Comparative Analysis using Various Performance Metrics in Imbalanced Data for Multi-class Text Classification

Slamet Riyanto[1], Imas Sukaesih Sitanggang[2], Taufik Djatna[3], Tika Dewi Atikah[4]
Department of Computer Science, IPB University, Bogor, Indonesia[1,2]
Department of Agroindustrial Technology, IPB University, Bogor, Indonesia[3]
Research Centre for Ecology and Ethnobiology, National Research and Innovation Agency[4]
Research Center for Data and Information Sciences, National Research and Innovation Agency[1]

***Abstract*—Precision, Recall, and F1-score are metrics that are often used to evaluate model performance. Precision and Recall are very important to consider when the data is balanced, but in the case of unbalanced data the F1-score is the most important metric. To find out the importance of these metrics, a comparative analysis is needed in order to determine which metric is appropriate for the data being analyzed. This study aims to perform a comparative analysis of various evaluation metrics on unbalanced data in multi-class text classification. This study uses an unbalanced multi-class text dataset including: association, negative, cause of disease, and treatment of disease. This study involves five classifiers as algorithm-level approach, namely: Multinomial Naive Bayes, K-Nearest Neigbors, Support Vector Machine, Random Forest, and Long Short-Term Memory. Meanwhile, data-level approach, this study involves under sampling, over sampling, and synthetic minority oversampling technique. Several evaluation metrics used to evaluate model performance include Precision, Recall, and F1-score. The results show that the most suitable evaluation metric for use on unbalanced data depends on the purpose of use and the desired priority, including the classifier that is suitable for handling multi-class assignments on unbalanced data. The results of this study can assist practitioners in selecting evaluation metrics that are in accordance with the goals and application needs of multi-class text classification.**

***Keywords*—Imbalanced data; undersampling; oversampling; smote; machine learning**

I. Introduction

An unbalanced dataset is a dataset in which some classes have much fewer data samples than others [2]. Imbalanced data occurs when the number of observations in one class is lower than in another class. It can be a problem in machine learning because models may be more accustomed to majority classes, leading to poor performance in minority classes. One of the ways to tackle imbalanced data is to do data sampling before applying machine learning algorithms. Nowaday, common methods of imbalanced data sampling mainly include data oversampling, data undersampling, and hybrid sampling [2], [3].

The method used to address unbalanced data depends on the characteristics of the data. Several previous studies have implemented undersampling [14], [17], [22], [30] to reduce the size of large sample data to balance different types of sample data. Beside undersampling, previous studies have also implemented an oversampling [11], [12], [13], [27] method that takes small samples as the object to generate new samples. Imbalanced data in text classification with multi-class need to be considered since a classification model that is usually based on a fair class distribution could have problems with imbalanced class [6].

The author in [37] has carried out research on comparative analysis of macro and micro accuracy through a three-classifier approach, namely: Naïve Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF) in the Movie Reviews dataset. In align previous research, this study proposes two additional classifiers, namely: k-Nearest Neighbors (KNN) and Long Short-Term Memory (LSTM), which will be tested on the Plant-Disease Relation (PDR) dataset. The main reason for using KNN and LSTM is that these algorithms are also proven to be used to solve unbalanced class problems like what was done by [38], [39], [40], [41]. Furthermore, reference [33] does not work on the KNN and LSTM algorithms, who used Linear SVC, RBF SVM, DTC, RF, LR, and MNB for multi-class text classification tasks.

In comparative analysis, it is necessary to evaluate the model using various performance metrics [5], and it is an interdisciplinary method that encompasses broad cross-sections of disciplines [1]. Comparative analysis is one way to solve the problem of model performance in classifying unbalanced datasets. It involves comparing the performance of different models on the imbalanced datasets, to determine which model is best suited for the task at hand. By comparing the performance of different models, it can identify the strengths and weaknesses of each model and choose a significant model that fits the problem of unbalanced data [4].

The results of the model evaluation usually use the confusion matrix as a model performance metric that is tested on each data. Metrics most commonly used to measure model performance include: Accuracy, Precision, Recall, Specificity, F1-score, and G-mean [7]. In text classification with multiple classes that experience data imbalance, micro and macro accuracy considerations can be useful in evaluating the model. Micro accuracy measures overall model accuracy by giving equal weight to all classes. It can be calculated by adding up the number of correct predictions from all classes and dividing by the total number of predictions [9]. Each algorithm used will produce a different confusion matrix, this is caused

by differences in methods for handling unbalanced data and classifier algorithms used. Several previous studies analyzed the balance of macro-F1 and micro-F1 [5], [8] only. In the case of imbalanced data, micro accuracy F1 may be higher than macro accuracy because the model may more easily predict the more dominant class with a higher level of accuracy [10].

Methods for dealing with class imbalance in machine learning can be divided into three groups, namely: data-level, algorithm-level, and hybrid approaches [11], [17]. Data-level methods aim to reduce class imbalance by adding new minority samples (i.e., oversampling) [13], removing redundant majority samples (i.e., undersampling) [14], or using a combination of both methods. Algorithm-level methods are designed to adapt standard classification methods to emphasize learning from minority samples, improve the training mechanism or predicted rule [11]. In hybrid approaches, the developed algorithms modify both the distribution of unbalanced classes and the learning mechanism to classify unbalanced data [15].

The contribution of this research to handle unbalanced classes are 1) data-level approach, including undersampling, oversampling, and synthetic minority oversampling technique, for tackle imbalanced class; 2) prepare five machine learning models, including NB, RF, SVM, KNN, and LSTM, for multi-class classification on Plant-Disease Relation datasets; 3) investigated and compared the performance of different machine learning models with various feature combinations and existing techniques.

This study employ both data-level and algorithm approach to handling highly imbalanced data aim to perform precision, recall and micro accuracy F1 score. Various test schemes employed through classification algorithm to obtain perspective analysis. The rest of this paper is organized as follows. Section II contains discussion on the dataset and the methodology used to undertake the research. This is followed by the results and discussion in Section III. Finally, in Section IV, the conclusion and future works are presented.

II. Materials and Method

A. Dataset

This study uses gold standard corpus dataset of the Plant-Disease Relation (PDR) developed by [16] available at http://gcancer.org/pdr (21^{st} January, 2022). The dataset consists of 8 columns, but this study only uses the "sentence" column as text data and the "relation" column as a label (Table I). This study only relies on a statistical analysis of a collection of texts converted into a set of numbers, thus ignoring semantic analysis for classification work.

The PDR dataset has four classes: Association (34 records), Cause of Disease (183 records), Treatment of Disease (507 records), and Negative (583 records). Table II shows samples of the PDR dataset.

B. Pre-processing

The sentence column used as input still contains several meaningless words, including $< e1start >$, $< e1end >$, $< e2start >$, and $< e2end >$, so it needs to be clean so that it doesn't have a biased impact when building the model. In this study, retaining words that are considered unimportant (a, an, cant́, have not, etc.) are usually called stop_words. We assume that these words will reduce the models performance in predicting classes.

TABLE I. Sample of Plant-Disease Relation Dataset

Sentence	Plant	Disease	Relation	Trigger
Studies on magnesiumś mechanism of action in $< e1start >$ digitalis $< e1end >$-induced $< e2start >$ arrhythmias $< e2end >$.	digitalis	arrhythmias	CoD	o

TABLE II. View of Plant-Disease Relation Dataset

No.	Sentence	Class
1	Studies on magnesiumś mechanism of action in $< e1start >$ digitalis $< e1end >$-induced $< e2start >$ arrhythmias $< e2end >$.	Cause_of_disease
2	Inhibitory effect of $< e1start >$ green tea $< e1end >$ on the growth of established $< e2start >$ skin papillomas $< e2end >$ in mice.	Treatment_of_disease
3	In 10 separate experiments, mice with established chemically induced or UV light-induced $< e2start >$ skin papillomas $< e2end >$ were treated continuously with green tea in the drinking water or with i.p. injections of a $< e1start >$ green tea $< e1end >$ polyphenol fraction or -epigallocatechin gallate three times a week for 4-10 weeks.	Negative
4	Although based on small numbers of end points, a prospective study has suggested a particularly strong association between recent $< e1start >$ coffee $< e1end >$ drinking and the incidence of $< e2start >$ cardiovascular disease $< e2end >$.	Association

The next step is to convert text into numbers, as the computer cannot process data in the text as the machines only recognize numbers. Therefore, the text shall be used as input and should be convert to numbers. Some methods can do this including Count Vectorizer, TF-IDF, Word2Vec, BERT, and ELMO, where the text will be encoded into a vector space with a fixed length. This study uses the Count Vectorizer approach for the vectorization process. When extracting and representing features from text data, this study also pays attention to n-grams. This study use CountVectorizer from sklearn as vectorizer and use n gram for character embedding algorithm (Fig. 1).

C. Data Training and Testing

This study divides the data into two groups with a ratio of 80:20 to test the reliability of all classifiers to be used, 80% for training data and 20% for test data. In addition to this approach, this study also uses a 5-fold cross validation approach. To divide the data into training and testing, this study also considers the random sample aspect when conducting training, which consists of under sampling, over sampling and synthetic sampling.

D. Class Imbalanced Learning

This study focuses on combine both data-level and algorithm-level approach techniques to measure performance of model through precision, recall and F1 score metrics. Various imbalanced learning techniques have been envolved in addressing the class imbalance problem. It requires either (1) reducing the bias a machine learning algorithm can impart

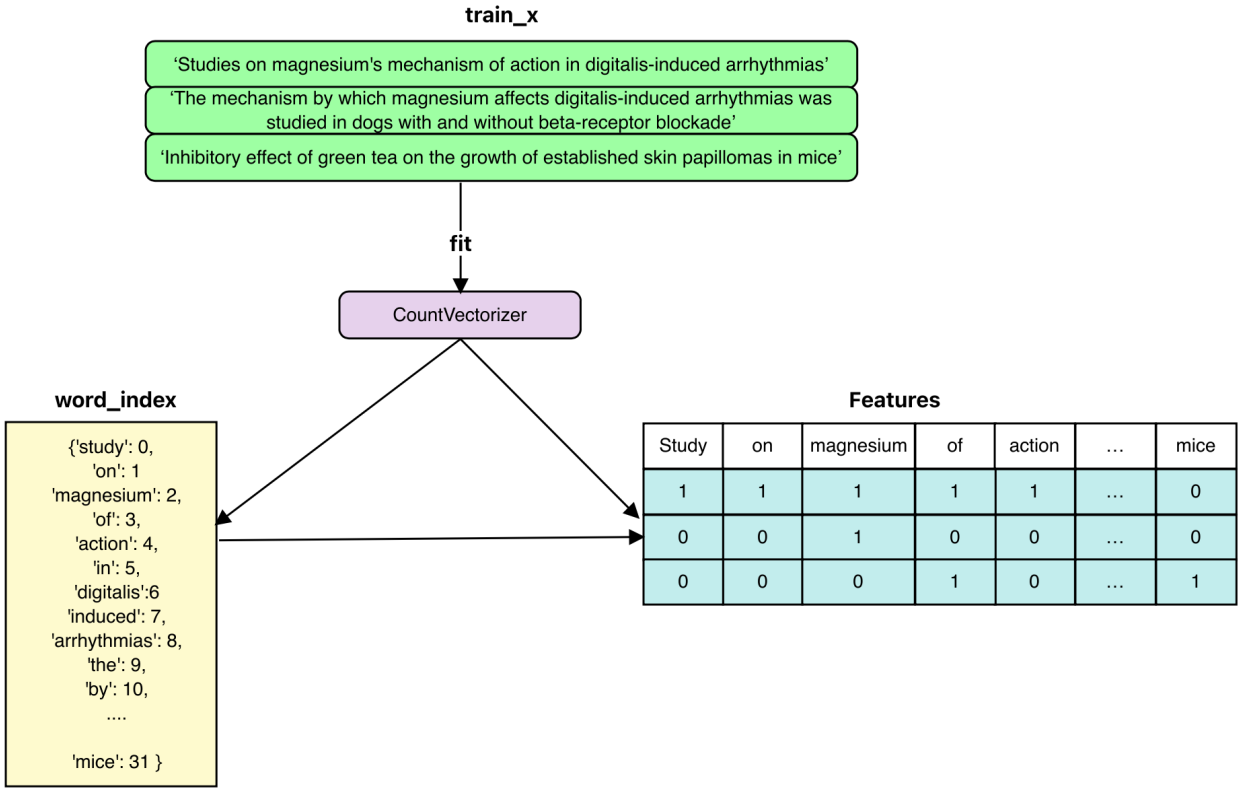

Fig. 1. Flowchart of extracting features.

to the majority class in the dataset, or (2) configuring the algorithm to be sensitive to the minority class [17]. Fig. 2 is a techniques to tackle class imbalance learning that can be classified into three main categories: (a) Data-level methods, (b) Algorithm-level methods, and (c) combination of both methods [11], [17].

In a data-level approach, researchers attempted to balance the dataset before applying traditional classification algorithms so that the majority class did not bias the results. In the algorithm-level approach, researchers worked on the internals of the algorithm and tried to remove the algorithm's sensitivity to the majority class, so that the results of the classification algorithm would not drift towards the majority class [12]. A third approach is a hybrid that combines data-level and algorithm-level approaches [18].

E. Classifier

The classifier is a machine learning model that is used to classify sample data into predetermined classes. The classifier receives input in the form of data samples and outputs in the form of classes of appropriate method for the data samples. The classifiers can be used for a various variety of applications, such as facial recognition, speech recognition, and natural language understanding.

Classifiers can be divided into two main types, namely binary classifiers and multi-class classifiers. A binary classifier is a classifier that can only classify data samples into two classes, while a multi-class classifier is a classifier that can classify data samples into more than two classes. Classifiers can be created using a variety of machine-learning algorithms.

Naïve Bayes Classifier, K-Nearest Neighbor, Support Vector Machine, and Random Forest are the most classifiers used in machine learning. This study uses the classifier referring to previous research [19] and adding LSTM classifier for handling multi-class classifier tasks to obtain optimal results through comparative analysis on Precision, Recall, and F1-score metrics.

1) Multinomial Naïve Bayes: A Multinomial Naïve Bayes (MNB) model is used to represent calculate or count rates. The additive smoothing parameter α is set to 1. Prior class probabilities are learned and adjusted according to the data [20], [21]. The purpose of this method is to classify probability based on supervised machine learning over other probabilities. This study use Multinomial Naïve Bayes as a classifier to classify a document into four classes.

Multinomial Naïve Bayes computes class probabilities for a given document. A collection of classes is denoted by *C*, *N* is the vocabulary size. Next step, MNB assigns a test document *t* to the class that has the highest probability $Pr(c \mid t_i)$ which, using Bayes rule (Equation 1) [21]. The class prior probability $Pr(c)$ can be estimated by dividing the number of documents belonging to class c by the total number of documents. $Pr(t_i \mid c)$ is the probability of obtaining a document like t_i in class c.

$$Pr(c|t_i) = \frac{Pr(c)Pr(c|t_i)}{Pr(t_i)}, c \in C \quad (1)$$

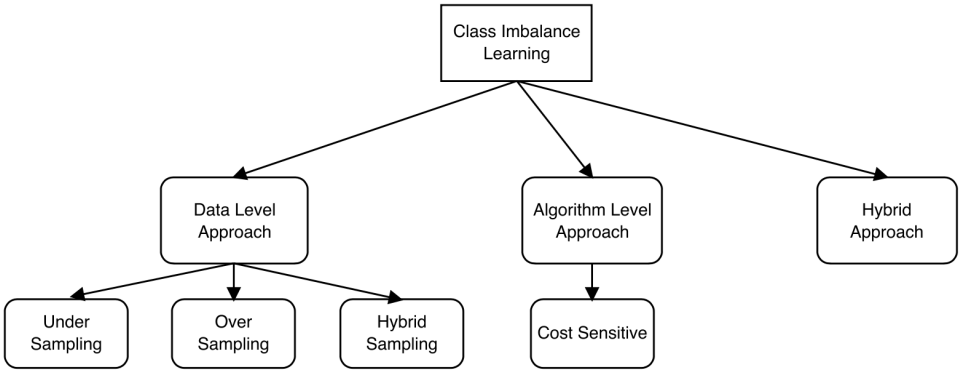

Fig. 2. Hierarchy of class imbalance learning techniques.

2) K-Nearest Neighbours: K-Nearest Neighbours (KNN) is a supervised classification algorithm and is considered one of the best data mining algorithms despite its simplicity. It creates a decision surface that adapts to the shape of the data distribution, resulting in high accuracy when the training set is large or representative [22]. The KNN algorithm assumes that similar things are close together. This means that similar items are placed close together determine best k-Nearest Neighbors using Grid Search.

The classification algorithm follows these steps: (1) compute the distance between a x_i instance and all instances of the training set T, (2) choose the k nearest neighbors, (3) The x_i instance is classified (labeled) with the most oftentimes class among the k nearest neighbors. It is also possible to use the neighborsdistance to weight the classification decision. Characteristically in the literature are found odd values for k, normally with $k = 5$ or $k = 7$ [23]. An approach to determine k as a function (2) of data size m.

3) Support Vector Machine: Support Vector Machine (SVM) works by constructing hyperplanes in a multidimensional space that separates the different cases of class labels. SVM has two important parameters, namely c and γ [19]. Parameter c adds a penalty for each misclassified data point. If c is small, the penalty for misclassified points is low, so decision boundaries with large margins are chosen at the cost of more misclassifications. The gamma parameter controls the influence distance of training points. When small gammas, c affects the model in the same way, it affects linear models. Typical values for c and gamma are $0.0001 < \gamma < 10$ and $0.1 < c < 100$. This study uses hyperparameter tuning for a grid search to determine the optimal γ and c values.

Given a training set of N data points $\{y_k, X_k\}N_k = 1$, where $X_k \in \mathbb{R}^n$ is the kth input pattern and yk $\in \mathbb{R}$ is the kth output pattern, the support vector method approach aims to building a classifier of the form (Equation 2) [24]:

$$y(x) = sign\left[\sum_{i=0}^{N} \alpha_k y_k \Psi(x, x_k) + b\right] \tag{2}$$

where α_k are positive real constants and b is real constant, $\Psi(x, x_k) = x_k^T$ x for linear SVM. In the above expression, $\Psi(x, x_k)$ α_k, y_k, x_k, b, and N represent a kernel function [25].

4) Random Forest: The Random Forest (RF) algorithm is a machine learning technique that can be employed to classify text into multiple classes. The Random Forest algorithm generates numerous decision trees that employ these features to predict the class of text samples, making it a more effective algorithm for datasets with many features than other machine learning techniques. The ensemble tree-based RF classifier chooses features from the training data randomly, and it reduces the correlation between trees [3]. This study utilized value of n_estimators= 100 as input into the RF model. The value of n_estimators get from a grid search approach for hyperparameter tuning. A grid search was applied to select the optimal n_estimators used to classify on multi-class dataset. These parameters are applied independently and interactively, with samples randomly chosen from the training dataset to arrive at a final prediction. This study aligns with previous research by [31] that used RF as a classifier to handle unbalanced classes.

5) Long Short-Term Memory: In the past decade, there has been a surge in the use of deep learning techniques, which have become increasingly popular due to their ability to enhance the state-of-the-art in fields such as speech recognition and computer vision, among others [26]. In this study, the classifier was trained using the Long Short-Term Memory (LSTM) algorithm. The training dataset consists of a certain number of time series vectors, which are not specified in the given text $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_N$ where $\mathbb{X}_k$ with k = 1, 2, … N reflect the trajectory sequence with mathematically integrated from kth sentence and corresponding labels y_1, y_2, … y_N. Equation 3, 4, 5, and 6 are computation stages for single LSTM unit.

$$Z_k = tanh(\, W.[\, X_k^t; h_k^{t-1}; 1]\,) \tag{3}$$

$$Z_k^i = \delta(\, W^i.[\, X_k^t; h_k^{t-1}; 1]\,) \tag{4}$$

$$Z_k^f = \delta(\, W^f.[\, X_k^t; h_k^{t-1}; 1]\,) \tag{5}$$

$$Z_k^o = \delta(\, W^o.[\, X_k^t; h_k^{t-1}; 1]\,) \tag{6}$$

where tanh stands for tanh function, δ represents sigmoid function, W, W^i, W^f, W^o are row vectors which stand for the weight combined with bias parameters for LSTM cell, input gate, forgate gate, and output gate, respectively. Z_k represents the output of the output of the LSTM cell for kth sequence Z_k^i, Z_k^f, Z_k^o represent the scalar outputs of input gate, forget gate, and output gate sequence, respectively. $[X_k^t; h_k^{t-1}; 1]$ is a column vector combined by column vector X_k^t,h_k^{t-1} and 1 [27], [32].

6) Class Balancing Techniques: The class balance technique is a way to overcome the problem of imbalance class in a dataset. Class imbalance occurs when a class has a much higher number of samples than the others. This event can pose problems for machine learning, as models tend to predict which classes are more common than others. One way to balance classes is to use a sampling technique. Sampling is a technique used to take samples from a larger data set and use them to create a smaller, balanced data set.

One technique for tackling this problem is to employing multiple sampling procedures, which are classified as random and special. In the first situation, remove a fixed number of examples from the majority class (undersampling) as shown in Fig. 3; in the second, increase the number of minority class examples (oversampling) [28]. In this study, use oversampling and undersampling (Fig. 3). All sampling will be applied to all classifiers to obtain the optimal mode in carrying out multi-class classification work on imbalanced data.

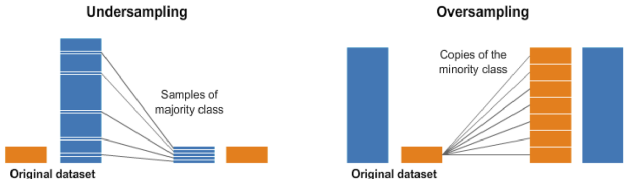

Fig. 3. Undersampling and oversampling balancing classes algorithms.

F. Text Classification Model

This study applied five different machine learning algorithms when performing multiclass text classification of plant-disease relation dataset: Multinomial Naïve Bayes, K-Nearest Neighbours, Support Vector Machine, Random Forest, and LSTM. The `scikit-learn` machine learning library running in the Python and programming system was used to implement these classifiers. Fig. 4 shows the steps in creating the classification model.

Sklearn.Naïve_bayes.MultinomialNB is used to implement a Multinomial Naïve Bayes classifier, with parameter $\alpha = 1$. On the other hand, sklearn.neighbors.KNeighborsClassifier is used to perform K-Nearest Neighbours classifier, with parameter metric=manhattan, n_neighbors=65, p=1, and weights=distance. Sklearn.svm.SVC was used to implement Linear SVC. For this classifier, the kernel and c parameter were chosen to be linear and 1, respectively. For kernel RBF, the SVM classifier was also run from sklearn.svm.SVC with parameters c=1, gamma=0.1, and kernel=linear. A Random Forest classifier was implemented based on the

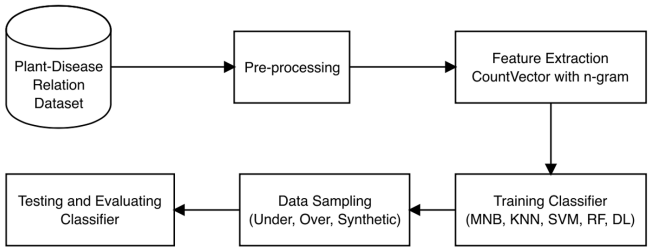

Fig. 4. Flowchart of methodology in this study.

sklearn.ensemble.RandomForestClassifier class in this case the parameters n_estimators = 100 and max features = 'auto'.

The hyperparameters of these classifiers were determined based on using a Grid Search algorithm. Based on these hyperparameters, the classifier provided the highest accuracy. Hyperparameters were determined using 5-fold cross-validation. The hyperparameter names and values for each classifier that correspond to the top accuracy values (Table III).

TABLE III. HYPERPARAMETER TUNING

Classifier	Feature selection method	Parameters for tuning	Best value
MNB	Character based unigram, bigram with CountVector	alpha=[0.001, 0.1, 1, 10, 100, 1000]	alpha=1
KNN	Character based unigram, bigram with CountVector	n_neighbors=[1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100] metric=[euclidean, manhattan, minkowski] p=[1, 2] weights=[uniform, distance]	n_neigbors=65 metrics=manhattan p=1 weights=distance
SVM	Character based unigram, bigram with CountVector	c=[0.1, 1, 10] gamma=[0.1, 1, 10] kernel=[linear, poly, rbf]	c=1 gamma=0.1 kernel=linier
RF	Character based unigram, bigram with CountVector	n_estimators= [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000]	n_estimators=100
LSTM	Character based unigram, bigram with CountVector	num_units=[16, 32, 64, 128] dropout=[0.2, 0.3, 0.5, 0.8] epochs=[10, 20, 30, 40] batch_size=[16, 32, 64, 128]	num_units=128 dropout=0.3 epochs=40 batch_size=64

III. RESULT AND DISCUSSION

The pre-processing stage is very influential on the result of model. This study processes data only through tokenization without involving stop words. The reason for not using the stop_word is some words that are considered meaningless play an important role in determining class. After going through the tokenization process, data increased to 25136. Apart from the pre-processing stage, each classifierś also influence the modelś result. Multinomial Naïve Bayes has parameters alpha and c, K-Nearest Neighbors has parameters n_neighbors, Random Forest has n_estimators, and Deep Learning has parameters dropout, activation, and optimizer. Table IV shows model

evaluation in this study which involve eight scenarios. This study evaluates models from different scenarios based on precision, recall, and f1-score.

TABLE IV. MODEL DEVELOPMENT SCENARIOS

No.	Scenario	Description
1	Ratio (80:20)	Training (80) and Testing (20)
2	5-fold cross validation	dividing training and testing data which divides equally into five parts and without considering imbalanced class
3	Under-sampling + Ratio	Ratio with due regard to imbalanced learning based on under-sampling
4	Under-sampling + 5-fold cross validation	5-fold cross validation with paying attention to imbalanced learning based on under-sampling
5	Over-sampling + Ratio	Ratio taking into account imbalance learning based on oversampling
6	Oversampling + 5-fold cross validation	5-fold cross validation taking into account imbalanced learning based on over-sampling
7	SMOTE + Ratio	Ratio taking into account imbalanced learning based on SMOTE
8	SMOTE + 5-fold cross validation	5-fold cross validation with imbalanced learning based on SMOTE

Precision is a metric for evaluating machine learning models that measures the accuracy of the models recognition of true positives out of the total positive predictions [29] (Equation 7).

$$Precision_{\mu} = \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i + fp_i)} \tag{7}$$

where $\sum_{i=1}^{l} tp_i$ is amount of true positive in whole class, while $\sum_{i=1}^{l}(tp_i + fp_i)$ is total summation of true positive and false positive.

In the context of the confusion matrix, recall is a metric used to evaluate the accuracy of a classification model in correctly identifying true positives out of the total number of positive classes in the actual data [29]. This metric provides information about the proportion of correct positive predictions and can be used to assess the quality of a classification model in minimizing the number of false negative predictions (Equation 8).

$$Recall_{\mu} = \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l}(tp_i + fn_i)} \tag{8}$$

The micro accuracy metric involves adding the correct prediction count for each class and dividing it by the total number of samples. This method assigns equal value to every sample when determining accuracy, making it an appropriate measure to use when the dataset is imbalanced, or when the number of samples for each class varies. μ represent micro averaging (Equation 9).

$$F1score_{\mu} = \frac{(\beta^2 + 1)\, Precision\mu.Recall\mu}{\beta^2 Precision\mu + Recall\mu} \tag{9}$$

The macro accuracy metric is determined by initially computing the accuracy for each class and then calculating the average accuracy of all the classes. This method assigns equal value to every class when determining accuracy, making it an appropriate measure to use when the dataset is balanced, or when the number of samples for each class is more or less equal (Equation 10).

$$F1score_{M} = \frac{(\beta^2 + 1)\, Precision_{M}.Recall_{M}}{\beta^2 Precision_{M} + Recall_{M}} \tag{10}$$

A. Multinomial Naïve Bayes

The result shows that the model works very efficiently on the ratio and 5-fold cross-validation schemes. However, this does not give an overall picture of the model's performance. The performance of the model can be affected by imbalanced data, so the model tends to correctly predict the majority class and ignore the minority class. Therefore, it is necessary to perform several techniques to balance classes such as oversampling, under-sampling, or synthetic minority oversampling techniques (SMOTE). The weighted average on Precision and Recall shows the overall model performance considering the frequency of each class with a score of 0.92, respectively.

Based on the scores obtained from all the schemes used, it can be concluded that MNB has the best performance on Ratio scheme. These results are in line with a study conducted by [20] which obtained the highest score compared to SVM, RF and KNN. The feature space used is unigram + bigram at 80% training data and 20% testing data (Table V). This achievement is optimal on unbalanced data. However, testing on balanced data through undersampling and oversampling techniques gives a low score.

The highest score is obtained through the oversampling + ratio scheme. The highest score was obtained through the oversampling + ratio scheme with a score of 0.76, while the SMOTE scheme obtained a score of 0.75. These results contrast with research conducted by [35] and [36], which obtained the highest score when using the SMOTE scheme.

TABLE V. MULTINOMIAL NAÏVE BAYES CLASSIFIER RESULTS

	Class	R	F	U+R	U+F	O+R	O+F	S+R	S+F
Precision	Assoc	0.83	0.84	0.33	0.27	0.62	0.53	0.50	0.54
	CoD	0.67	0.82	0.54	0.46	0.63	0.58	0.65	0.58
	Neg	0.77	0.94	0.59	0.70	0.83	0.79	0.83	0.78
	ToD	0.89	0.94	0.72	0.70	0.76	0.82	0.75	0.82
	macro avg	0.79	0.88	0.55	0.53	0.71	0.68	0.68	0.68
	weight avg	**0.80**	**0.92**	0.63	0.65	**0.77**	0.76	0.76	0.76
Recall	Assoc	0.71	0.76	1.00	1.00	0.71	0.68	0.71	0.62
	CoD	0.57	0.95	0.84	0.72	0.81	0.77	0.81	0.75
	Neg	0.82	0.87	0.49	0.42	0.63	0.65	0.61	0.67
	ToD	0.89	0.97	0.59	0.72	0.89	0.87	0.89	0.86
	macro avg	0.75	0.88	0.73	0.71	0.76	0.74	0.76	0.72
	weight avg	**0.80**	**0.92**	0.60	0.60	**0.76**	0.75	0.75	0.75
F1-score	Assoc	0.77	0.80	0.50	0.42	0.67	0.60	0.59	0.58
	CoD	0.62	0.88	0.66	0.57	0.71	0.66	0.72	0.65
	Neg	0.79	0.90	0.53	0.52	0.72	0.71	0.71	0.72
	ToD	0.89	0.95	0.65	0.71	0.82	0.84	0.82	0.84
	accuracy	**0.80**	**0.92**	0.60	0.60	**0.76**	0.75	0.75	0.75
	macro avg	0.77	0.88	0.59	0.55	0.73	0.70	0.71	0.70
	weight avg	0.80	0.91	0.60	0.60	0.75	0.75	0.74	0.75

- R=Ratio 80:20, F=5-fold cross validation, U=under sampling, O=over sampling, S=SMOTE, Assoc=association, CoD=cause of disease, Neg=negative, ToD=treatment of disease.

B. K-Nearest Neighbors

K-Nearest Neighbors (KNN) is an uncomplicated and widely used classification algorithm that predicts the category

of a sample based on the category of the closest k-neighbors in the feature space. KNN operates by comparing the distance between the samples being predicted with the categories of other samples already present in the dataset. The way to determine the number of nearest neighbors is through hyperparameter tuning and the values n_neigbors=65, metrics=manhattan, p=1, and weights=distance are obtained.

The hyperparameter results the optimal value n_neigbors=65, metrics=manhattan, p=1, and weights= distance. The results of modeling using the KNN classifier as shown in Table VI that the data-level approach based on under sampling with the ratio had optimal performance in Precision, Recall and F1-sore are 0.94 respectively. This achievement outperforms other schemes with balanced data. These results indicate that unbalanced data can be overcome by under-sampling like as previous research [22] and an 80:20 ratio approach for separating training data and test data. On the other hand, the study conducted by [20] obtained the highest score on the Recall metric when using KNN, this outperformed other classifiers such as RF, SVM and MNB.

TABLE VI. K-NEAREST NEIGHBOR CLASSIFIER RESULTS

	Class	R	F	U+R	U+F	O+R	O+F	S+R	S+F
Precision	Assoc	0.84	1.00	0.84	0.62	1.00	1.00	1.00	0.21
	CoD	0.89	0.66	0.89	0.33	0.66	0.62	0.62	0.35
	Neg	0.96	0.60	0.96	0.51	0.70	0.64	0.64	0.79
	ToD	0.95	0.79	0.95	0.62	0.71	0.76	0.76	0.82
	macro avg	0.91	0.76	0.91	0.52	0.77	0.75	0.75	0.54
	weight avg	**0.94**	0.69	**0.94**	0.52	0.71	0.69	0.69	0.72
Recall	Assoc	0.94	0.62	0.94	0.71	0.57	0.57	0.57	0.71
	CoD	0.93	0.56	0.93	0.60	0.64	0.50	0.50	0.69
	Neg	0.91	0.75	0.91	0.59	0.63	0.74	0.74	0.52
	ToD	0.98	0.65	0.98	0.28	0.83	0.69	0.69	0.67
	macro avg	0.94	0.65	0.94	0.54	0.67	0.63	0.63	0.65
	weight avg	**0.94**	0.68	**0.94**	0.48	0.70	0.68	0.68	0.61
F1-score	Assoc	0.89	0.77	0.89	0.67	0.73	0.73	0.73	0.32
	CoD	0.91	0.61	0.91	0.43	0.65	0.55	0.55	0.46
	Neg	0.93	0.67	0.93	0.55	0.66	0.68	0.68	0.63
	ToD	0.97	0.71	0.97	0.38	0.76	0.72	0.72	0.74
	accuracy	**0.94**	0.68	**0.94**	0.48	0.70	0.68	0.68	0.61
	macro avg	0.92	0.69	0.92	0.51	0.70	0.67	0.67	0.54
	weight avg	0.94	0.68	0.94	0.47	0.70	0.68	0.68	0.63

- R=Ratio 80:20, F=5-fold cross validation, U=under sampling, O=over sampling, S=SMOTE, Assoc=association, CoD=cause of disease, Neg=negative, ToD=treatment of disease.

C. Support Vector Machine

The SVM classifier performed best using oversampling and ratio schemes, based on the test results. All the evaluated metrics, achieved a score of 0.91, even when using the weighted average for unbalanced classes. The model's prediction results for all classes were evenly scored, indicating that the oversampling approach effectively addressed the problem of imbalanced data. Although the SMOTE approach was not as effective as oversampling, it still outperformed all other data-level schemes (Table VII).

As seen from the Table VIII, the highest accuracy of 0.91% was obtained when using scheme ratio using oversampling. This achievement is influenced by character level settings such as research conducted by [2]. In the research, the fourgram level character greatly influences SVM performance which is superior compared to using unigrams, bigrams, and trigrams than MNB and RF.

TABLE VII. SUPPORT VECTOR MACHINE CLASSIFIER RESULTS

	Class	R	F	U+R	U+F	O+R	O+F	S+R	S+F
Precision	Assoc	0.67	0.68	0.83	0.17	1.00	0.62	0.71	0.29
	CoD	0.75	0.70	0.88	0.57	0.88	0.67	0.85	0.58
	Neg	0.79	0.76	0.60	0.67	0.88	0.81	0.86	0.89
	ToD	0.88	0.82	0.44	0.67	0.88	0.86	0.85	0.92
	macro avg	0.77	0.74	0.69	0.52	0.91	0.74	0.82	0.65
	weight avg	0.81	0.77	0.71	0.64	**0.91**	0.80	0.82	0.80
Recall	Assoc	0.57	0.56	0.83	0.71	1.00	0.71	0.99	0.71
	CoD	0.64	0.62	0.70	0.62	0.96	0.67	0.81	0.76
	Neg	0.82	0.74	0.50	0.50	0.77	0.77	0.71	0.71
	ToD	0.90	0.88	0.67	0.70	0.93	0.89	0.80	0.84
	macro avg	0.73	0.70	0.67	0.63	0.91	0.76	0.83	0.76
	weight avg	0.81	0.78	0.68	0.60	**0.91**	0.80	0.81	0.77
F1-score	Assoc	0.62	0.61	0.83	0.27	1.00	0.67	0.83	0.42
	CoD	0.69	0.66	0.78	0.59	0.92	0.67	0.83	0.66
	Neg	0.80	0.75	0.55	0.57	0.82	0.79	0.78	0.76
	ToD	0.89	0.85	0.53	0.69	0.90	0.88	0.82	0.88
	accuracy	0.81	0.78	0.68	0.60	**0.91**	0.80	0.81	0.77
	macro avg	0.75	0.72	0.67	0.53	0.91	0.75	0.81	0.68
	weight avg	0.81	0.77	0.69	0.61	0.90	0.80	0.81	0.78

- R=Ratio 80:20, F=5-fold cross validation, U=under sampling, O=over sampling, S=SMOTE, Assoc=association, CoD=cause of disease, Neg=negative, ToD=treatment of disease.

D. Random Forest

Table VIII shows that the Random Forest classifier has good performance on imbalanced data through a 5-fold cross-validation scheme with an even score of 0.92 on Precision, Recall, and F1-score, respectively. On the other hand, this classifier can handle imbalanced data through a data-level over-sampling scheme. The difference between the macro average and the weighted average on Precision, Recall, and F1-score is only 1-2 points. It's shows that the model can work on balanced or unbalanced classes. Macro average gives the same weight to each class, regardless of the frequency of each class. Meanwhile, the weighted average gives different weights to each class depending on the frequency of each class.

Based on testing, this classifier produces an optimal model through a 5-fold cross validation scheme without sampling. If this classifier uses sampling (under and over), the model has poor performance. This is in line with research [34] which has found that the implementation of SMOTE to Random Forests has an impact on reducing model performance.

TABLE VIII. RANDOM FOREST CLASSIFIER RESULTS

	Class	R	F	U+R	U+F	O+R	O+F	S+R	S+F
Precision	Assoc	1.00	0.84	0.60	0.74	0.83	0.79	0.38	0.79
	CoD	0.75	0.82	0.57	0.75	0.71	0.73	0.58	0.73
	Neg	0.78	0.94	0.78	0.76	0.86	0.77	0.87	0.77
	ToD	0.78	0.94	0.53	0.78	0.76	0.80	0.84	0.80
	macro avg	0.82	0.88	0.62	0.76	0.79	0.77	0.67	0.77
	weight avg	0.77	**0.92**	0.65	0.77	0.80	0.78	0.80	0.78
Recall	Assoc	0.57	0.76	0.86	0.52	0.71	0.65	0.71	0.65
	CoD	0.43	0.95	0.74	0.50	0.69	0.52	0.76	0.52
	Neg	0.78	0.87	0.18	0.76	0.73	0.78	0.68	0.78
	ToD	0.93	0.97	0.96	0.90	0.93	0.89	0.90	0.89
	macro avg	0.68	0.89	0.68	0.67	0.77	0.71	0.77	0.71
	weight avg	0.77	**0.92**	0.56	0.77	0.79	0.78	0.77	0.78
F1-score	Assoc	0.73	0.80	0.71	0.61	0.77	0.71	0.50	0.71
	CoD	0.55	0.88	0.65	0.60	0.70	0.61	0.66	0.61
	Neg	0.78	0.90	0.29	0.76	0.79	0.78	0.76	0.78
	ToD	0.84	0.95	0.68	0.84	0.84	0.84	0.87	0.84
	accuracy	0.77	**0.92**	0.56	0.77	0.79	0.78	0.77	0.78
	macro avg	0.72	0.88	0.58	0.70	0.77	0.73	0.70	0.73
	weight avg	0.76	0.91	0.50	0.76	0.79	0.78	0.78	0.78

- R=Ratio 80:20, F=5-fold cross validation, U=under sampling, O=over sampling, S=SMOTE, Assoc=association, CoD=cause of disease, Neg=negative, ToD=treatment of disease.

E. Long Short-Term Memory

The LSTM classifier also shows performance that is not inferior to the previous classifier. The classifier got a score of 0.94 for precision but got a score of 0.93 for Recall and F1-score. This study highlights scores in bold text on weighted averages, where they are generally very effective when classes have unequal numbers (Table IX). This achievement is obtained through an under-sampling scheme, in which the sample was adjusted using minority data to make it balanced. Unfortunately, even though the data-level method uses under-sampling, it is only suitable through 5-fold cross-validation for the distribution of training and testing samples. The ratio approach has the worst results compared to other schemes.

The performance of this classifier pays attention to the minority class, namely the Association, which only has 34 sample data. Meanwhile, the other class was 10 multiplied that of the Association class. It is shows that this classifier is suitable for use on unbalanced data through under-sampling and 5-fold cross-validation methods. This result is different from the study conducted by [34] which found the fact that the use of SMOTE in LSTM had an impact on improving model performance.

TABLE IX. LONG SHORT-TERM MEMORY CLASSIFIER RESULTS

	Class	R	F	U+R	U+F	O+R	O+F	S+R	S+F
Precision	Assoc	0.71	1.00	0.17	1.00	0.38	0.57	0.42	0.75
	CoD	0.85	0.67	0.34	1.00	0.84	0.58	0.56	0.60
	Neg	0.69	0.82	0.74	0.75	0.77	0.82	0.85	0.66
	ToD	0.91	0.96	0.70	1.00	0.84	0.86	0.84	0.80
	macro avg	0.79	0.86	0.49	0.94	0.71	0.71	0.67	0.70
	weight avg	0.80	0.87	0.67	**0.94**	0.80	0.80	0.80	0.71
Recall	Assoc	0.56	0.56	1.00	1.00	0.67	1.00	0.71	0.74
	CoD	0.53	0.73	0.57	0.67	0.61	0.58	0.86	0.54
	Neg	0.88	0.87	0.50	1.00	0.83	0.80	0.70	0.80
	ToD	0.80	0.92	0.73	0.89	0.82	0.86	0.84	0.75
	macro avg	0.69	0.77	0.70	0.89	0.73	0.81	0.78	0.71
	weight avg	0.78	0.86	0.60	**0.93**	0.78	0.80	0.77	0.70
F1-score	Assoc	0.63	0.71	0.30	1.00	0.48	0.73	0.53	0.74
	CoD	0.66	0.70	0.42	0.80	0.71	0.58	0.68	0.57
	Neg	0.77	0.85	0.60	0.86	0.80	0.81	0.77	0.72
	ToD	0.85	0.94	0.71	0.94	0.83	0.86	0.84	0.77
	accuracy	0.78	0.86	0.60	**0.93**	0.78	0.80	0.77	0.70
	macro avg	0.73	0.80	0.51	0.94	0.71	0.75	0.70	0.70
	weight avg	0.78	0.86	0.62	0.93	0.78	0.80	0.78	0.70

- R=Ratio 80:20, F=5-fold cross validation, U=under sampling, O=over sampling, S=SMOTE, Assoc=association, CoD=cause of disease, Neg=negative, ToD=treatment of disease.

IV. CONCLUSION AND FUTURE WORK

The purpose of comparative analysis is to understand the differences and similarities between the objects being compared and to evaluate the advantages and disadvantages of each object. This study compares Precision (P), Recall (R), and F1-score (F1) metrics from various algorithms to produce an optimal model. Because the data is unbalanced, a sampling method is needed which involves under sampling, over sampling, and synthetic sampling. Each classifier is tested on eight schemes consisting of: R, F, U+R, U+F, O+R, O+F, S+R, and S+F. In addition consider to the number of balanced classes, the scheme also aims to test the performance of models with unbalanced data. This aims to find out whether the application of sampling as a mandatory thing is used or not. Test results on five classifiers through eight schemes on imbalance data produce varying P, R, and F1 metrics. However, the main goal is to find the most optimal model. P and R values are very important when the data is balanced, meaning that the model can predict classes with high accuracy and is able to identify most of the true class samples. On the other hand, if the data is unbalanced, P and R are not sufficient to evaluate the performance of the classification model as a whole. To evaluate the performance of the classification model on unbalanced data, F1-score is the most suitable metric. The F1-score measures of the harmonic average of the P and R scores. This study still requires improvisation and is still very much open for further study. In the future, this study will continue to classify through various approaches by considering the semantics of each word.

REFERENCES

[1] Azarian, R. Potentials and Limitations of Comparative Method in Social Science. International Journal of Humanities and Social Science (2011), 1(4), 113–125.

[2] Kim, M., & Hwang, K. B. An empirical evaluation of sampling methods for the classification of imbalanced data. PLoS ONE (2022), 17(7 July), 1–22. https://doi.org/10.1371/journal.pone.0271260

[3] Chabalala, Y., Adam, E., & Adem Ali, K. Exploring the Effect of Balanced and Imbalanced Multi–Class Distribution Data and Sampling Techniques on Fruit–Tree Crop Classification Using Different Machine Learning Classifiers. Geomatics (2023), 3(January), 70–92. https://doi.org/10.3390/geomatics3010004.

[4] Alsafy, B. M., Aydam, Z. M., & Mutlag, W. K. Multiclass Classification: A Review. International Journal of Advanced Engineering Technology and Innovative Science (2014), 3(4), 65–69.

[5] Suhaimi, N. S., Othman, Z., & Yaakub, M. R. (2023). Comparative Analysis Between Macro and Micro-Accuracy in Imbalance Dataset for Movie Review Classification. In X.-S. Yang, S. Sherratt, N. Dey, & A. Joshi (Eds.), Proceedings of Seventh International Congress on Information and Communication Technology (pp. 83–93). Springer Nature Singapore.

[6] Li, H., Zou, P., Han, W., & Xia, R. (2013). A combination method for multi-class imbalanced data classification. Proceedings - 2013 10th Web Information System and Application Conference, WISA 2013, 1, 365–368. https://doi.org/10.1109/WISA.2013.75.

[7] Arafat, M. Y., Hoque, S., Xu, S., & Farid, D. M. (2019). Machine learning for mining imbalanced data. IAENG International Journal of Computer Science, 46(2), 332–348.

[8] Zhou, H., Li, X., Wang, C., & Ma, Y. (2022). A feature selection method based on term frequency difference and positive weighting factor. Data and Knowledge Engineering, 141(August), 102060. https://doi.org/10.1016/j.datak.2022.102060.

[9] Takahashi, K., Yamamoto, K., Kuchiba, A., & Koyama, T. (2022). Confidence interval for micro-averaged F 1 and macro-averaged F 1 scores. Applied Intelligence, 52(5), 4961–4972. https://doi.org/10.1007/s10489-021-02635-5.

[10] Vong, C. M., & Du, J. (2020). Accurate and efficient sequential ensemble learning for highly imbalanced multi-class data. Neural Networks, 128, 268–278. https://doi.org/10.1016/j.neunet.2020.05.010.

[11] Zhu, T., Liu, X., & Zhu, E. (2022). Oversampling with Reliably Expanding Minority Class Regions for Imbalanced Data Learning. IEEE Transactions on Knowledge and Data Engineering, 14(8). https://doi.org/10.1109/TKDE.2022.3171706.

[12] Kaur, P., & Gosain, A. (2018). Comparing the behavior of oversampling and undersampling approach of class imbalance learning by combining class imbalance problem with noise. Advances in Intelligent Systems and Computing, 653(January), 23–30. https://doi.org/10.1007/978-981-10-6602-3_3

[13] Krawczyk, B., Koziarski, M., & Wozniak, M. (2020). Radialbased oversampling for multiclass imbalanced data classification. IEEE Transactions on Neural Networks and Learning Systems, 31(8), 2818–2831. https://doi.org/10.1109/TNNLS.2019.2913673

[14] Krawczyk, B., Bellinger, C., Corizzo, R., & Japkowicz, N. (2021). Undersampling with Support Vectors for Multi-Class Imbalanced Data Classification. Proceedings of the International Joint Conference on Neural Networks, 2021-July. https://doi.org/10.1109/IJCNN52387.2021.9533379

[15] Liu, C. L., & Hsieh, P. Y. (2020). Model-Based Synthetic Sampling for Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering, 32(8), 1543–1556. https://doi.org/10.1109/TKDE.2019.2905559

[16] Kim, B., Choi, W., & Lee, H. (2019). A corpus of plant–disease relations in the biomedical domain. PLoS ONE, 14(8), 1–19. https://doi.org/10.1371/journal.pone.0221582.

[17] Wongvorachan, T., He, S., & Bulut, O. (2023). A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining. Information (Switzerland), 14(1). https://doi.org/10.3390/info14010054.

[18] Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). RUSBoost: A hybrid approach to alleviating class imbalance. IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans, 40(1), 185–197. https://doi.org/10.1109/TSMCA.2009.2029559.

[19] Hajibabaee, P., Pourkamali-Anaraki, F., & Hariri-Ardebili, M. A. (2023). Dimensionality reduction techniques in structural and earthquake engineering. Engineering Structures, 278(January), 115485. https://doi.org/10.1016/j.engstruct.2022.115485.

[20] Barua, A., Sharif, O., & Hoque, M. M. (2021). Multi-class Sports News Categorization using Machine Learning Techniques: Resource Creation and Evaluation. Procedia Computer Science, 193, 112–121. https://doi.org/10.1016/j.procs.2021.11.002.

[21] Kibriya, A. M., Frank, E., Pfahringer, B., & Holmes, G. (2004). Multinomial naive bayes for text categorization revisited. Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), 3339, 488–499. https://doi.org/10.1007/978-3-540-30549-1_43.

[22] Beckmann, M., Ebecken, N. F. F., & Pires de Lima, B. S. L. (2015). A KNN Undersampling Approach for Data Balancing. Journal of Intelligent Learning Systems and Applications, 07(04), 104–116. https://doi.org/10.4236/jilsa.2015.74010.

[23] Cover, T. M., & Hart, P. E. (1967). Nearest Neighbor Pattern Classification. IEEE Transactions on Information Theory, 13(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

[24] Sang, Y., Zhang, H., & Zuo, L. (2008). Least squares support vector machine classifiers using PCNNs. 2008 IEEE International Conference on Cybernetics and Intelligent Systems, CIS 2008, 290–295. https://doi.org/10.1109/ICCIS.2008.4670890.

[25] Kumar, M., Pachori, R. B., & Acharya, U. R. (2017). Automated diagnosis of myocardial infarction ECG signals using sample entropy in flexible analytic wavelet transform framework. Entropy, 19(9). https://doi.org/10.3390/e19090488.

[26] Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. Journal of Big Data, 6(1). https://doi.org/10.1186/s40537-019-0192-5.

[27] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(6), 1735–1780. https://doi.org/10.17582/journal.pjz/2018.50.6.2199.2207.

[28] Sevastianov, L. A., & Shchetinin, E. Y. (2020). On methods for improving the accuracy of multi-class classification on imbalanced data. CEUR Workshop Proceedings, 2639, 70–82.

[29] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing and Management, 45(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

[30] Yao, B., & Wang, L. (2021). An Improved Under-sampling Imbalanced Classification Algorithm. Proceedings - 2021 13th International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2021, 775–779. https://doi.org/10.1109/ICMTMA52658.2021.00178

[31] Xiaojuan, M. (2018). Research on the classification of high dimensional imbalanced data based on the optimization of random forest algorithm. ACM International Conference Proceeding Series, 60–67. https://doi.org/10.1145/3297730.3297747

[32] Shi, Q., & Zhang, H. (2021). An improved learning-based LSTM approach for lane change intention prediction subject to imbalanced data. Transportation Research Part C: Emerging Technologies, 133(November), 103414. https://doi.org/10.1016/j.trc.2021.103414

[33] Rabbimov, I. M., & Kobilov, S. S. (2020). Multi-Class Text Classification of Uzbek News Articles using Machine Learning. Journal of Physics: Conference Series, 1546(1). https://doi.org/10.1088/1742-6596/1546/1/012097

[34] Turner, K. E., Thompson, A., Harris, I., Ferguson, M., & Sohel, F. (2022). Deep learning based classification of sheep behaviour from accelerometer data with imbalance. Information Processing in Agriculture, xxxx, 1–14. https://doi.org/10.1016/j.inpa.2022.04.001

[35] N. S. Rahmi, N. W. S. Wardhani, M. B. Mitakda, R. S. Fauztina, and I. Salsabila, "SMOTE Classification and Random Oversampling Naive Bayes in Imbalanced Data : (Case Study of Early Detection of Cervical Cancer in Indonesia)," Proc. 2022 IEEE 7th Int. Conf. Inf. Technol. Digit. Appl. ICITDA 2022, pp. 1–6, 2022, doi: 10.1109/ICITDA55840.2022.9971421.

[36] D. N. Pratama, O. N. Pratiwi, and E. Sutoyo, "Classification of Questions Based on Difficulty Levels using Support Vector Machine and Naïve Bayes Algorithms for Imbalanced Class," Proc. - 2021 4th Int. Conf. Comput. Informatics Eng. IT-Based Digit. Ind. Innov. Welf. Soc. IC2IE 2021, pp. 40–45, 2021, doi: 10.1109/IC2IE53219.2021.9649149.

[37] N.S. Suhaimi, Z. Othman, and M. R. Yaakub, "Comparative Analysis Between Macro and Micro-Accuracy in Imbalance Dataset for Movie Review Classification," in Proceedings of Seventh International Congress on Information and Communication Technology, 2022, vol. 3, pp. 83–94.

[38] S. Chua, C. I. Sii, and P. N. E. Nohuddin, "Comparative Analysis of Machine Learning Models for Fitness Level Prediction with Imbalanced Dataset," in International Conference on Digital Transformation and Intelligence, ICDI 2022, 2022, no. Icdi, pp. 102–106, doi: 10.1109/ICDI57181.2022.10007339.

[39] Y. Xu, Y. Zhang, J. Zhao, Z. Yang, and X. Pan, "KNN-based maximum margin and minimum volume hyper-sphere machine for imbalanced data classification," Int. J. Mach. Learn. Cybern., vol. 10, no. 2, pp. 357–368, 2019, doi: 10.1007/s13042-017-0720-6.

[40] M. Maydanchi et al., "Comparative Study of Decision Tree, AdaBoost, Random Forest, Naïve Bayes, KNN, and Perceptron for Heart Disease Prediction," in Conference Proceedings - IEEE SOUTHEASTCON, 2023, vol. 2023-April, pp. 204–208, doi: 10.1109/SoutheastCon51012.2023.10115189.

[41] E. Ekinci, "Classification of Imbalanced Offensive Dataset – Sentence Generation for Minority Class with LSTM," Sak. Univ. J. Comput. Inf. Sci., vol. 5, no. 1, 2022, doi: 10.35377/saucis...1070822.