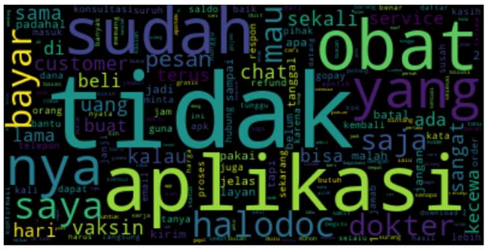

Fig. 5. Wordcloud of negative sentiment words

TABLE II. Sample Result of Data Preprocessing

Data	Result
Case Folding	
Saya kecewa sekali, ya. Konsul gak bisa, diminta cari dokter pengganti, tapi saldo uang terpotong. Padahal saya lagi khawatir kondisi anak saat itu dan butuh penanganan cepat.	saya kecewa sekali, ya. konsul gak bisa, diminta cari dokter pengganti, tapi saldo uang terpotong. padahal saya lagi khawatir kondisi anak saat itu dan butuh penanganan cepat.
Punctuation Removal	
saya kecewa sekali, ya. konsul gak bisa, diminta cari dokter pengganti, tapi saldo uang terpotong. padahal saya lagi khawatir kondisi anak saat itu dan butuh penanganan cepat.	saya kecewa sekali ya konsul gak bisa diminta cari dokter pengganti tapi saldo uang terpotong padahal saya lagi khawatir kondisi anak saat itu dan butuh penanganan cepat
Stopword Removal	
saya kecewa sekali ya konsul gak bisa diminta cari dokter pengganti tapi saldo uang terpotong padahal saya lagi khawatir kondisi anak saat itu dan butuh penanganan cepat	kecewa sekali konsul gak diminta cari dokter pengganti saldo uang terpotong padahal lagi khawatir kondisi anak itu butuh penanganan cepat
Tokenization	
kecewa sekali konsul gak diminta cari dokter pengganti saldo uang terpotong padahal lagi khawatir kondisi anak itu butuh penanganan cepat	kecewa, sekali, konsul, gak, diminta, cari, dokter, pengganti, saldo, uang, terpotong, padahal, lagi, khawatir, kondisi, anak, itu, butuh, penanganan, cepat
Normalization and Stemming	
kecewa, sekali, konsul, gak, diminta, cari, dokter, pengganti, saldo, uang, terpotong, padahal, lagi, khawatir, kondisi, anak, itu, butuh, penanganan, cepat	kecewa, sekali, konsul, tidak, minta, cari, dokter, ganti, saldo, uang, potong, padahal, lagi, khawatir, kondisi, anak, itu, butuh, tangan, cepat

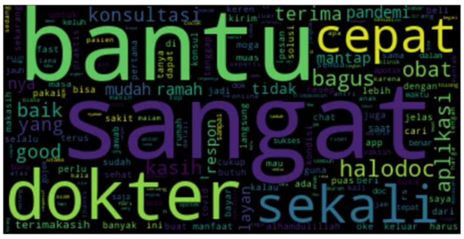

Fig. 6. Wordcloud of positive sentiment words

Based on the visualization above, the most frequent words with negative sentiment are 'tidak,' 'aplikasi,' and 'obat.' The most frequent words with a positive view are 'sangat,' 'Bantu,' and 'dokter.' We created a fishbone diagram to identify the possible cause of the problem, as shown in Fig 7.

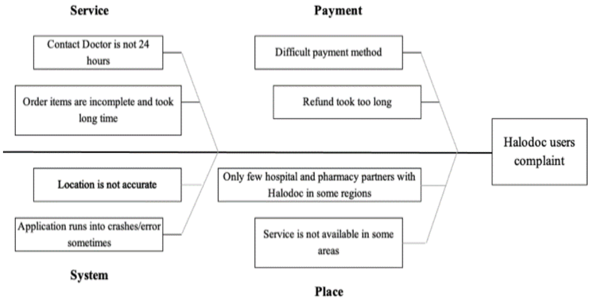
Service
Payment
Contact Doctor is not 24 hours
Difficult payment method
Order items are incomplete and took long time
Refund took too long
Halodoc users complaint
Location is not accurate
Only few hospital and pharmacy partners with Halodoc in some regions
Application runs into crashes/error sometimes
Service is not available in some areas
System
Place

Fig. 7. Fishbone diagram of possible causes of the problem

We found out four main factors the users most complain about; payment, place, service, and system. Users complain about difficult payment methods and refunds that took too long. Users also complain about the service that has not available in some regions, the incomplete order items, and the crashes or errors that sometimes happen on apps.

B. *Training*

After preprocessing the data, we split our dataset into train data and test data with a proportion of 75:25. We got 8383 rows of train data and 2795 rows of test data. Then, we performed feature extraction using TF-IDF to convert text into vectors. We used the TfidfVectorizer function, which is available in the sklearn library in Python. After extracting features, we defined our model and performed training using our data train. This experiment was conducted in Python using an open-source machine learning library, Scikit-Learn. To enhance our model, we did hyperparameter tuning with the following parameters in Table 3.

TABLE III. LIST OF PARAMETERS AND ITS BEST VALUES

Parameters	Definition	Values	Best Values
learning_rate	Step size	[0.05, 0.1, 0.3, 0.5]	0.1
n_estimators	The number of trees built in the model	[500,1000]	500
max_depth	Maximum number of tree depth	[4, 6, 8]	4

C. *Model Evaluation*

We evaluated the model using K-Fold Cross-Validation and confusion matrix. The results of each fold of cross-validation can be seen in Table 4. A confusion matrix is a performance measurement for machine learning classification problems, is shown in Table 5. The confusion matrix for our model can be seen in Fig 8. With the confusion matrix, we can get the value of accuracy, precision, recall, and f1-score.

TABLE IV. XGBOOST CLASSIFIER EVALUATION METRICS

Sentiment	Precision	Recall	F1-Score
Negative	0.85	0.77	0.81
Positive	0.97	0.98	0.98

TABLE V. K-FOLD CROSS VALIDATION RESULTS

Iteration	F1-Score
1	97.8%
2	98.5%
3	98.2%
4	97.8%
5	97.8%
6	98.5%
7	97.7%
8	97.5%
9	98.6%
10	97.6%

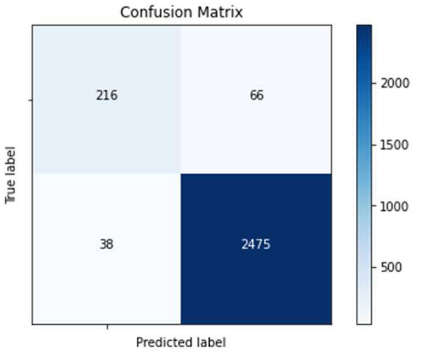

Fig. 8. Confusion Matrix

The equations 4-6 can calculate the three evaluation measures above:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (4)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5)$$

$$F1 - Score = 2 \times \frac{Precision\ \times\ Recall}{Precision + Recall} \quad (6)$$

IV. CONCLUSION

This study aims to examine public sentiment towards using a telemedicine application in Indonesia, especially during the pandemic of COVID-19. We choose Halodoc since it is the most popular telemedicine application in Indonesia. In this study, we use the XGBoost classifier, which is known as the best algorithm in terms of speed and performance. XGBoost combines weak learners to build strong learners. To enhance the result of XGBoost, we performed hyperparameter tuning using the grid search method by setting the value of the parameter of learning_rate, n_estimators, and max_depth into specific values that have been shown in the previous section. This study proves another assumption about XGBoost, which says XGBoost is quite good at handling imbalanced data. We got 96.24% of accuracy. Overall, the public sentiment towards Halodoc is quite good. People appreciate Halodoc's services, especially during the pandemic of COVID-19. However, we analyzed the negative reviews and found the four main factors Halodoc can improve: payment, place, service, and system. Users complain about the problematic payment method, refund, unavailability of service in some areas, unsatisfying services, and crashes on apps. We suggest using one of the techniques for handling imbalanced data and compare to this study.

REFERENCES

[1] M. Z. Ansari, M. B. Aziz, M. O. Siddiqui, H. Mehra, and K. P Singh, "Analysis of political sentiment orientations on Twitter", Procedia Computer Science, 167, 1821-1828, 2020.

[2] W. Budiharto, and M. Meiliana, M, "Prediction and analysis of Indonesia Presidential election from Twitter using sentiment analysis", Journal of Big data, 5(1), 1-10, 2018.

[3] J. Brownlee, A Gentle Introduction to k-fold Cross-Validation, 2018, Accessed: Sept 30, 2021, https://machinelearningmastery.com/k-fold-cross-validation/.

[4] J. Brownlee, Extreme Gradient Boosting (XGBoost) Ensemble in Python, 2020,. Accessed: Sept 30, 2021.

https://machinelearningmastery.com/extreme-gradient-boosting-ensemble-in-python/.

[5] D. A. Al-Qudah, A. M. Al-Zoubi, P. A. Castillo-Valdivieso and H. Faris. *Sentiment Analysis for e-Payment Service Providers Using Evolutionary eXtreme Gradient Boosting*. IEEE Access, vol. 8, pp. 189930-189944, 2020, doi: 10.1109/ACCESS.2020.3032216.

[6] B. Gaye, and A Wulamu, "Sentimental analysis for online reviews using machine learning algorithms", International Research Journal of Engineering and Technology (IRJET), *6*(08), 2395-0056, 2018.

[7] R. Hikmat, and N. Dimililer, "SentiXGboost: enhanced sentiment analysis in social media posts with ensemble XGBoost classifier", Journal of the Chinese Institute of Engineers, 44:6, 562-572, DOI: 10.1080/02533839.2021.1933598, 2021.

[8] A. Kulkarni, and A. Shivananda, Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python. 10.1007/978-1-4842-4267-4, 2021.

[9] G. Kundi, G. How to Scrape Google Play Reviews in 4 simple steps using Python. Accessed: Sept 30, 2021. https://www.linkedin.com/pulse/how-scrape-google-play-reviews-4-simple-steps-using-python-kundi/.

[10] B. Liu, "Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies", 2018, https://doi.org/10.2200/S00416ED1V01Y201204HLT016.

[11] S. Murnion, W. J. Buchanan, A. Smales, and G. Russell, "Machine learning and semantic analysis of in-game chat for cyberbullying". ArXiv, abs/1907.10855.

[12] G. A. Ruz, P. A. Henríquez, and A. Mascareño, "Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers", Future Generation Computer Systems, 106, 92-104, 2016.

[13] M. S. Saputri, R. Mahendra, and M. Andriani, M. "Emotion Classification on Indonesian Twitter Dataset", Proceeding of International Conference on Asian Language Processing 2018.

[14] S. E. Saad and J. Yang, "Twitter Sentiment Analysis Based on Ordinal Regression," in IEEE Access, vol. 7, pp. 163677-163685, 2019, doi: 10.1109/ACCESS.2019.2952127.

[15] J. Singh, G. Singh, and R. Singh, "Optimization of sentiment analysis using machine learning classifiers", Hum. Cent. Comput. Inf. Sci. **7,** 32, 2017

[16] B. N. Supriya, and C. B. Akki, "Sentiment Prediction using Enhanced XGBoost and Tailored Random Forest", International Journal of Computing and Digital Systems, 2021.

[17] L. Yang Y. Li, J.Wang, J., and R. S. Sherratt, "Sentiment analysis for E-commerce product reviews in Chinese based on sentiment lexicon and deep learning, IEEE Access, 8, 23522-23530, 2020.