Leave this box blank

Multi-class Classification with Many Categories

Peng-Yong Heng[1], Choo-Yee Ting[2], Hui-Ngo Goh[3]

Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, Cyberjaya, 63100, Malaysia
E-mail: 1211306514@student.mmu.edu.my, cyting@mmu.edu.my, hngoh@mmu.edu.my

Abstract— **Multi-class classification requires distinguishing between multiple classes, often 10 or more classes. Researchers are dealing with imbalanced data, computational challenges and others. Imbalanced data can lead to biased predictions for minority classes, making it difficult for models to learn efficient decision boundaries. This study aimed to investigate the performance of the proposed framework to improve multiclass classification. Different classifiers, such as Random Forest (RF), Gaussian Naive Bayes (Gaussian NB), Decision Tree (DT), K-Nearest Neighbours (KNN), and Support Vector Machine (SVM), were evaluated on three datasets. The first dataset had the target variable 'JS_Job_sector' with 13 classes. The second dataset with the target variable 'Course' with 17 classes. The third dataset with the target variable 'Brands' comprises 22 classes. Macro metrics like accuracy, precision, recall, and F_1-score were considered. A comparison was made between models with and without using the sentence embedding method on feature variables. The sentence transformer models used were 'paraphrase-MiniLM-L6-v2' and 'paraphrase-multilingual-MiniLM-L12-v2' from Hugging Face. The proposed framework incorporated feature selection (BorutaShap or Boruta) and SMOTE, which generally improved classifier performance. However, exceptions were noted as the Decision Tree (DT) on dataset 1, the accuracy and precision of the Random Forest (RF) on dataset 2, as well as the RF, DT and Gaussian NB on dataset 3, where the baseline model performed better. Focusing on the proposed framework, without the sentence embedding method, DT on dataset 1 and RF on dataset 2 achieved the highest accuracies of 88.8% and 51.1%, respectively. For dataset 3, using the sentence embedding method, RF, KNN, and SVM achieved the highest accuracy of 99.7%.**

Keywords— **multi-class classification, machine learning classifiers, data visualization, feature selection, sentence embedding method.**

I. INTRODUCTION

Multi-class classification, unlike binary classification which involves only two categories, is a machine learning task where data points are categorized into one of three or more classes. In this research title, the term 'many categories' typically refers to having 10 or more classes. The application of multi-class classification can be a crucial aspect across various domains such as document classification, image recognition and others. A good multi-classifier helps in improving decision-making, personalization, and resource allocation. Multi-class classification is a powerful tool that drives applications across various fields. It facilitates tasks like sentiment analysis in natural language processing [1], enabling businesses to gauge public opinion and customer feedback. In image classification, it aids in organizing visual data and automating tasks from autonomous driving to medical diagnostics. In healthcare, it helps doctors identify illnesses and create treatment approaches specific to each patient [2], while in marketing, it allows for grouping customers based on shared characteristics to develop targeted advertising campaigns. Despite its importance, challenges hinder its accuracy and limit its potential.

Real-world datasets often demonstrate imbalanced class distributions, where some classes are significantly more frequent than others. It is often the case that the stated standard learning algorithms, designed for balanced training datasets, struggle with the skewed distribution inherent to these problems, leading to significantly reduced accuracy in predicting minority classes [3]. Consider a scenario in healthcare applications where data imbalance is a common challenge. For instance, disease samples might make up only 0.01% to 0.02% of the entire dataset [4]. This highlights the prevalent issue of multi-class highly imbalanced data, demanding effective solutions for accurate disease detection. Imbalanced datasets can result in a bias towards the majority class because certain machine learning algorithms may give more weight to the class with a larger number of observations, even though this class may not be as significant as another [5].

To prevent low accuracy in multi-class classification results, sampling algorithms may be employed to address class or data imbalance [6]. Integrating different sampling techniques can result in notable enhancements in classifier performance in imbalanced class distributions [7]. Additionally, feature selection methods are instrumental in tackling the issue of imbalanced data [5].

The objectives of this project are:

1. To recommend suitable methods for different dataset characteristics
2. To evaluate the methods on different datasets
3. To recommend suitable multi-class classification framework for different dataset

The remainder of the paper is structured to provide a clear flow of information. Section II presents existing literature relevant to the research topic. Section III presents the materials and methods used. Section IV presents the results obtained from the experiments and discusses their significance. Finally, Section V summarises the key findings and offers concluding remarks.

II. Literature Review

Multi-class classification presents several challenges in real-world applications due to the complexity of handling many categories. One key challenge is the imbalanced classes problem with some classes having significantly more data points than others in multi-class classification datasets [1]. This can lead to models biased towards the majority class and performing poorly on the minority classes.

Alongside class imbalance, class overlap in real-world data often has classes that share similar characteristics, making it difficult for models to distinguish between them accurately. Class overlap, where classes share similar characteristics, making them difficult to distinguish accurately for models [8]. This can lead to models biased towards the majority class and performing poorly on the minority classes, especially when dealing with overlapping classes.

A. Dataset Types Used by Researchers

TABLE I

DATASET TYPES USED BY RESEARCHERS

Author	Unstructured data	Structured Data	Image Classification
[9]	✓		
[10]		✓	
[1]	✓		
[11]			✓
[12]		✓	
[13]	✓		
[14]		✓	
[15]		✓	
[16]		✓	
[17]	✓		
[18]	✓	✓	
[19]	✓		
[20]	✓		
[21]		✓	

TABLE I showed the dataset types used by researchers. The datasets utilized in various research papers encompass diverse fields and purposes. The Iris dataset in [10], [19] is common in machine learning studies, offering three species with four measured features. Examined English song lyrics in [9], presenting challenges due to varied sample sizes and stylistic diversity. Reference [1] focused on sentiment analysis using manually labelled tweet datasets. Reference [12] explored imbalanced datasets from the KEEL repository. Reference [15] used real-life datasets for regression analysis sourced from the KEEL Dataset Repository. Reference [18] employed a Twitter-extracted dataset for cyberbullying classification. Reference [14] employed Cardiotocography, Wine Quality, Hypo Thyroid, and Yeast datasets for classification tasks. Reference [16] examined undergraduate student data for predicting academic outcomes. Reference [17] utilized datasets like Car Evaluation, Lymphography, Nursery, and Balance Scale for multi-class classification testing. Reference [18] employed wine quality, Hypo Thyroid, Yeast, E-coli, and COVID-19 Q&A datasets for classification purposes.

B. Feature Selection Techniques Used by Researchers

TABLE II

FEATURES SELECTION TECHNIQUES

Author	Chi-square	Boruta	BorutaShap	ANOVA	LASSO
[22]	✓				
[13]	✓				
[23]		✓			
[6]		✓			
[24]		✓	✓		
[25]			✓		
[2]				✓	
[26]		✓			
[27]					✓
[28]					✓

TABLE II shows feature selection techniques used by researchers. Chi-square feature selection is used to identify and select the most significant features out of the complete set that impact the performance of the intrusion detection system. By applying the chi-square test, the authors were able to reduce the number of features from 41 to 31, keeping only those attributes that have a higher statistical significance about the classification of network traffic as normal or intrusive [22]. This step is crucial for the model's accuracy because it ensures that only relevant information is fed into the classifier, thereby improving the efficiency and

effectiveness of the intrusion detection process. Studies have shown that Boruta, a feature selection technique utilizing random forest-like shadow variables, offers advantages over other methods [23]. These advantages include improved variable selection accuracy, faster computation times, and simpler interpretation of the results. BorutaShap, a feature selection method merging Boruta's strength with SHAP values, helps identify the most impactful features for classification tasks [24]. Both Boruta and BorutaShap reduce data dimensionality by filtering out redundant information, ultimately boosting classification model performance [25].

Applying feature selection with ANOVA helped achieve better prediction performance in the classification task. By identifying 275 out of 1664 features as the most relevant, ANOVA effectively reduced model complexity while maintaining accuracy [2]. The least absolute shrinkage and selection operator (LASSO) automatically identifies and discards redundant features, resulting in a more focused model [27], [28].

C. Method for handling imbalanced large classes

TABLE IIII

METHOD FOR HANDLING IMBALANCED LARGE CLASSES

Author	Hybrid approach	Ensemble Methods	Transfer Learning	Feature Engineering	Sampling algorithm	Semi-supervised learning	Random shuffling method	Term weighting method
[6]					✓			
[12]		✓		✓				
[14]	✓							
[15]		✓						
[16]		✓						
[29]					✓	✓		
[30]							✓	
[21]	✓							
[17]					✓			
[18]					✓			
[13]					✓			
[11]			✓					
[31]					✓			
[32]					✓			✓
[33]					✓			
[34]					✓			

The challenge of imbalanced classes in classification tasks is significant and can lead to overfitting, affecting the accuracy of models. TABLE III showed various methods have been proposed to address this issue. Reference [6] utilized random sampling, Synthetic Minority Oversampling Technique (SMOTE), and Adaptive Synthetic Sampling (ADASYN). Reference [18] employed oversampling and undersampling techniques. Reference [13] utilized SMOTE combined with the weighted cost for misclassification. Reference [12] introduced under-sampling using support vectors. Reference [14] proposed a hybrid approach with a duo decision tree model and iterative ensemble sampling. Reference [15] proposed an ensemble method combining data partitioning and multiple classifiers. Reference [16] utilized ensemble classifiers like Random Forest and AdaBoost. Reference [29] employed the DFCM-MC algorithm with semi-supervised learning and random sampling techniques. Reference [30] introduced novel mini-batch shuffling strategies to prevent bias towards the majority class. Reference [11] combined AdaBoost with CNN through transfer learning. Reference [31] used fine-tuned generative models for oversampling in text classification. Reference [32] proposed the Similarity Oversampling and Undersampling Preprocessing algorithm. Reference [33] explored Supervised Term Weighting (STW) schemes, with the sqrt tf-igm scheme showing promising results.

D. Machine Learning Techniques Used by Researchers

TABLE IV

MACHINE LEARNING TECHNIQUES USED BY RESEARCHERS

Author	DT	SVM	RF	KNN	NB
[34]		✓		✓	
[9]		✓		✓	
[14]	✓				
[10]					✓
[35]					✓
[33]		✓		✓	✓
[16]			✓		
[19]		✓			
[20]		✓			
[21]	✓				

TABLE IV shows the machine learning techniques used by researchers.

The Gaussian Naive Bayes (Gaussian NB) classifier is a supervised learning method used for classification tasks. It is based on Bayes' theorem and assumes independence between predictors. In this instance, a Gaussian distribution is assumed, making it suitable for continuous data[10].

Random Forest (RF) is an ensemble classifier that consists of numerous decision tree models. It is a type of bagging ensemble used for both regression and classification tasks. It helps increase the robustness of the model by reducing the likelihood of overfitting [16].

Decision Tree (DT) algorithm, a form of supervised learning, constructs a tree-like model where decision nodes correspond to features or attributes and leaf nodes denote class labels or predicted values. This hierarchical structure is built through iterative partitioning of input data based on feature values, aimed at minimizing impurity or maximizing Information Gain (IG). DT can be used for classification 13 tasks, such as determining whether a student might fail in a certain semester [21].

K-Nearest Neighbors (KNN) is a simple, non-parametric method that classifies new instances based on the majority vote of their k closest neighbors from the training set. This algorithm can naturally handle multiclass problems by considering the class membership of the k-nearest points to the query point and choosing the most common class among them [9]. Support Vector Machine implements the Support Vector Classification (SVC) algorithm. In general, SVM is a supervised learning model used for classification and regression problems. However, standard SVM is inherently a binary classifier [19]. For multiclass problems, the technique needs to be extended either by using a one-vs-all approach, where a classifier is trained for each class against all other classes, or a one-vs-one approach, where a classifier is trained for every pair of classes. The paper explores ways to optimize SVM classification for datasets with multiple classes by improving or engineering the features that are fed into the model.

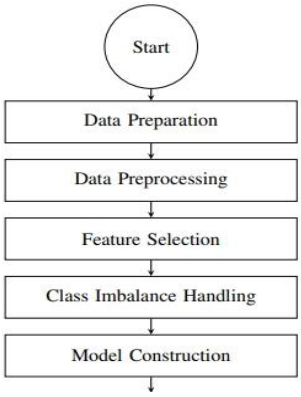
On the other hand, the word embeddings method was numerical representations of words, typically in the form of vectors in a high-dimensional space, that encode the semantic meaning of words such that words with similar meanings are located close to each other in that space. These embeddings are learned from text data and can capture complex relationships between words, such as synonyms, antonyms, and analogies. Word embeddings are used in various applications, including natural language processing tasks like text classification, sentiment analysis, and machine translation [36].

After reviewing all the related works, it is evident that handling imbalanced data is important for dealing with the multi-class classification problem as it ensures that the model can accurately predict minority classes, prevents bias towards majority classes, and improves overall performance metrics such as accuracy, precision, recall, and F1-score.

III. The Material and Method

In this study, there were 3 datasets involved in the experiments. Fig. 1 illustrates the project's methodology flowcharts. Feature selection identifies important features, and a class imbalance technique balances the dataset. There were 5 classifier models RF, Gaussian NB, DT, KNN and SVM were used.

Before deployment of the classifier model, train test split was performed towards all three datasets with the ratio of 80% training data to 20% testing data, whenever applicable.



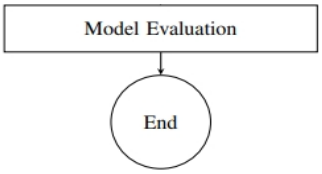


Fig. 1 Flowchart of methods

A. Data Preparation

1) *Dataset 1*: The first dataset was collected from the university Graduate Tracer Study, which was an annual research endeavour overseen by the Ministry of Higher Education (MOHE) to assess post-graduation employment statuses. The target variable for dataset 1 included 13 classes.

TABLE V presented the overview of dataset 1 including the target variable. Fig. 2 presented the target variable, which included 13 classes.

TABLE V

Overview of Dataset 1

Data	**Features**
Variables	Student Information, Nationality, Race, Gender, Disability, Marital Status, Date of Birth, Age, Student Address, Educational Information, Enrollment and Graduation Status, Academic Achievement, Credit Transfer Status, Academic Score
Target Variable	JS_Job_sector

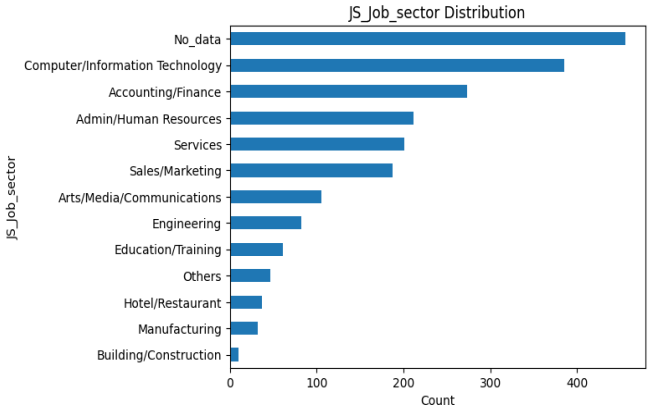


Fig. 2 Classes of the Target variable (dataset 1)

2) *Dataset 2:* The second dataset was to pinpoint students who may be at risk of dropping out early in their academic journey and it was collected from the UCI Machine Learning Repository. The data for dataset 2 were preprocessed into numerical features and are available at the UCI Machine Learning Repository [37].

TABLE VI presented the overview of dataset 2 including the target variable. Fig. 3 presented the target variable, which included 17 classes.

TABLE IVI

Overview of Dataset 2

Data	**Features**

Variables	Application mode, Daytime/evening attendance, qualification, Nationality, Admission grade, Mother's occupation, Father's occupation, Gender, Age at enrollment, Curricular units
Target Variable	Course

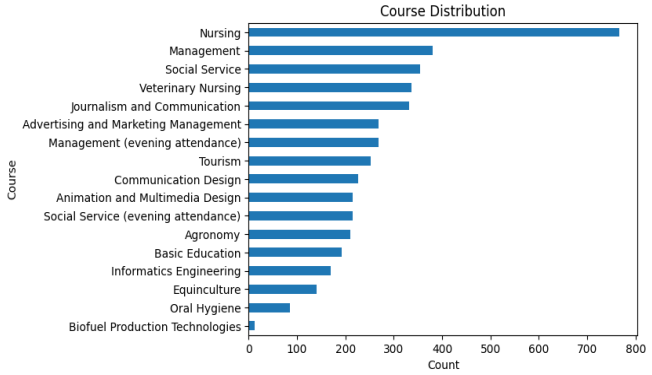


Fig. 3 Classes of the Target variable (dataset 2)

3) *Dataset 3:* The third dataset consisted of information on car prices in Australia for the year 2023. This study focused on car brands. The data was collected from the Kaggle website.

TABLE VII presented the overview of dataset 3 including the target variable. Fig. 4 presented the target variable, which included 22 classes.

TABLE VII

OVERVIEW OF DATASET 3

Data	**Features**
Variables	Year, Model, UsedOrNew, Transmission, DriveType, FuelType, FuelConsumption, Kilometres, Location, CylindersinEngine, BodyType, Doors, Seats, Price
Target Variable	Brand

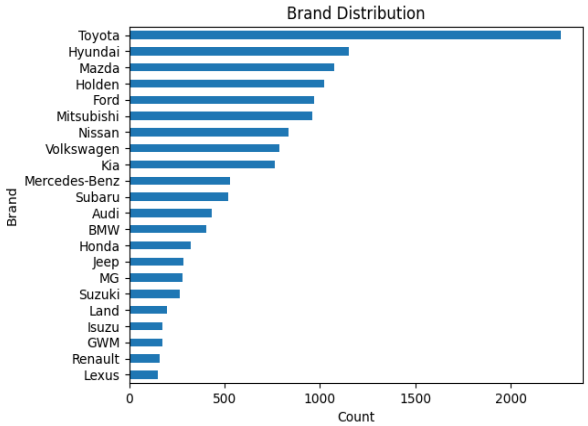


Fig. 4 Classes of the Target variable (dataset 3)

B. Data Preprocessing

Removing columns from all three datasets based on their correlation with other variables. Eliminate columns with high correlation coefficients (indicating redundancy) and low correlation coefficients.

In Dataset 1, 7 rows with null values were dropped, and the categorical columns with null values were filled in with 'No_data'. Datasets 2 and 3 had no missing values.

For data transformation, Label encoding was applied to categorical variables in datasets 1 and 3. However, dataset 2 was already preprocessed and contained numerical features, eliminating the need for further encoding.

Then, for all the datasets, sentence embedding techniques were employed to capture the semantic meaning of textual variables. Sentence transformers model from Hugging Face were used, specifically the 'paraphrase-MiniLM-L6-v2' and 'paraphrase-multilingual-MiniLM-L12-v2'.

Main_job_scope	Main_job_Title	Work_in_the_same_field_of_learning	Faculty_Description	JS_Job_sub_sector
Professional	Lawyer	Yes	Faculty of Law	Lawyer/Legal Asst
Professional	Human Resources Executive	Yes	Faculty of Management	Human Resources
No_data	No_data	No_data	Faculty of Law	No_data
Technicians and Associate Professionals	Assistant Counsel	Yes	Faculty of Law	Lawyer/Legal Asst

Fig. 5 Origin of Data Structure

Fig. 5 shows the origin of the data structure.

```
Main_job_scope: Professional,
Main_job_Title: Lawyer,
Work_in_the_same_field_of_learning: Yes,
Faculty_Description: Faculty of Law,
JS_Job_sub_sector: Lawyer/Legal Asst,
Employed_Company_lat: 3.162835,
Faculty: FOL,
Program_Description: BACHELOR OF LAW (HONOURS),
Employed_Company_Postcode: 50400,
Economic_sector: Administrative activities and support services,
Current_Job_scope: Temporary/parttime
```

Fig. 6 Compiled data structure

Fig. 6 shows the data structure in which the data were compiled. These compiled data were then used as input to load the sentence transformer model, which subsequently encoded the sentences into embeddings.

C. Experiments

TABLE VII shows the experiment conducted on the datasets in this project. Two different methods were used on the feature variables as the table shown were label encoder and sentence transformer. Experiments 1 and 2 were conducted on all the three datasets.

TABLE VIII

DESCRIPTION OF EXPERIMENTS

Experiments	**Description**
1	Using label encoding method on feature variables
2	Using sentence embedding method on feature variables (sentence transformer models used were 'paraphrase-MiniLM-L6-v2' and 'paraphrase-multilingual-MiniLM-L12-v2')

D. Feature Selection

Feature selection helps identify the variables with the strongest influence on a machine learning classifier's prediction accuracy. This process reduces model complexity by removing less important features, ultimately retaining only the essential ones for optimal performance.

First, datasets were split into 80% training and 20% testing data. After using BorutaShap in dataset 1, 17 feature variables have been selected. Next, after using Boruta in datasets 2 and 3, 18 and 12 feature variables were selected respectively.

E. Class Imbalance Treatment

To address the class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) technique was applied to the training data after splitting the dataset for training and testing. SMOTE generates synthetic samples for the minority class to balance the class distribution based on the original data.

For each positive example x_i, SMOTE creates a new instance by combining this example with one of its k nearest neighbors from the positive class in the feature space [38], as defined in equation (1):

$$x_i' = x_i + \lambda(x_n - x_i) \tag{1}$$

x_i is an original minority class sample. x_n is a randomly selected k-nearest neighbor of x_i. λ is a random number between 0 and 1.

F. Model Construction

Various classification models used were Random Forest (RF), Gaussian Naive Bayes (Gaussian NB), Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbors (KNN). Standard Scaler was applied to the training and testing data on both DT and KNN classifiers to mitigate biases in the classification models.

RF is a way of combining multiple Decision Trees to improve prediction accuracy. It trains different Decision Trees on different parts of the data and then combines their predictions for a final result. This helps prevent overfitting and makes it suitable for data with many features.

In the pictorial overview of the random forest (RF) algorithm below shown in Fig. 7, there are four individual decision trees that collectively form a Random Forest. Random Forest is an ensemble learning method, which means it enhances accuracy by combining multiple models to reach a final decision [39].

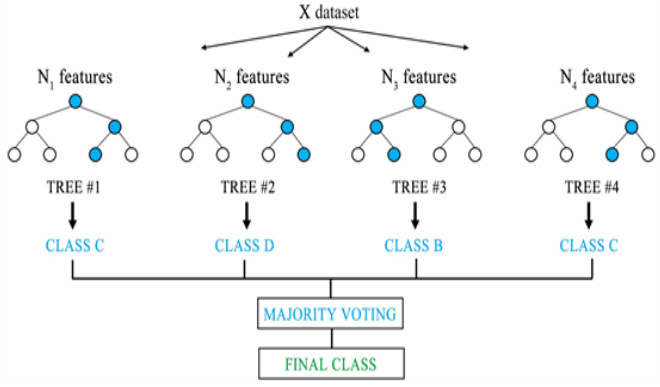


Fig. 7 A pictorial overview of the random forest (RF) algorithm.

When performing Random Forests for classification tasks, it's important to understand that the Gini index is frequently used to decide how nodes in a decision tree split. This index uses the class and its probability to determine the Gini value for each branch at a node, indicating the likelihood of each branch occurring. In the equation (2), p_i represents the relative frequency of the observed class in the dataset, and c denotes the number of classes. p_i is the proportion of instances of class i in node t, and C is the total number of classes.

$$Gini(t) = 1 - \lambda \sum_{i=1}^{c} (p_i)^2 \tag{2}$$

Equation (3) shows that using entropy to determine how nodes branch in a decision tree. Entropy uses the probability of a certain outcome to decide how the node should branch. Unlike the Gini index, it is more computationally intensive due to the logarithmic function used in its calculation. p_i is the proportion of instances of class i in node t.

$$Entropy(t) = \sum_{i=1}^{c} - p_i * log_2(p_i) \tag{3}$$

Next, Gaussian NB offers a simple yet powerful approach to classification problems with continuous features. Its ease of use, speed, and interpretability make it a popular choice for various machine-learning applications.

Equation (4) shows the Bayes' Theorem expression, where P(A|B) is the posterior probability and states probability of occurrence of A given B has happened, P(A) is the prior probability, P(B) is the probability of occurrence of event B, P(B|A) is the probability of occurrence of B given A has already happened.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{\mathrm{P(B)}} \tag{4}$$

Next, SVM is a powerful supervised learning algorithms known for their ability to tackle both classification and regression problems. In classification tasks, SVMs excel at finding the optimal decision boundary that separates data points belonging to different classes with the widest margin possible. This margin refers to the distance between the decision boundary and the closest data points from each class, called support vectors.

Equation (5) shows polynomial kernel, $K(x_i, x_i')$ of degree d (where d is positive). Classification results of the combination of non-linear kernel and support vector classifier are called the SVM [39].

$$K(x_i, x_i') = (1 + \sum_{j=1}^{p} x_{ij}\, x_{ij}')^d \tag{5}$$

Next, Decision Trees are a popular supervised learning method that builds a tree-like model for prediction. Each internal node represents a question asked about a feature in the data, and the branches represent the possible answers. By following these branches based on the data point's features,

we reach a leaf node containing the predicted outcome. Decision Trees are known for their versatility, interpretability, and ability to handle both classification and regression tasks effectively.

Entropy and Gini index formula has shown above in Equation (2) and Equation (3). Equation (6) shows calculation of information gain. S is the original set of instances. A is the attribute being split. S_i is the number of instances in the i^{th} child node, S is the total number of instances, and m is the number of child nodes.

$$Information\ Gain(S, A) = Entropy(s) - \sum_{i=1}^{m} \frac{|S_i|}{|S|} Entropy(S_i) \quad (6)$$

Lastly, KNN is a non-parametric supervised learning algorithm used for both classification and regression tasks. Unlike many algorithms, KNN doesn't require a complex training phase. Instead, it predicts the class or value of a new data point by considering the k closest data points from the training data. The prediction is based on either the majority vote of the neighbors' class labels or the average value of the neighbors. While KNN offers simplicity and doesn't require extensive training, it can be computationally expensive for very large datasets.

Equation (7) shows the Euclidean distance metrics $d(x, y)$ between two points x and y. N is the number of features such that, x = $\{x_1,\ x_2,\ x_3, \ldots,\ x_N\}$ and y = $\{y_1,\ y_2,\ y_3, \ldots,\ y_N\}$ [39].

$$d(x, y) = \sum_{i=1}^{N} \sqrt{x_i^2 - y_i^2} \quad (7)$$

G. Model Evaluation

Performance evaluation is crucial in machine learning. It is a process to evaluate the predictive models based on various performance metrics such as accuracy, precision, recall, and F_1-score are the four measurements used in the performance metric. The evaluation includes important measurements like True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), giving a complete picture of how well the model performs.

In classification model evaluation, TP indicates when the model correctly identifies positive cases, reflecting its ability to make accurate positive predictions. Similarly, TN represents instances where the model accurately identifies negative cases, showing its precision in predicting negative outcomes. FP arises when the model mistakenly classifies instances as positive when they belong to the negative category. Conversely, FN occurs when the model incorrectly predicts instances as negatives when they belong to the positive category.

Equation (8) showed a formula to calculate accuracy. A widely used metric for evaluating model performance, it may not be the most suitable choice for classification tasks with imbalanced class distributions. This is because accuracy does not adequately account for the importance of correctly identifying rare cases. Prioritizing accuracy in such scenarios can result in giving too much importance to the majority class, resulting in poor performance of minority classes [35].

$$Accuracy = \frac{\mathrm{TP} + \mathrm{T}N}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{F}N} \quad (8)$$

Equation (9) showed a formula to calculate the macro average that is used for calculating the metric for each class separately and then averages these values, treating all classes equally regardless of their size. This approach ensures that the performance of each class is given equal importance in the overall assessment.

Macro precision, macro recall and macro F_1 -score a valuable metrics for evaluating the overall performance of a classifier across multiple classes, ensuring that each class's precision is considered equally, which is particularly useful in the context of imbalanced datasets.

$$Macro\ Average = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} Metric_i \quad (9)$$

Equations (10), (11), and (12) showed the equations to calculate precision, recall and F_1-score.

Precision is a classification metric that measures the proportion of accurately predicted positive cases among all cases predicted as positive. It is especially important when the consequences of false positives are significant, as it ensures the reliability of positive predictions made by the model.

$$Precision = \frac{\mathrm{TP} + \mathrm{FP}}{\mathrm{TP} + \mathrm{FP}} \quad (10)$$

Recall measures the total number of positive cases that are captured by the positive predictions. In [40] stated that recall is not affected by class imbalance because it is only dependent on the positive group.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{F}N} \quad (11)$$

F_1-score or F score is an evaluation of the class imbalance problem and the combination of recall and precision.

$$\mathrm{F}_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (12)$$

IV. Results and Discussion

This section showed and described the results of the proposed classification models, namely LR, Gaussian NB, RF, DT and SVM classifier. Accuracy, macro precision, macro recall and macro F_1-score are presented in this section. The tables showed the method using the label encoder and the sentence embedding method. The experiment for the sentence embedding method was applied within the Boruta and SMOTE framework. The sentence transformer models used were 'paraphrase-MiniLM-L6-v2' and 'paraphrase-multilingual-MiniLM-L12-v2'.

TABLE IX

MODEL PERFORMANCES (DATASET 1)

Experiments	**Framework/Model**	**Classifier**	**Metric**			
			Accuracy	**Precision**	**Recall**	**F_1-score**
Label Encoding	Baseline	RF	0.830	0.710	0.596	0.603
		Gaussian NB	0.457	0.443	0.381	0.341
		DT	**0.921**	**0.822**	**0.832**	**0.825**
		KNN	0.600	0.394	0.346	0.339
		SVM	0.706	0.472	0.461	0.456
	SMOTE	RF	0.821	0.650	0.634	0.633
		Gaussian NB	0.529	0.484	0.392	0.377
		DT	**0.844**	**0.694**	**0.685**	**0.682**
		KNN	0.493	0.342	0.349	0.333
		SVM	0.706	0.489	0.465	0.468
	BorutaShap and SMOTE	RF	**0.885**	**0.788**	**0.795**	**0.787**
		Gaussian NB	0.744	0.587	0.614	0.595
		DT	**0.888**	**0.778**	**0.759**	**0.748**
		KNN	0.667	0.513	0.534	0.508
		SVM	0.737	0.549	0.544	0.540
Sentence Embedding	'paraphrase-MiniLM-L6-v2'	RF	0.768	0.609	0.571	0.564
		Gaussian NB	0.715	0.542	0.528	0.517
		DT	0.529	0.345	0.339	0.336
		KNN	0.677	0.560	0.643	0.566
		SVM	**0.818**	**0.699**	**0.643**	**0.638**
	'paraphrase-multilingual-MiniLM-L12-v2'	RF	0.648	0.444	0.440	0.429
		Gaussian NB	0.548	0.386	0.378	0.361
		DT	0.433	0.271	0.288	0.269
		KNN	0.500	0.365	0.340	0.338
		SVM	**0.682**	**0.433**	**0.448**	**0.429**

TABLE IX showed the comparison of model performance for dataset 1 with different frameworks and models. Using the sentence embedding method did not increase the overall percentage for all the classifiers.

Using a framework with Boruta and SMOTE increased the overall performance of all models. However, the accuracy, precision, recall and F_1-score of the DT model slightly dropped to 88.8%, 77.8%, 75.9% and 74.8% compared to the baseline. Then, SVM reached the best performance using the MiniLM-L6 model.

Considering the results presented in TABLE IX, the framework utilizing Boruta and SMOTE is chosen, as it led to an overall increase in performance metrics compared to the baseline and other methods. Notably, the RF and DT classifier achieved the highest accuracy of 88.5% and 88.8% among all classifiers in the 'BorutaShap and SMOTE' row.

TABLE X

MODEL PERFORMANCES (DATASET 2)

Experiments	Framework/Model	Classifier	Metric			
			Accuracy	**Precision**	**Recall**	**F_1-score**
Label Encoding	Baseline	RF	**0.528**	**0.482**	**0.449**	**0.438**
		Gaussian NB	0.160	0.209	0.228	0.335
		DT	0.383	0.334	0.340	0.825
		KNN	0.354	0.305	0.309	0.288
		SVM	0.405	0.364	0.338	0.318
	SMOTE	RF	**0.499**	**0.454**	**0.470**	**0.447**
		Gaussian NB	0.229	0.320	0.267	0.217
		DT	0.366	0.327	0.343	0.331
		KNN	0.310	0.283	0.312	0.284
		SVM	0.379	0.340	0.365	0.342
	Boruta and SMOTE	RF	**0.511**	**0.454**	**0.478**	**0.455**
		Gaussian NB	0.251	0.348	0.302	0.230
		DT	0.409	0.368	0.385	0.372
		KNN	0.367	0.347	0.373	0.350
		SVM	0.392	0.347	0.384	0.351
Sentence Embedding	'paraphrase-MiniLM-L6-v2'	RF	0.315	0.243	0.278	0.251
		Gaussian NB	0.216	0.163	0.205	0.165
		DT	0.205	0.184	0.180	0.180
		KNN	0.227	0.221	0.232	0.213
		SVM	**0.384**	**0.323**	**0.348**	**0.329**
	'paraphrase-multilingual-MiniLM-L12-v2'	RF	0.301	0.226	0.256	0.235
		Gaussian NB	0.244	0.198	0.220	0.185
		DT	0.195	0.161	0.171	0.163
		KNN	0.215	0.210	0.218	0.199
		SVM	**0.340**	**0.274**	**0.299**	**0.281**

TABLE X showed the comparison of model performance for dataset 2 with different frameworks and models. Using the sentence embedding method did not increase the overall percentage for all the classifiers.

Using a framework with Boruta and SMOTE increased the overall percentage. However, the accuracy and precision of the RF classifier slightly dropped to 51.1% and 45.4% compared to the baseline (52.8% and 48.2%), but the recall and F_1-score increased from 44.9% and 43.8% to 47.8% and 45.5%.

Considering the results presented in TABLE X, the framework utilizing Boruta and SMOTE is chosen, as it led to

an overall increase in performance metrics compared to the baseline and other methods. Notably, the RF classifier achieved the highest accuracy of 51.1% among all classifiers in the 'Boruta and SMOTE' row.

TABLE XI

MODEL PERFORMANCES (DATASET 3)

Experiments	**Framework/Model**	**Classifier**	**Metric**			
			Accuracy	**Precision**	**Recall**	**F_1-score**
Label Encoding	Baseline	RF	**0.981**	**0.983**	**0.967**	**0.974**
		Gaussian NB	0.257	0.340	0.345	0.267
		DT	**0.981**	**0.984**	**0.972**	**0.977**
		KNN	0.599	0.560	0.531	0.537
		SVM	0.673	0.682	0.603	0.616
	SMOTE	RF	**0.971**	**0.967**	**0.961**	**0.963**
		Gaussian NB	0.226	0.343	0.331	0.252
		DT	0.936	0.916	0.912	0.914
		KNN	0.620	0.578	0.630	0.594
		SVM	0.732	0.701	0.759	0.719
	Boruta and SMOTE	RF	**0.977**	**0.972**	**0.967**	**0.969**
		Gaussian NB	0.214	0.397	0.324	0.237
		DT	0.975	0.962	0.966	0.964
		KNN	0.855	0.822	0.858	0.837
		SVM	0.723	0.712	0.779	0.727
Sentence Embedding	'paraphrase-MiniLM-L6-v2'	RF	**0.996**	**0.995**	**0.992**	**0.993**
		Gaussian NB	0.882	0.873	0.873	0.869
		DT	0.949	0.933	0.937	0.934
		KNN	**0.996**	**0.993**	**0.995**	**0.994**
		SVM	**0.996**	**0.996**	**0.993**	**0.995**
	'paraphrase-multilingual-MiniLM-L12-v2'	RF	**0.997**	**0.997**	**0.996**	**0.996**
		Gaussian NB	0.863	0.852	0.873	0.854
		DT	0.946	0.935	0.933	0.933
		KNN	**0.997**	**0.996**	**0.997**	**0.996**
		SVM	**0.997**	**0.997**	**0.995**	**0.996**

TABLE XI showed the comparison of model performance before hyperparameter tuning for dataset 3 with different frameworks and models. Using the sentence embedding method increased the overall percentage for all the classifiers, with KNN and SVM showing significant improvement.

However, achieving such a high accuracy of 99.7% might also mask underlying biases within the dataset. A deeper analysis suggests that the dataset size was limited, potentially leading to overfitting and biases.

It is essential to use a larger dataset with a greater number of samples to enhance the model's generalizability and robustness. This will help address the limitations identified and ensure a more accurate assessment of the model's performance.

V. Conclusions

Using the proposed framework, which used feature selection (BorutaShap or Boruta) and SMOTE helped to address the imbalanced class problem by improving the performance for most of the model. Although some baseline models achieved better performance, this study still considered the proposed framework as the best performance.

For dataset 1 and dataset 2, without using the sentence embedding method, DT and RF were the winners, achieving an accuracy of 88.8% and 51.1% respectively. On the other hand, the sentence embedding method improved the performance of the classifier on dataset 3, RF, KNN and SVM reaching an accuracy of 99.7%.

A limitation of this study is the size of Dataset 3. Future work should employ a larger dataset with a greater number of samples to enhance the model's generalizability and robustness. This approach will help validate the findings from Dataset 3 and ensure a more accurate assessment of the model's performance across various datasets.

Additionally, exploring ensemble methods, which combine different machine learning models to work on the datasets.

VI. Disclosure statement

No potential competing interest was reported by the author(s).

VII. Data Availability Statement

For dataset 1, the participants of this study did not give written consent for their data to be shared publicly, so due to the sensitive nature of the research supporting data is not available. For dataset 2, the data that support the findings of this study are openly available in UCI Machine Learning Repository at https://doi.org/10.24432/C5MC89. For dataset 3, the data that support the findings of this study are openly available in Kaggle at https://doi.org/10.34740/KAGGLE/DSV/7062095.

References

[1] M. Bouazizi and T. Ohtsuki, "Multi-class sentiment analysis on twitter: Classification performance and challenges," *Big Data Mining and Analytics*, vol. 2, no. 3, pp. 181–194, Sep. 2019, doi: 10.26599/BDMA.2019.9020002.

[2] H. Nasiri and S. A. Alavi, "A Novel Framework Based on Deep Learning and ANOVA Feature Selection Method for Diagnosis of COVID-19 Cases from Chest X-Ray Images," *Comput Intell Neurosci*, vol. 2022, 2022, doi: 10.1155/2022/4694567.

[3] B. Krawczyk, C. Bellinger, R. Corizzo, and N. Japkowicz, "Undersampling with Support Vectors for Multi-Class Imbalanced Data Classification," in *Proceedings of the International Joint Conference on Neural Networks*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021. doi: 10.1109/IJCNN52387.2021.9533379.

[4] C. M. Vong and J. Du, "Accurate and efficient sequential ensemble learning for highly imbalanced multi-class data," *Neural Networks*, vol. 128, pp. 268–278, Aug. 2020, doi: 10.1016/j.neunet.2020.05.010.

[5] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 3, pp. 703–715, May 2019, doi: 10.1109/JAS.2019.1911447.

[6] L. A. Sevastianov and E. Y. Shchetinin, "On methods for improving the accuracy of multi-class classification on imbalanced data," 2020. [Online]. Available: http://ceur-ws.org

[7] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," *ACM Computing Surveys*, vol. 52, no. 4. Association for Computing Machinery, Aug. 01, 2019. doi: 10.1145/3343440.

[8] P. Soltanzadeh, M. R. Feizi-Derakhshi, and M. Hashemzadeh, "Addressing the class-imbalance and class-overlap problems by a metaheuristic-based under-sampling approach," *Pattern Recognit*, vol. 143, Nov. 2023, doi: 10.1016/j.patcog.2023.109721.

[9] T. Dönicke, F. Lux, and M. Damaschk, "Multiclass Text Classification on Unbalanced, Sparse and Noisy Data." [Online]. Available: https://opennlp.apache.org/docs/1.8.

[10] Z. Iqbal and M. Yadav, "International Journal of Computer Science and Mobile Computing Multiclass Classification with Iris Dataset using Gaussian Naive Bayes," 2020. [Online]. Available: www.ijcsmc.com

[11] A. Taherkhani, G. Cosma, and T. M. McGinnity, "AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning," *Neurocomputing*, vol. 404, pp. 351–366, Sep. 2020, doi: 10.1016/j.neucom.2020.03.064.

[12] Institute of Electrical and Electronics Engineers, *2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA).*

[13] B. A. Talpur and D. O'sullivan, "Multi-Class Imbalance in Text Classification: A Feature Engineering Approach to Detect Cyberbullying in Twitter," *Informatics*, vol. 7, no. 4, Dec. 2020, doi: 10.3390/informatics7040052.

[14] S. Sridhar and Anusuya, "A Hybrid Approach to Classify the Multiclass Imbalanced Datasets," in *Proceedings of the 5th International Conference on Inventive Research in Computing Applications, ICIRCA 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1084–1089. doi: 10.1109/ICIRCA57980.2023.10220626.

[15] T. Alam, C. F. Ahmed, S. A. Zahin, M. A. H. Khan, and M. T. Islam, "An effective recursive technique for multi-class classification and regression for imbalanced data," *IEEE Access*, vol. 7, pp. 127615–127630, 2019, doi: 10.1109/ACCESS.2019.2939755.

[16] H. Hassan, N. B. Ahmad, and S. Anuar, "Improved students' performance prediction for multi-class imbalanced problems using hybrid and ensemble approach in educational data mining," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jun. 2020. doi: 10.1088/1742-6596/1529/5/052041.

[17] W. Yustanti, N. Iriawan, Irhamah, I. K. D. Nuryana, and A. D. Indriyanti, "A Cross-Sampling Method for Hidden Structure Extraction to Improve Imbalanced Multiclass Classification Accuracy," in *2023 Sixth International Conference on Vocational Education and Electrical Engineering (ICVEE)*, IEEE, Oct. 2023, pp. 353–358. doi: 10.1109/ICVEE59738.2023.10348228.

[18] N. Tepper, E. Goldbraich, N. Zwerdling, G. Kour, A. Anaby-Tavor, and B. Carmeli, "Findings of the Association for Computational Linguistics Balancing via Generation for Multi-Class Text Classification Improvement."

[19] Lipetskiĭ gosudarstvennyĭ tekhnicheskiĭ universitet, Institute of Electrical and Electronics Engineers, IEEE Industrial Electronics Society, and IEEE Industry Applications Society, *Proceedings, 2019 1st International Conference on Control Systems, Mathematical Modelling, Automation and Energy Efficiency*

(SUMMA) : Lipetsk State Technical University, Lipetsk, Russia, November, 20-22 2019.

[20] V. Manoj and T. Devi, "An Effective Approach for Newspaper Article Classification using Multi-Class Support Vector Machine in Comparison with Binary Classifier to improve Accuracy," in *Proceedings of 8th IEEE International Conference on Science, Technology, Engineering and Mathematics, ICONSTEM 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICONSTEM56934.2023.10142872.

[21] S. D. Abdul Bujang *et al.*, "Imbalanced Classification Methods for Student Grade Prediction: A Systematic Literature Review," *IEEE Access*, vol. 11. Institute of Electrical and Electronics Engineers Inc., pp. 1970–1989, 2023. doi: 10.1109/ACCESS.2022.3225404.

[22] I. Sumaiya Thaseen and C. Aswani Kumar, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, Oct. 2017, doi: 10.1016/j.jksuci.2015.12.004.

[23] F. Nurrahman, H. Wijayanto, A. H. Wigena, and N. Nurjanah, "PRE-PROCESSING DATA ON MULTICLASS CLASSIFICATION OF ANEMIA AND IRON DEFICIENCY WITH THE XGBOOST METHOD," *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, vol. 17, no. 2, pp. 0767–0774, Jun. 2023, doi: 10.30598/barekengvol17iss2pp0767-0774.

[24] S. Sharma, Prachi, R. Chhikara, and K. Khanna, "An efficient android malware detection method using BorutaShap algorithm," *International Journal of Experimental Research and Review*, vol. 34, pp. 86–96, 2023, doi: 10.52756/ijerr.2023.v34spl.009.

[25] J. Emakhu *et al.*, "Acute coronary syndrome prediction in emergency care: A machine learning approach," *Comput Methods Programs Biomed*, vol. 225, Oct. 2022, doi: 10.1016/j.cmpb.2022.107080.

[26] R. C. Chen, C. Dewi, S. W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *J Big Data*, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40537-020-00327-4.

[27] I. M. Zubair, Y.-S. Lee, and B. Kim, "A New Permutation-Based Method for Ranking and Selecting Group Features in Multiclass Classification," *Applied Sciences*, vol. 14, no. 8, p. 3156, Apr. 2024, doi: 10.3390/app14083156.

[28] P. Huang *et al.*, "Classification of cervical biopsy images based on LASSO and EL-SVM," *IEEE Access*, vol. 8, pp. 24219–24228, 2020, doi: 10.1109/ACCESS.2020.2970121.

[29] A. Arshad, S. Riaz, and L. Jiao, "Semi-supervised deep fuzzy c-mean clustering for imbalanced multi-class classification," *IEEE Access*, vol. 7, pp. 28100–28112, 2019, doi: 10.1109/ACCESS.2019.2901860.

[30] Y. Mao, V. Gupta, K. Wang, W. K. Liao, A. Choudhary, and A. Agrawal, "To Shuffle or Not to Shuffle: Mini-Batch Shuffling Strategies for Multi-class Imbalanced Classification," in *Proceedings - 2022 International Conference on Computational Science and Computational Intelligence, CSCI 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 298–301. doi: 10.1109/CSCI58124.2022.00057.

[31] N. A. Cloutier and N. Japkowicz, "Fine-tuned generative LLM oversampling can improve performance over traditional techniques on multiclass imbalanced text classification," in *Proceedings - 2023 IEEE International Conference on Big Data, BigData 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 5181–5186. doi: 10.1109/BigData59044.2023.10386772.

[32] M. Janicka, M. Lango, and J. Stefanowski, "Using Information on Class Interrelations to Improve Classification of Multiclass Imbalanced Data: A New Resampling Algorithm," *International Journal of Applied Mathematics and Computer Science*, vol. 29, no. 4, pp. 769–781, Dec. 2019, doi: 10.2478/amcs-2019-0057.

[33] J. Polpinij and K. Namee, "Improving of Imbalanced Data in Multiclass Classification for Sentiment Analysis using Supervised Term Weighting," in *Proceedings - 2021 Research, Invention, and Innovation Congress: Innovation Electricals and Electronics, RI2C 2021*, Institute of Electrical and Electronics Engineers Inc., Sep. 2021, pp. 19–24. doi: 10.1109/RI2C51727.2021.9559797.

[34] A. Glazkova, "A Comparison of Synthetic Oversampling Methods for Multi-class Text Classification," Aug. 2020, [Online]. Available: http://arxiv.org/abs/2008.04636

[35] F. Thabtah, S. Hammoud, F. Kamalov, and A. Gonsalves, "Data imbalance in classification: Experimental evaluation," *Inf Sci (N Y)*, vol. 513, pp. 429–441, Mar. 2020, doi: 10.1016/j.ins.2019.11.004.

[36] R. Ferreira Mello, E. Freitas, F. D. Pereira, ; Luciano Cabral, P. Tedesco, and G. Ramalho, "Education in the age of Generative AI Context and Recent Developments."

[37] M. V. Martins, D. Tolledo, J. Machado, L. M. T. Baptista, and V. Realinho, "Early Prediction of student's Performance in Higher Education: A Case Study," 2021, pp. 166–175. doi: 10.1007/978-3-030-72657-7_16.

[38] J. Chen, J. Lalor, and W. Liu, "Detecting Hypoglycemia Incidents Reported in Patients' Secure Messages: Using Cost-Sensitive Learning and Oversampling to Reduce Data Imbalance Repository Citation." [Online]. Available: https://escholarship.umassmed.edu/oapubs

[39] E. Y. Boateng, J. Otoo, and D. A. Abaye, "Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review," *Journal of Data Analysis and Information Processing*, vol. 08, no. 04, pp. 341–357, 2020, doi: 10.4236/jdaip.2020.84020.

[40] J. Li, A. Sun, J. Han, and C. Li, "A Survey on Deep Learning for Named Entity Recognition," *IEEE Trans Knowl Data Eng*, vol. 34, no. 1, pp. 50–70, Jan. 2022, doi: 10.1109/TKDE.2020.2981314.