Classification of Natural Disaster Reports from Social Media using K-Means SMOTE and Multinomial Naïve Bayes

Nor Indrani[1], Mohammad Reza Faisal[1]*, Irwan Budiman[1] , Dwi Kartini[1] , Friska Abadi [1] , Septyan Eka Prastya[2] , Mera Kartika Delimayanti[3]

[1] Computer Science Department, Universitas Lambung Mangkurat
Jl. A. Yani Km 36, Banjarbaru, Kalimantan Selatan, INDONESIA
[2] Teknologi Informasi Department, Universitas Sari Mulia
Jl. Pramuka No.2, Banjarmasin, Kalimantan Selatan, INDONESIA
[3] Department of Computer and Informatics Engineering, Politeknik Negeri Jakarta
Jl. Prof. DR. G.A. Siwabessy, Depok, Jawa Barat, INDONESIA
Email: nor.indriani5@gmail.com, reza.faisal@ulm.ac.id, irwan.budiman@ulm.ac.id, dwikartini@ulm.ac.id, friska.abadi@ulm.ac.id, septyan.e.prastya@unism.ac.id, mera.kartika@tik.pnj.ac.id

Corresponding Author

Abstract **Disasters can occur anytime and anywhere. Floods and forest fires are two types of disasters that occur in Indonesia. South Kalimantan Province is an area that frequently experiences floods and forest fires. The dataset used for previous research's flood and forest fire disaster data is unbalanced. Unbalanced data conditions can complicate the classification method in carrying out the classification process. The sampling method for the data level approach that can be used to solve imbalance problems is oversampling, one of the derivatives of oversampling, namely SMOTE. The K-Means SMOTE method is a modification of SMOTE. One Naïve Bayes model often used in text classification is Multinomial Naïve Bayes. Multinomial Naïve Bayes has a good performance in classifying text. The research results on flood disaster data using K-Means SMOTE with Multinomial Naïve Bayes yielded an f1 score of 66.04%, and forest fire disaster data using K-Means SMOTE with Multinomial Naïve Bayes produced an f1 score of 66.31%.**

Keywords**: unbalanced data; natural disasters; k-means smote; classification; multinomial naïve bayes**

I. Introduction

Indonesia is a country that frequently experiences natural disasters, including the province of South Kalimantan. Natural disasters that occurred in South Kalimantan were floods and forest fires. Twitter social media can be a place to share and receive important information during a disaster. The extensive collection of Twitter data about disaster information will make it difficult for someone to find out the existing disaster messages, so it needs to be processed using the concept of text mining to find disaster messages.

Research on disaster messages was carried out by [1]. In this research, the data used was flood and forest fire disaster data from Twitter, where the data used was not balanced. In essence, data in the real world are primarily unbalanced. Unbalanced data conditions can make it difficult for classification methods to process data mining. Data imbalance has a negative impact on the classification results, where the minority class is often misclassified as the majority class. Sometimes, the minority class is even more critical to identify than the majority class.

Categorized datasets have unbalanced classes if one class has a smaller number than the others. The class imbalance problem can be handled by two approaches, namely the data level approach and the algorithm level approach. The sampling method in the data level approach that can be used to solve class imbalance problems is oversampling, one of the derivatives of oversampling, namely SMOTE. In research [2] using SMOTE to handle unbalanced data and using Naïve Bayes and Random forest for classification, combining SMOTE with Naïve Bayes produces an f-measure of 0.896. In contrast, SMOTE with Random Forest makes an f-measure of 0.883. However, the SMOTE method has a weakness, namely it randomly selects minority class instances to be oversampled using a uniform probability, so it is prone to generate data noise because it does not distinguish overlapping class areas. To overcome the weaknesses of the SMOTE method, the K-Means SMOTE method has been developed, which performs better than the SMOTE algorithm. In research [3] using K-means SMOTE to handle unbalanced data, the use of K-Means SMOTE shows increased performance compared to the SMOTE method for all classification methods, namely SVM, Naïve Bayes and C4.5. The combination of K-Means SMOTE with SVM has better accuracy and sensitivity, namely 82% and 77%, while Naïve Bayes produces the best specificity, namely 89%.

One Naïve Bayes model often used in text classification is Multinomial Naïve Bayes. Multinomial Naïve Bayes performs well in classifying text. In research

[4], the Multinomial Naïve Bayes method shows promising results for unbalanced data classification, where Multinomial Naïve Bayes produces an accuracy of 73.2% and a recall of 75.3%.

II. Methods

The processed in this research can be seen in Fig 1.

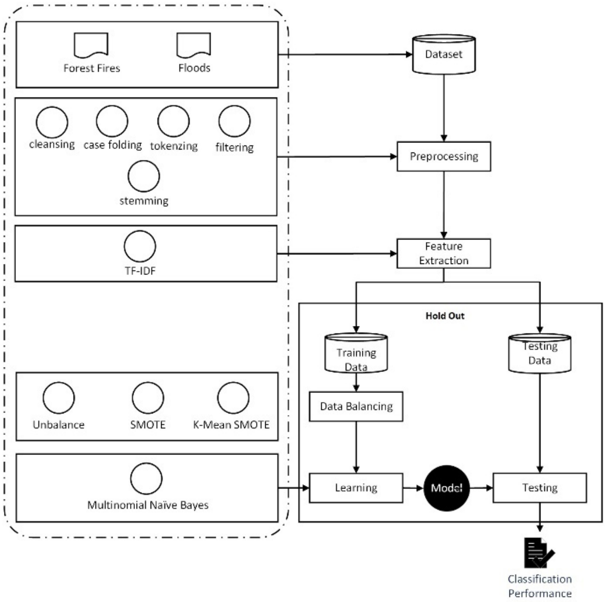

Fig 1. Research procedure.

A. Data

Natural disasters that often occur in South Kalimantan are floods and forest fires. This is why we use data on natural disaster messages from floods and forest fires from Twitter.

The data used in this study are flood and forest fire disaster data from Twitter (tweets) taken from research [1]. This data has three classes that identify the tweet's source: eyewitness, don't know, and non-eyewitness. The amount of data on floods and forest fires from each class can be seen in Table I.

TABLE I. Dataset

Data	#	Class		
		Don't Know	Eyewitness	Non-Eyewitness
Flood	2000	822	627	551
Forest fire	2000	432	189	1379

This dataset is data with an unequal number of samples between classes. For example, in the forest fire dataset, the number of samples in the don't know class is 432, and the sample class eyewitness is 189. The number of samples for the two classes is very different compared to those in the non-eyewitness class. The difference in the number of samples makes this research a case of unbalanced data classification.

B. Preprocessing

Preprocessing is a process for preparing data to be structured by converting data into a form that is easy for the system to process so that data can be analyzed. Commonly used preprocessing techniques are removing user names, removing RTs, case folding, filtering, stemming, removing numbers and punctuation marks, removing stopwords, and tokenization [5].

C. TF-IDF

In the field of natural language processing (NLP), feature extraction refers to the process of converting raw text data into a numerical format that can be effectively understood and processed by machine learning algorithms. This step is essential in NLP research as it allows for the extraction of valuable information from text data and enables the application of various machine learning techniques..

Several methods have been developed to extract features from text data, and one popular approach is the Term Frequency-Inverse Document Frequency (TF-IDF) method. TF-IDF is a technique used to convert textual data into numeric data by assigning weights to each word or feature. It is a statistical measure that assesses the importance of a word in a document. TF represents the frequency of a word's occurrence in a given document, indicating its significance. DF represents the frequency of documents that contain the word, indicating its commonness. IDF is the inverse of the DF value. The result of applying TF-IDF is obtained by multiplying the TF value with the IDF value for each word. The weight of a word is higher if it appears frequently in a document and lower if it appears in multiple documents [6].

The TF-IDF score is calculated using the following formula:

$$TF - IDF(t, d) = TF(t, d) * IDF(t) \tag{1}$$

Where $TF(t, d)$ represents the Term Frequency, which measures the frequency of a term (t) in a specific document (d). It can be calculated by simply counting the number of occurrences of the term within the document. $IDF(t)$ stands for Inverse Document Frequency, which measures the rarity of a term (t) across the entire corpus. It is computed as the logarithm of the total number of documents (N) divided by the number of documents containing the term (df(t)), and then taking the inverse.

TF-IDF has been widely used in many text-based applications, including information retrieval, text classification, and document clustering. It provides a way to capture the semantic importance of terms within a document and helps to identify distinctive features that contribute to the overall meaning of the text. Additionally, TF-IDF can handle both short and long documents effectively, making it a versatile feature extraction technique for textual data.

D. *Multinomial Naïve Bayes*

Naïve Bayes is a probabilistic machine learning method. As the name suggests, this method assumes that each data attribute is independent. The assumption that each word is independent of one other in the Naïve Bayes method is contrary to the actual situation [7]. This is because a document or text needs to have interconnected words so that the document has meaning. However, this method is proven to provide satisfactory results when applied in the field of text classification. One Naïve Bayes model often used in text classification is Multinomial Naïve Bayes. Multinomial Naïve Bayes is a supervised learning method, so each data must be labeled before training is carried out [8].

E. *SMOTE*

SMOTE (Synthetic Minority Over-sampling Technique) is a widely used algorithm in machine learning and data mining for addressing the class imbalance problem. The class imbalance problem refers to the situation where the number of instances in one class (referred to as the minority class) is significantly lower than the number of instances in the other class (referred to as the majority class). This imbalance can lead to biased predictions and reduced performance of machine learning models, as the minority class may be overshadowed by the majority class during training.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip proposed a technique known as Synthetic Minority Over-sampling Technique (SMOTE) [9]. SMOTE addresses the issue of class imbalance in data by increasing the number of samples in the minority class. This is achieved by randomly replicating data points from the minority class to match the quantity of data in the majority class. SMOTE is an oversampling method specifically designed for the minority class, which involves generating synthetic data points to overcome class imbalance. SMOTE generates artificial data by considering the nearest neighbors based on the k-nearest neighbor principle. The value of k, representing the number of nearest neighbors, is determined with ease of implementation [10].

F. *K-Means SMOTE*

K-means SMOTE is an extension of the SMOTE (Synthetic Minority Over-sampling Technique) algorithm, which addresses the class imbalance problem in machine learning. It combines the concepts of SMOTE and k-means clustering to generate synthetic samples for the minority class. It developed by Douzas, Bacao and Last, 2018 [11].

The K-means SMOTE algorithm works as follows. The first step, Perform k-means clustering on the minority class instances: Initially, the minority class instances are clustered into k clusters using the k-means algorithm. Each cluster represents a group of similar minority instances. The second step, Identify the nearest neighbors: For each minority class instance, the algorithm identifies its k nearest neighbors within the same cluster. This step is similar to the process in the original SMOTE algorithm. The third step, Generate synthetic samples: Synthetic samples are created by interpolating between the selected instance and its nearest neighbors within the same cluster. The interpolation process is performed as in the traditional SMOTE algorithm. Generate synthetic samples: Synthetic samples are created by interpolating between the selected instance and its nearest neighbors within the same cluster. The interpolation process is performed as in the traditional SMOTE algorithm [12].

In summary, K-means SMOTE is an extension of SMOTE that incorporates k-means clustering to generate diverse synthetic samples for the minority class. It offers advantages such as improved diversity, reduced risk of overfitting, better scalability for large datasets, and enhanced generalization. These benefits make K-means SMOTE a valuable technique for addressing the class imbalance problem and improving the performance of machine learning models on imbalanced datasets.

G. *Classification Performance Measurement*

A confusion matrix is a tool that functions to analyze whether or not the classification model is good at recognizing data patterns from different classes. The performance evaluation of a classification model is determined using various value terms in the confusion matrix [13]. The confusion matrix comprises four key metrics: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These metrics are computed based on the model's correct and incorrect predictions.

In the case of multiclass classification, the confusion matrix extends the binary classification version and offers a comprehensive assessment of a classification model's performance when dealing with more than two classes. It aids in understanding the model's predictive accuracy for each class and identifying any misclassifications. The multiclass confusion matrix is a square matrix with dimensions corresponding to the number of classes involved in the classification task. Each row in the matrix represents instances from the true class, while each column represents instances predicted to belong to a specific class. The elements in the confusion matrix represent the count or proportion of instances falling into each category. The diagonal elements of the matrix represent the correctly classified instances, where the predicted class matches the true class. The off-diagonal elements represent the misclassified instances, where the predicted class differs from the true class.

The confusion matrix in the multiclass classification type that uses three class outputs can be presented as shown below in Table II.

TABLE II. CONFUSION MATRIX

Actual	**Prediction**		
	Don't Know	**Eyewitness**	**Non-Eyewitness**
Don't Know	DD	DE	DN
Eyewitness	ED	EE	EN
Non-eyewitness	ND	NE	NN

Precision is the level of accuracy or thoroughness in classification. Meanwhile, recall measures the proportion of actual positives that are correctly identified. Here are the calculations for precision and recall [14].

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}} \quad (1)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \quad (2)$$

The F1-score is the harmonic average of precision and recall; the higher the value, the better the performance [11] [12]. This F1-Score or F-Measure is used to measure the minority class classification in unbalanced classes, and the F1-Score can also measure the performance of a model which is the harmonic average of recall and precision [17]. F-Measure is an evaluation that is used for imbalance by combining recall/sensitivity and precision so that it is effective in performance for searching information in data with imbalance problems. F-Measure is the harmonic average between recall and precision so that if the F-Measure has a high value, it can guarantee that the value of recall and precision is high. The following is the calculation for the F1-Score.

$$\text{f1(micro \& macro)} = \frac{\text{2 x Precisionx Recall}}{\text{Precision+ Recall}} \quad (3)$$

The use of macro-averaged calculates the average of each f1 value from each class. For macro and micro calculations, the formula used is the same, it's just that there are several different stages. In the first macro calculation, namely looking for each confusion matrix for each class, calculating precision, recall, and f1 after getting each f1 from each class, the macro value is obtained from the average f1 value. In contrast, micro calculations make a confusion matrix where the contents are the sum of the true positives of all classes, true negatives of all classes, false negatives of all classes, and false positives of all classes after that we just calculate precision, recall, and f1-score [18].

III. Result and Discussion

A. Result

The results of this study are presented sequentially according to the research procedure.

A.1. Data Collection

Each of the data used already has three classes, namely (i) don't know, (ii) eyewitness, and (iii) non-eyewitness. Table III is an example of flood disaster data used in this study.

Table IV is an example of forest fire disaster data used in research. Forest fire disaster data in this study is also divided into 3 classes, namely (i) don't know, (ii) eyewitness, and (iii) non-eyewitness.

TABLE III. Flood Dataset Examples

Class	Text
Don't Know	But let's just let them flood in with no repercussions and give them tons of aid....? They're breaking the laws! Wh https://t.co/1OW8wUd7mB
Don't Know	Nothing like a good ol flash flood
Don't Know	It was great to meet with business leaders representing GNOinc! We talked about the importance of flood insurance https://t.co/wuEVb13Pu5
Eyewitness	My state, everybody. Drought, fire, flood, hail, and tornados are possible in the same space at the same time. You https://t.co/mxKq9jPrnU
Eyewitness	Wow my alarm rings in 30 minutes. Haha thanks tho I don t wanna wade in flood and leptospirosis today
Eyewitness	Got trapped inside my apartment today due to heavy rain and flooding. Still gonna try and make the most of today tho https://t.co/mEHIG1BpOM
Non-eyewitness	MORNING FLOODING IN BEAUFORT Heavy rain and slow moving storms meant localized flooding in #Beaufort this morning. https://t.co/yv3c2ilnkW
Non-eyewitness	Flood Warning on the West Fork of the San Jacinto River has been canceled as the river is below flood stage. https://t.co/KOAjRcHlcY
Non-eyewitness	Flash flood watch in Pittsburgh again rip

TABLE 1. Forest Fire Dataset Examples

Class	Text
Don't Know	A more pointed question would be why more people *don't* suffer in this spoiled & wicked world #thewallholds #fornow https://t.co/K7Wjl1jyBV
Don't Know	I would gladly take a Ft. McMurray-style wildfire in my town over an HRC establishment president any day of the week
Don't Know	This makes me laugh and then feel bad I laughed every time #Wildfire #TheWalkingDead #BGBRewatch https://t.co/rGFFbQzpEZ
Eyewitness	Damn this Canadian Wildfire got the whole city smelling like a bonfire
Eyewitness	i'm having like allergies or some shit from the smoke from the fire,, thanks for burning up California
Eyewitness	Just in case y'all haven't heard, California is on fire. Even in Oakland we're literally inhaling and driving through clouds of smoke...
Non-eyewitness	(#TeamVillanosFDL) Escaping a wildfire -- in a convoy: At least 88,000 people have been forced to flee the ... https://t.co/oHKr8hziFC
Non-eyewitness	Canadian wildfire spreads, 88,000 people evacuated: A massive wildfire burning out of https://t.co/yJmWyA7iuD
Non-eyewitness	Canadian wildfire increases tenfold, cuts off evacuees https://t.co/w7C8M0j56s

A.2. Preprocessing

This study's preprocessing stages of flood and forest fire disaster data include techniques. The first technique, cleansing is the stage of removing punctuation marks such as: commas (,), periods (.), question marks (?), exclamation marks (!) and so on, symbols, numbers, hashtags, emoticons, usernames, and links. An example of the cleansing process can be seen in Table V.

TABLE V. CLEANSING RESULT

Input	Result
But let's just let them flood in with no repercussions and give them tons of aid....? They're breaking the laws! Wh https://t.co/1OW8wUd7mB	But lets just let them flood in with no repercussions and give them tons of aid They re breaking the laws Wh

Case folding is the stage of changing the form of words into the same form, whether all of them become lowercase (lowercase letters) or become uppercase. In this study, all text is changed to lowercase. An example of a case folding process can be seen inTable VI.

TABLE VI.CASE FOLDING RESULT

Input	Result
But lets just let them flood in with no repercussions and give them tons of aid They re breaking the laws Wh	but lets just let them flood in with no repercussions and give them tons of aid they re breaking the laws wh

Tokenizing is a process in which a collection of words strung together in a sentence will be split into single-word fractions or in the form of tokens. An example of the tokenizing process can be seen in Table VII.

TABLE VII. TOKENIZING RESULT

Input	Result
but lets just let them flood in with no repercussions and give them tons of aid they re breaking the laws wh	[‘but’ ‘lets’ ‘just’ ‘let’ ‘them’ ‘flood’ ‘in’ ‘with’ ‘no’ ‘repercussions’ ‘and’ ‘give’ ‘them’ ‘tons’ ‘of’ ‘aid’ ‘they’ ‘re’ ‘breaking’ ‘the’ ‘laws’ ‘wh’]

Filtering is the process of removing words included in the stopword category or words considered to have no role in the sentiment analysis process. An example of the filtering process can be seen in Table VIII.

TABLE VIII. FILTERING RESULT

Input	Result
[‘but’ ‘lets’ ‘just’ ‘let’ ‘them’ ‘flood’ ‘in’ ‘with’ ‘no’ ‘repercussions’ ‘and’ ‘give’ ‘them’ ‘tons’ ‘of’ ‘aid’ ‘they’ ‘re’ ‘breaking’ ‘the’ ‘laws’ ‘wh’]	['lets', 'let', 'flood', ' repercussions', 'give', 'tons', 'aid', 'breaking', 'laws', 'wh']

Stemming is the process of transforming words that previously had affixes into their basic word forms according to their morphological structure. So it can be assumed that the word that transforms has the same meaning and meaning as the base word. An example of the stemming process can be seen in Table IX.

TABLE IX. STEMMING RESULT

Input	Result
['lets', 'let', 'flood', ' repercussions', 'give', 'tons', 'aid', 'breaking', 'laws', 'wh']	['let', 'let', 'flood', 'repercuss', 'give', 'ton', 'aid', 'break', 'law', 'wh']

A.3. TF-IDF Weighting

TF-IDF (Term Frequency-Inverse Document Frequency) is a method for weighting words in documents. The following is an example of the results from the TF-IDF in Table X.

TABLE X. TF-IDF WEIGHTING RESULT

aid	flood	…	Class
0,303458	0,112194	…	Don’t know
0	0,103101	…	Don’t Know
0	0,18178	…	Don’t Know
0	0,122315	…	Eyewitness
0	0,117747	…	Eyewitness
0	0,178198	…	Eyewitness
0	0,13188	…	Eyewitness
0	0,119439	…	Non-eyewitness
0	0,200469	…	Non-eyewitness
0	0,141167	…	Non-eyewitness

After the TF-IDF word was weighted using 2000 flood disaster data, it produced 4889 features and the resulting features such as "aid", "alert", "flood", "yard", "yay", and so on. In the forest fire disaster data which totaled 2000 data produced 3092 features with the resulting features such as "ab", "abandon", "fire", "zone", and so on.

A.4. Classification

The method used in this research is Multinomial Naïve Bayes as data classification. SMOTE and K-Means SMOTE are also used in this study for data balancing methods. In this study, three classification experiments were carried out.

The first test uses the Multinomial Naïve Bayes classification without balancing the data. In the first test the flood disaster data produced precision, recall and f1 score performance which can be seen in Table XI.

TABLE XI. CLASSIFICATION PERFORMANCE OF FLOOD REPORTS USING MULTINOMIAL NAÏVE BAYES

Precision	Recall	F1-score
69,71%	61,49%	62,37%

In the first test, the forest fire disaster data produced precision, recall and f1 score performance which can be seen in Table XII.

TABLE XII. CLASSIFICATION PERFORMANCE OF FOREST FIRE REPORTS USING MULTINOMIAL NAÏVE BAYES

Precision	Recall	F1-score
83,37%	40,82%	40,91%

In the second test using SMOTE for data balancing and Multinomial Naïve Bayes classification. The second test on the flood disaster data produces precision, recall and f1 score performance, which can be seen in Table XIII.

TABLE XIII. CLASSIFICATION PERFORMANCE OF FLOOD REPORTS USING SMOTE AND MULTINOMIAL NAÏVE BAYES

Precision	Recall	F1-score
68,38%	67,99%	66,66%

The second test on forest fire disaster data produces precision, recall and f1 score performance, which can be seen in Table XIV.

TABLE XIV. CLASSIFICATION PERFORMANCE OF FOREST FIRE REPORTS USING SMOTE AND MULTINOMIAL NAÏVE BAYES

Precision	Recall	F1-score
63,76%	71,14%	65,42%

In the third test using K-Means SMOTE for data balancing and Multinomial Naïve Bayes classification. The third test on the flood disaster data produces precision, recall and f1 score performance, which can be seen in Table XV.

TABLE XV. CLASSIFICATION PERFORMANCE OF FLOOD REPORTS USING K-MEANS SMOTE AND MULTINOMIAL NAÏVE BAYES

Presisi	Recall	F1-score
68,01%	67,44%	66,04%

The third test on forest fire disaster data produces precision, recall and f1 score performance, which can be seen in Table XVI.

TABLE XVI. CLASSIFICATION PERFORMANCE OF FOREST FIRE REPORTS USING K-MEANS SMOTE AND MULTINOMIAL NAÏVE BAYES

Presisi	Recall	F1-score
64,88%	70,70%	66,31%

B. Discussion

In the flood disaster data using the Multinomial Naïve Bayes classification, the f1 score performance increases when using SMOTE and K-Means SMOTE. In testing using SMOTE and K-Means SMOTE produces a different f1 score performance, when using SMOTE to balance data it produces an f1 score performance of 66.66% whereas when using K-Means SMOTE to balance data it produces an f1 score performance of 66.04% so when using K-Means SMOTE decreased performance by a difference of 0.62%. A comparison of the performance of the f1 score of each test on the flood disaster data is presented in a graph that can be seen in Fig 2.

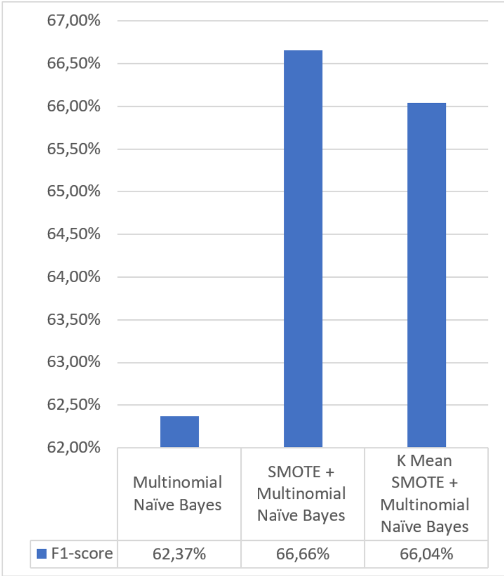

Fig 2. Classification performance comparison of flood reports.

In forest fire disaster data using the Multinomial Naïve Bayes classification, the f1 score performance increases when using SMOTE and K-Means SMOTE. In testing using SMOTE and K-Means SMOTE produces a different f1 score performance, when using SMOTE to balance data it produces an f1 score performance of 65.42% whereas when using K-Means SMOTE to balance data it produces an f1 score performance of 66.31% so when using K-Means SMOTE experienced a performance increase of 0.89%. A comparison of the performance of f1 scores from each test on forest fire disaster data is presented in a graph that can be seen in Fig 3.

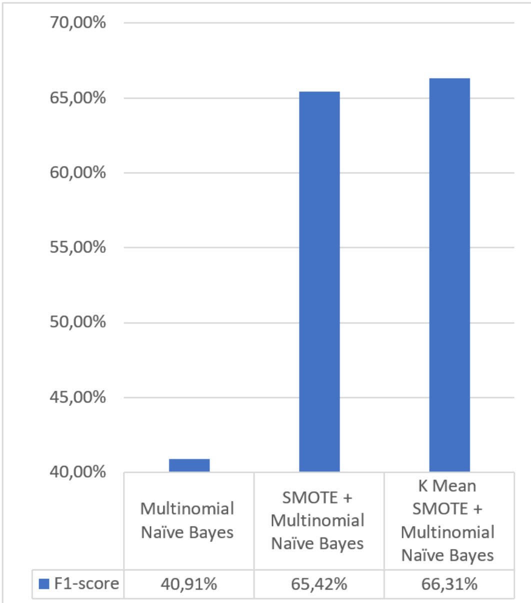

Fig 3. Classification performance comparison of forest fire reports.

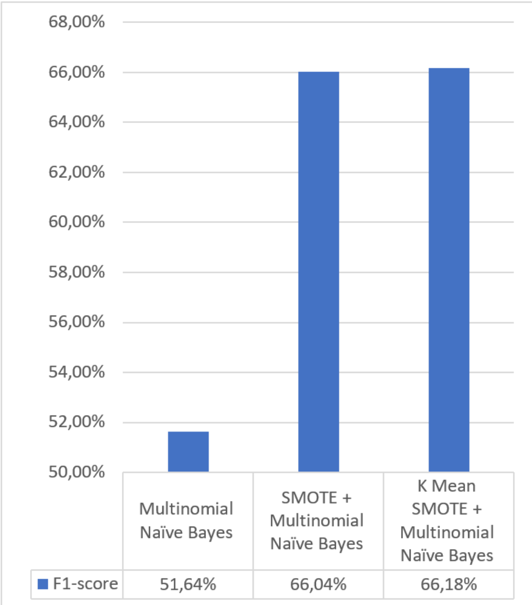

Fig 4. Average Classification Performance of flood and forest fire reports.

The difference in the results of the performance of the f1 score on flood and forest fire disaster data is caused by the different number of majority and minority classes. In the flood disaster data there is a close number of differences between the majority and minority classes where the don't know class is 822, eyewitness 627 and non-eyewitness 551. Meanwhile in the forest fire disaster data there is a large difference in numbers between the majority and minority classes where the don 't know totaled 432, eyewitness 189 and non-eyewitness 1379. So that the differences between classes affect the process of balancing data using SMOTE and K-means SMOTE, as well as the Multinomial Naïve Bayes classification also shows unsatisfactory results for the classification of flood disaster messages and forest fires. The results of the comparison of the average performance of f1 scores from each test on flood and forest fire disaster data are presented in a graph that can be seen in Fig 4.

The classification model produced by Multinomial Naïve Bayes in this study did not produce good performance. This might be due to several reasons. First, N-Gram feature extraction in this study produces high-dimensional data with a total of 4889 features for the flood dataset and 3092 features for the forest fire dataset. In high-dimensional data, features may make up the classification model's performance. Another cause is the feature extraction technique used to produce sparse data. This can also affect the performance of the classification model.

IV. Conclusion and Future Work

This study shows that social media messages can be used to know when a natural disaster occurs by automatically classifying using the naïve Bayes multinomial classification algorithm. Natural disaster messages Disaster reports from eyewitnesses can be much smaller than messages from other categories. This raises the problem of unbalanced data which can affect the classification performance. Classification performance results using the naïve Bayes multinomial algorithm on unbalanced data show low performance, namely 62.37% for the case of classification of flood reports and 49.91% for cases of forest fire reports. After balancing the data with SMOTE and K-Mean SMOTE, there was an increase in classification performance. The influence of the two data balancing methods results in relatively the same average performance of 60.4% for SMOTE and Multinomial Naïve Bayes, and 66.18% for K-Mean SMOTE and Multinomial Naïve Bayes.

In further research, the application of feature extraction, feature selection methods and classification algorithms will be carried out to improve classification performance. The feature extraction method that will be used in future research is a word embedding-based technique that the Multinomial Naïve Bayes classification algorithm will process.

Acknowledgment

We would like to express our highest gratitude to Ministry of Research, Technology and Higher Education Indonesia and Lambung Mangkurat University for supporting this research through Penelitian Kerjasama - Dalam Negeri Research Schema 2023.

References

[1] K. Zahra, M. Imran, and F. O. Ostermann, “Automatic identification of eyewitness messages on twitter during disasters,” Inf. Process. Manag., vol. 57, no. 1, p. 102107, 2020, doi: 10.1016/j.ipm.2019.102107.

[2] A. Andreyestha and Q. N. Azizah, “Analisa Sentimen Kicauan Twitter Tokopedia Dengan Optimalisasi Data Tidak Seimbang Menggunakan Algoritma SMOTE,” Infotek J. Inform. dan Teknol., vol. 5, no. 1, pp. 108–116, 2022, doi: 10.29408/jit.v5i1.4581.

[3] H. Hairani, K. E. Saputro, and S. Fadli, “K-means-SMOTE for handling class imbalance in the classification of diabetes with C4.5, SVM, and naive Bayes,” J. Teknol. dan Sist. Komput., vol. 8, no. 2, pp. 89–93, 2020, doi: 10.14710/jtsiskom.8.2.2020.89-93.

[4] T. T. A. Putri, S. Sriadhi, R. D. Sari, R. Rahmadani, and H. D. Hutahaean, “A comparison of classification algorithms for hate speech detection,” IOP Conf. Ser. Mater. Sci. Eng., vol. 830, no. 3, 2020, doi: 10.1088/1757-899X/830/3/032006.

[5] M. R. Faisal, R. A. Nugroho, R. Ramadhani, F. Abadi, R. Herteno, and T. H. Saragih, “Natural Disaster on Twitter: Role of Feature Extraction Method of Word2Vec and Lexicon Based for Determining Direct Eyewitness,” Trends Sci., vol. 18, no. 23, p. 680, 2021.

[6] J. A. Septian, T. M. Fachrudin, and A. Nugroho, “Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor,” J. Intell. Syst. Comput., vol. 1, no. 1, pp. 43–49, 2019, doi: 10.52985/insyst.v1i1.36.

[7] M. Sindhuja, K. S. Nitin, and K. S. Devi, “Twitter Sentiment Analysis using Enhanced TF-DIF Naive Bayes Classifier Approach,” in 2023 7th International Conference on

Computing Methodologies and Communication (ICCMC), 2023, pp. 547–551.

[8] A. Sabrani, I. G. W. Wedashwara W., and F. Bimantoro, "Multinomial Naïve Bayes untuk Klasifikasi Artikel Online tentang Gempa di Indonesia," J. Teknol. Informasi, Komputer, dan Apl. (JTIKA), vol. 2, no. 1, pp. 89–100, 2020, doi: 10.29303/jtika.v2i1.87.

[9] D. Elreedy, A. F. Atiya, and F. Kamalov, "A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning," Mach. Learn., pp. 1–21, 2023.

[10] R. Azmatul Barro, I. D. Sulvianti, and M. Afendi, "Penerapan Synthetic Minority Oversampling Technique (Smote) Terhadap Data Tidak Seimbang Pada Pembuatan Model Komposisi Jamu," Xplore J. Stat., vol. 1, no. 1, pp. 1–6, 2013.

[11] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE," Inf. Sci. (Ny)., vol. 465, pp. 1–20, 2018.

[12] H. Hairani, "Peningkatan Konerja Metode SVM Menggunakan Metode KNN Imputasi dan K-Means-Smote untuk Klasifikasi Kelulusan Mahasiswa Universitas Bumigora," J. Teknol. Inf. dan Ilmu Komput., vol. 8, no. 4, p. 713, 2021, doi: 10.25126/jtiik.2021843428.

[13] M. Fluorida Fibrianda and A. Bhawiyuga, "Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM)," J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 2, no. 9, pp. 3112–3123, 2018.

[14] T. B. Sasongko, "Komparasi dan analisis kinerja model algoritma SVM dan PSO-SVM," J. Tek. Inform. dan Sist. Inf., vol. 2, no. 2, pp. 244–253, 2016.

[15] D. Gyoten, M. Ohkubo, and Y. Nagata, "Imbalanced data classification procedure based on SMOTE," Total Qual. Sci., vol. 5, no. 2, pp. 64–71, 2020, doi: 10.17929/tqs.5.64.

[16] R. Siringoringo, "KLASIFIKASI DATA TIDAK SEIMBANG MENGGUNAKAN ALGORITMA SMOTE DAN k-NEAREST NEIGHBOR," Isd , vol. 3, no. 1, pp. 2528–5114, 2018.

[17] A. Saifudin, U. Pamulang, R. S. Wahono, U. Dian, and N. Semarang, "Pendekatan Level Data untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software," J. Softw. Eng., vol. 1, no. 2, pp. 76–85, 2015.

[18] G. Tika, A. Adiwijaya, and N. D. W. Cahyani, "Klasifikasi Topik Berita Berbahasa Indonesia Menggunakan Multilayer Perceptron," eProceedings Eng., vol. 6, no. 1, pp. 2137–2143, 2019.