A Comparative Analysis of Machine Learning Models for Banking News Extraction by Multiclass Classification with Imbalanced Datasets of Financial News: Challenges and Solutions

Varun Dogra[1], Sahil Verma[2], Kavita[2], NZ Jhanjhi[3], Uttam Ghosh[4], Dac-Nhuong Le[5,6*]

[1]School of Computer Science and Engineering, Lovely Professional University, India
[2]Department of Computer Science and Engineering, Chandigarh University, Mohali, India
[3]School of Computer Science and Engineering, Taylor's University, Malaysia
[4]Department of Computer Science and Data Science, Meharry School of Applied Computational Sciences, Nashville, TN, USA
[5]School of Computer Science, Duy Tan University, Danang, 550000, Vietnam
[6]Institute of Research and Development, Duy Tan University, Danang, 550000, Vietnam

Abstract

Online portals provide an enormous amount of news articles every day. Over the years, numerous studies have concluded that news events have a significant impact on forecasting and interpreting the movement of stock prices. The creation of a framework for storing news-articles and collecting information for specific domains is an important and untested problem for the Indian stock market. When online news portals produce financial news articles about many subjects simultaneously, finding news articles that are important to the specific domain is nontrivial. A critical component of the aforementioned system should, therefore, include one module for extracting and storing news articles, and another module for classifying these text documents into a specific domain(s). In the current study, we have performed extensive experiments to classify the financial news articles into the predefined four classes Banking, Non-Banking, Governmental, and Global. The idea of multi-class classification was to extract the Banking news and its most correlated news articles from the pool of financial news articles scraped from various web news portals. The news articles divided into the mentioned classes were imbalanced. Imbalance data is a big difficulty with most classifier learning algorithms. However, as recent works suggest, class imbalances are not in themselves a problem, and degradation in performance is often correlated with certain variables relevant to data distribution, such as the existence in noisy and ambiguous instances in the adjacent class boundaries. A variety of solutions to addressing data imbalances have been proposed recently, over-sampling, down-sampling, and ensemble approach. We have presented the various challenges that occur with data imbalances in multiclass classification and solutions in dealing with these challenges. The paper has also shown a comparison of the performances of various machine learning models with imbalanced data and data balances using sampling and ensemble techniques. From the result, it's clear that the performance of Random Forest classifier with data balances using the over-sampling technique SMOTE is best in terms of precision, recall, F-1, and accuracy. From the ensemble classifiers, the Balanced Bagging classifier has shown similar results as of the Random Forest classifier with SMOTE. Random forest classifier's accuracy, however, was 100% and it was 99% with the Balanced Bagging classifier.

Keywords

Class Imbalance, Down-Sampling, Ensemble Approaches, Machine Learning, N-grams, Over-Sampling Techniques, TFIDF.

I. Introduction

In the equity market, stocks or funds belong to the different business sectors. And sector-based news has become an inseparable part of the management of financial assets, with news-driven stock and bond markets explosively growing. Fund managers take advantage of this reality and make use of sector-oriented news to select individual stocks to diversify their investment portfolios to optimize returns. There is no such structured framework available for classifying the news on specific sectors of someone's interest. This problem is increasing by the day, necessitating a system for news classification methodology for specific sectors.

Machine learning (ML) techniques have demonstrated impressive performance in the resolution of real-life classification problems in many different areas such as financial markets [1], medical diagnosis [2], vehicle traffic examination [3], fraud detection [4]. There are plenty of document classification systems in the commercial world.

* Corresponding authors:
E-mail address: ledacnhuong@duytan.edu.vn (Dac-Nhuong Le)

For instance, usually, the news stories are grouped by topics [5], medical images are tagged by disease categories [6], and many products are branded according to categories [7]. Different methods of statistical and machine learning are implemented in text labeling, where one of the predefined labels is automatically assigned to a given item of the unlabeled pool of textual articles.

However, the vast majority of articles on the internet about text classification are binary text classification [8] such as email filtering [9], political preferences [10], sentiment analysis [11], etc. Our real-world problem is in most cases much more complex than the binary classification. More formally, if some d is a document in the whole set of documents D and C is the set of all categories i.e. $C = \{c_1, c_2, c_3, \dots, c_n,\}$ the classification of text assigns one category c_i to the document d. Such a classification function with more than two classes is known as multiclass classification; for example, identify a set of news categories as business, political, economic or entertainment..

In our paper, we're interested in isolating news on the banking sector and its most associated domains from the pool of financial news articles. We feel that 'banking news' of any nation is most correlated with their 'governmental news-events ' that covers news on government initiatives for good governance, state or national elections, change or new development of governmental policies, and 'global' financial news that covers global trade, changes in currency-commodities prices, and global sentiments. So, we have a 4-class classification problem of a set of news articles to extract banking, and its most correlated news i.e. Government, and Global from entire financial news articles. We decide to label the news articles into banking, governmental, global, and non-banking classes with a total of 10000 instances. The non-banking news covers all the financial news scrapped from various new portals divergent from these 3 categories (banking, governmental and global). The news reports on different categories are usually imbalanced. The distribution of the news articles in our dataset is shown in Fig. 1. The news articles are manually labeled into these four classes. [12] mentions that labeling is normally done manually by human experts (or users), which is a time-consuming and labor-intensive process but it results in higher accuracy due to expert knowledge being involved in labeling text articles with appropriate. In the process, we label a set of representative news articles for each class. The labelers are experts in the financial domain and financial markets. A team of three experts is used to perform feature selection to identify important or representative words for each class used in a 4-class classification, followed by inspecting each text document and label it to the respective class based on representative words for each class. An agreement is made with the experts to label the given instances of the news articles. The process is used to derive a set of documents from entire unlabeled documents for each class to form the initial training package. The different machine learning techniques are then applied to build and compare the classifiers. The whole process is explained in the later part of the paper in sections 3-4.

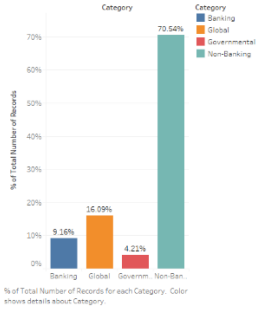

Fig. 1. The distribution of news article instances amongst 4-classes.

A. Multiclass Classification

For machine learning, the problem of classifying instances into three or more classes in multiclass classification. Although some classification algorithms of course allow the use of more than two classes, some are by definition binary algorithms; however, a variety of strategies may transform these into multi-classification. In a multiclass classification problem, some classes may be represented with only a few samples (called the minority class), and the rest falls into the other class (called the majority class). The data disparity in machine learning creates difficulties in conducting data analytics in virtually all fields of real-world problems. The problem of classifying textual news articles is a two-step process. In our experiment, in the first step, the documents are collected from various websites like Bloomberg, Financial Express, and Moneycontrol using web scrapping code written in Python. It is followed by partitioned news articles into their respective category of banking, non-banking, global, and governmental using manual labeling. In the next step, the news articles are trained and tested using machine learning approaches to achieve the classification goal for a new sample of news articles. A comparative analysis is performed based on the results of the experiment to rate the tested machine learning algorithms in descending order so they can be used to evaluate news classification tasks with imbalanced datasets. We are not detailing the process of downloading news from the various sources in the paper.

In turn, multiclass classification can be divided into three groups:

- Native classifiers: These include most common classifiers such as Support Vector Machines (SVM), Classification and Regression Trees (CART), KNN, Naïve Bayes (NB), and multi-layer output nodes i.e. Neural Nets.
- Multi-class wrappers: These hybrid classifiers reduce the problem to smaller chunks that can then be solved with different binary classifiers.
- Hierarchical Classifiers: Using a tree-based architecture this group uses hierarchical methods to partition output space into target class nodes.

B. Learning from Imbalanced Dataset

A dataset is considered class-imbalanced if the number of examples that represent each class is not equal. Dealing with an imbalanced dataset has been a popular subject in the research study of classifying news articles. The conventional machine learning algorithms may introduce biases while dealing with imbalanced datasets [1]. The accuracy of many classification algorithms is considered to suffer from imbalances in the data (i.e. when the distribution of the examples is significantly distorted across classes) [13]. Most binary text classification applications are of this kind, with the negative examples far outnumbered positive examples of the class of interest [2]. Many classifiers assume that examples are evenly distributed among classes and assume an equal cost of misclassification. For example, someone works in an organization and is asked to create a model that predicts whether news belongs to class A, based on the distribution of news in classes A and B at your side. He chooses to use his favorite classifier, train it on data, and before he knows it, he gets an accuracy of 95%. Without further testing, he wants to use the model. A couple of days later he underlines the model's uselessness. Indeed, from the time it was used to gather news, the model he created did not find any news belonging to class A. He figures out after some investigations that there is only about 5 percent of the news produced in the pool that belongs to Class A and that the model always responds to Class "B," resulting in 95 percent accuracy. The kind of "guileless" findings that he obtained were due to the imbalanced dataset with which he works. The goal of this paper

is to examine the various methods that can be used with imbalanced groups to tackle classification problems.

In the imbalanced data set, basically with this problem, a classifier's output leans to be partial towards certain classes (majority class) [14]. In Natural Language Processing (NLP) and Machine Learning in general, the problems of imbalanced classification, under which the number of items in each class for a classification process differs extensively and the capacity to generalize on dissimilar data remained critical issues [15]. Most classification data set do not have precisely the same number of instances in each class but a slight variation is often insignificant. There are problems where class inequality is believed to not just normal.

Also, classifiers are typically built to optimize precision, which in the situation of imbalanced training data is not a reasonable metric for determining effectiveness. Therefore, we are presenting the comparison of various machine learning classification techniques which might result in high accuracy even with imbalanced datasets, however, it is worth mentioning certain challenges we find to deal with imbalanced data and evaluating certain measures along with accuracy to evaluate performance. Also, we conduct machine learning on documents to perform multi-classification, where the data sample belongs to one of the multiple categories exactly.

The readers will also come to know the following key points after they have studied this paper:

- Imbalanced classification is the classification issue when the training dataset has an uneven distribution of the classes. As a result, appropriate sampling techniques must be implemented to balance the distribution by taking into consideration the various characteristics and the balanced performance of all of them.
- The class distribution imbalance may vary, but a serious imbalance is more difficult for modeling and may require advanced techniques. It is possible to introduce an efficient hybrid ensemble classifier architecture that incorporates density-based under-sampling or over-sampling and cost-effective methods by examining state-of-the-art solutions using a multi-objective optimization algorithm.
- Most real-world classification problems, such as scam detection, news headlines categorization, and churn prediction, have an imbalanced class distribution. Certain issues should be addressed when constructing multi-class classifiers in the case of class imbalances.

The paper's structure is as follows. In Section 2 we present a review of several current literature methods that handle the classification of imbalanced datasets for text classification. In section 3 we present our framework of classifying news articles along with challenges and possible solutions for the classification of imbalanced datasets. Sections 4 presents the comparative study of different techniques along with the experimental outcomes. Section 5 summarizes the paper and presents the future direction in the area of classification of imbalanced datasets.

II. Literature Review

We will present the necessary review in text classification and imbalanced learning in the subsequent subsections. We also assess the state-of-art research involving both the learning of imbalances and multiclass text classification.

A. Machine Learning for Text Classification

Here, we present the relevant literature work in the area of text classification using approaches to machine learning. Most of the preceding research had effective results using supervised methods of learning [7], [9], [16]. The following sub-sections present the literature work on feature extraction, selection, representation, and classification using learning models.

1. Document Representation

The efficiency of machine learning approaches largely depends on the option of representation of the data on which they would be implemented. For this purpose, most of the practical work in implementing machine learning algorithms runs further into the creation of pre-processing pathways and data conversion that leads to the representation of data that can help efficient machine learning. These representations or attribute development is essential, yet labor-intensive, and illustrates the vulnerability of current learning algorithms: their weakness to isolate and arrange the data discriminatively. However, the goal is clear when it comes to classification; we want to reduce the number of misclassifications upon testing data and overcoming the mentioned challenges in our framework.

Several machine learning implementations within the text field use bag-of-words representation where terms are defined as dimensions with word frequencies corresponding values. Normalized representation of the word frequencies is used by many applications as the dimensional values. One of the significant techniques of describing a document is Bag of Word (BoW). Use the frequency count of every term throughout the text, the BoW is used to form a vector describing document. This method of representation of documents is called a Vector Space Model [17]. However, the relative frequencies of terms often vary widely, which contributes to the differential meaning of the different words in classification applications [18]. With the varying lengths of various text documents, one needs to normalize when measuring distances between them. To solve these issues, term weighting methods are used to assign correct weights to the word for improving text classification efficiency [19]. Term weighting has long been developed in machine learning in the form of term frequency times inverse document frequency i.e. tfidf [20]. [21] suggests techniques to improve the TF-IDF scores to improve the representation of the term spreading between classes. Such practices may be used in various services where bag-of-word-based TF-IDF features are used. Equation (1) is given as:

$$tfidf\left(t_i,d_j\right)=tf\left(t_i,d_j\right)\times\log\left(N\mid N\left(t_i\right)\right) \tag{1}$$

Here, N represents the overall number of documents and $N(t_i)$ denotes the number of documents in which the term t_i occurs in the collection of documents. $tf\left(t_i,d_j\right)$, it represents the number of times term t_i occurs in document d_j. The newer version is mentioned in (2):

$$w_{i,j}=\frac{tfidf\left(t_i,d_j\right)}{\sqrt{\sum_{k=1}^{|T|}tfidf\left(t_k,d_j\right)^2}} \tag{2}$$

|T| represents the unique terms available in the collection of documents,

$$tf\left(t_i,d_j\right)=1+\log\left(n\left(t_i,d_j\right)\right),if\ n\left(t_i,d_j\right)>0,\ otherwise\ 0 \tag{3}$$

The outline in (2) is concerned with the words that belong to document d_j.

The importance of the standard term weighting outlines in (1), (2) is that three basic principles of word frequency distribution have been integrated into a pool of documents.

1. No less important are uncommon terms than a regular terms-*idf* hypothesis.

2. Numerous presences of a word in a text are no less relevant compared to the presumption of a single appearance-*tf*.
3. Long documents are no less necessary for the equivalent amount of term matching than short documents – the assumption of normalization.

The big drawback of this model is that it results in a large sparse matrix, which poses a high-dimensionality problem. The design of such high-dimensional feature spaces is usually inadequate in the number of items to represent adequately. The reduction of dimensionality is therefore a significant problem for a variety of applications. The literature has suggested several methods for the reduction of dimensionality [3], [22], [23]. For such representations, for instance, linear support-vector machines are comparatively effective [24]; whereas other techniques like Decision trees have to be built and modified with attention to allow their proper usage [25]. When a decision tree induction method computes a decision tree that depends very much on arbitrary features of the training examples, and works well only on trained data, but badly on unknown data, the data becomes overfit. There is a way to reduce the chance of overfitting by choosing the perfect subspace for the function at each node [26]. Cross-validation is an important prevention method to tackle overfitting. We segment the data into sub-sets k, called folds, for regular k-fold cross-validation. We then train the algorithm iteratively on folds of k-1, thus using the remainder of the fold as the test set [27].

In several studies, word n-grams were used effectively [21]. N-gram feature sets include the usage of feature selection approaches to obtain correct attributes subsets. Word n-grams contain bag-of-words (BOWs) and word n-grams in higher-order (e.g. bigrams, trigrams). [28] uses modified n-grams by integrating syntactic information on n-gram relationships. In most document classification activities, this n-gram model is implemented, and almost always boosts precision. This is because the n-gram model allows us to take the sequences of terms into account, as opposed to what will require to do just by using single words (unigrams). Looking into the benefits of the n-grams feature selection, in this paper, a rich collection of n-gram features that encompassed several fixed and variable n-gram categories is studied for classifying textual news articles.

2. Feature Selection

The selection of features serves as a crucial technique for reducing input data space dimensionality to minimize the computational cost. It was designed as a natural sub-part of the process of classification for many learning algorithms. Generally, three feature selection methods i.e. filter method, wrapper method, and embedded method achieve the objective of selecting important features. The ultimate goal of feature selection is always to find the collection of the best features out of the entire dataset to obtain improved classification results. Among all of the feature selection methods, information gain, chi-square, and Gini index have been used effectively [18], [29], [30]. These methods have shown promising results for classification [31]. CHI square reflects one of the more traditional feature selection strategies. In statistics, the CHI square test is used to analyze the independence of two instances. The instances, X and Y, are taken as separate if:

$$p(XY)=p(X)p(Y) \quad (4)$$

These two instances result in a particular word and class occurring respectively in the collection of text features. It can be calculated as given in equation (5):

$$Chi2(t,C)=\sum_{t,C\in\{0,1\}}\frac{(N_{t,C}-E_{t,C})^2}{E_{t,C}} \quad (5)$$

Here, N is termed an observed frequency and E is the expected frequency for every term state t and class C. CHI square would be the function of how often the expected value E counts and N counts observed to deviate from one another. A high value of CHI square means that the independence supposition is wrong. If these two instances are related, then the term existence increases the probability of the class existence. This determines the weighted average score for all classes and then chooses the maximum score between all classes. In this paper, as in (6) given by [29], the former method is ideal to globalize the CHI square value for all classes. Here $P(C_i)$ is the likelihood of a class and $Chi^2(t, C_i)$ is the unique Chi^2 value of a term t.

$$Chi^2(t)=\sum_{i=1}^{M}\mathrm{P}(C_i).Chi^2(t,C_i) \quad (6)$$

Another effective method has been used by researchers i.e. Information Gain. This assesses the overall knowledge that the existence or absence of a word allows one to make the right classification judgment for every class [32]. In other words, it can be used in the selection of features by assessing each variable's gain in the target variable sense. The measurement between the two random variables is considered mutual information.

$$\mathrm{IG}(t)=-\sum_{i=1}^{M}\mathrm{P}(c)logP(c)+P(t)\sum_{i=1}^{M}P(c/t)logP(c/t)+P(\bar{t})\sum_{i=1}^{M}P(c/\bar{t})logP(c/\bar{t}) \quad (7)$$

In equation (7), the total classes are represented by M, probability of class c is represented by P(c), the presence and absence of term t are denoted by P(t) and P($\bar{t}$), P(c|t) and P(c|$\bar{t}$) are class c possibilities provided the existence and absence of a term t.

The other filter method which has been effectively used is the Gini Index [20]. In general, it has simpler computations than the other methods. It can be calculated as given in equation (8):

$$\mathrm{GI}(t)=\sum_{i=1}^{M}\mathrm{P}(t/C_i)^2\mathrm{P}(C_i/t)^2 \quad (8)$$

In (8), $P(t/C_i)$ is the likelihood of a term t provided that the class C_i is present. $P(C_i/t)$ is a class C_i probability given the presence of term t.

3. Classification models

Classification is a supervised technique of machine learning wherein the computer algorithm learns from the data it receives as inputs and then uses the experience to classify new data. This data collection may be purely binary or multi-class classification. Types of classification tasks include voice recognition, handwriting recognition, scam detection, news labeling, etc. There has been several machine learning discovered from time to time with different approach and application. One of the models is *Naive Bayes*, simple to build and use for an extremely large volume of data. The classifier Naive Bayes claims that every other feature is unrelated to the inclusion of a specific feature in a class. Even though these characteristics depend on each other or the presence of the other characteristics, each of these properties contributes to the likelihood independently. It can be calculated as given in equation (9) and (10):

$$P(c|x)=\frac{P(x|C)P(c)}{P(x)} \quad (9)$$

$$P(c|x)=P(x_1|c)\times P(x_2|c)\times\ldots\times P(x_n|c)\times P(c) \quad (10)$$

Here c refers to class and x represents inputs. Given the data x, $P(c|x)$ is mentioned as the posterior probability of c, $P(x|c)$ probability of input value x provided hypothesis was true, $P(c)$

represents the prior probability of c, and $P(x)$ is the prior probability of x.

[33] uses Naïve Bayesian classifier along with two feature evaluation metrics to multi-class text datasets i.e. multi-class Odds Ratio (MOR) and Class Discriminating Measure (CDM) to achieve the best feature selecting results. The other k-nearest-neighbors classifier algorithm takes up a lot of labeled points and uses them to know how to classify certain items. It looks at the points nearest to the new point to identify a new point, so whatever label most neighbors have is the new point label. [16] uses the neighbor-weighted *K-nearest neighbor* algorithm achieving significant performance gains in the classification of an imbalanced data set.

The statistical method, *Logistic Regression*, is used for evaluating a data set in which a result is calculated by one or more independent variables. Uses the probability log-odds of an event that is a linear combination of independent or prediction variables. Logistic Regression uses the Sigmoid activation function which results in either 0 or 1. It can be calculated as given in equation (11):

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{11}$$

Here, z represents the input variable.

It is proven superior to other binary classification such as *KNN*, as it also describes quantitatively the factors leading to classification [34]. The goal is to identify the best fit model to explain the relationship between the dichotomous value attribute and a series of independent variables. *Decision Tree* algorithm gives significant results for treating both categorical and numerical data. In the form of classification or regression models, the decision tree builds a tree structure. It splits down a collection of data into smaller and smaller subsets, thus constructing a linked decision tree incrementally. The tree splitting uses Chi-square, Gini-Index, and Information gain methods. A decision tree with improved chi-square feature selection outperforms in terms of recall for multiclass text classification [35].

The various classifiers being studied in the different applications have shown varied results. The authors have been proposed ensemble methods to further improve classification accuracy measures. Ensemble learning is the mechanism by which several models are systematically created and merged to solve a specific computational intelligence problem. *Random forests* are an ensemble learning system for classification, regression, and other functions that operates by creating a multitude of decision trees during training and providing class mode (classification) or mean forecasting (regression) of the individual trees. [36] uses ensemble methods for keyword extraction where Random Forest shows promising results. The authors have been improving such methods for effective text classification [37].

In the current scenario where data has been converting to big data, *Neural Networks* have been the most studied algorithms for text classification. A neural network is a type of layer-organized units (neurons) that transforms some output into an input vector. Every unit can take input, impose a function on it, and pass the output to the next layer. The networks are commonly known as feed-forward: a unit feeds its output to all the units on the next layer but no input is given to the previous layer. Weights are added to the signals that travel from one unit to another, and it is these weights that are adjusted during the training phase to fit a neural network to a specific problem. [38] proposes three distinct frameworks for sharing information with task-specific and shared layers to model text, based on *recurrent neural networks*. These deep learning algorithms' successes depend on their ability to model complex and nonlinear interactions within the data. Finding suitable architectures for these models, however, has been a problem for researchers addressing leveraging.

B. Techniques for dealing with Imbalanced data

We will illustrate in this section the various techniques which have been experienced so far by researchers for training a model to perform well against highly imbalanced data sets. The authors mentioned that where it comes to text classification, the normal distribution of textual data is often unbalanced. To better differentiate documents into minor categories, they used a basic probability-based word weighting scheme to solve the problem [39]. Many real-world text classification tasks, according to the authors, require unbalanced training instances. However, in the text domain, the methods introduced to resolve the imbalanced description have not been consistently tested. They conducted a survey based on the taxonomy of strategies suggested for imbalanced classification, such as resampling and instance weighting, among others [40]. The following sub-sections cover the literature of various techniques used so far to deal the text classification with imbalanced data sets.

1. Data Level technique

Dealing with imbalanced data sets requires techniques such as enhancing classification algorithms or balancing the training data classes until the machine learning algorithm provides the data as input. The primary goal of balancing classes is either to raise the frequency of the minority class or to decrease the frequency of the majority class. This is provided for all classes to get roughly the same number of instances.

- Under-sampling aids in optimizing class allocation by randomly eliminating instances of majority classes. This is achieved when the majority and minority class cases are balanced completely. Evolutionary undersampling outperforms the non-evolutionary models by increasing the degree of imbalance [41]. They describe a performance function that incorporates two values: the classification factor aligned with both the sub-set of training instances and the percentage of reduction associated with the training set of the same sub-set of instances. A novel under-sampling technique is implemented, called cluster-based instance selection, which incorporates clustering analysis with instance selection [42]. The clustering analysis framework groups identical data samples of the majority class dataset into subclasses, while the instance selection framework extracts out unaccountable data samples from each subclass. It is also proven that under-sampling with KNN is the most powerful approach [43].
- Over-sampling raises the amount of minority class instances by arbitrarily replicating them to make the minority class more represented in the study. The author suggests a Random Walk Over-Sampling method by generating synthetic samples by randomly walking from real data to match different class samples [44]. This sampling method is designed to address the imbalanced grouping of data by producing some samples of a synthetic minority class. The synthetic samples, that properly follow the initial minority training set and extend the minority class boundaries, are coupled with the actual samples to make a more efficient full dataset, and the entire is used to build unbiased classifiers. Unfortunately, traditional over-sampling approaches have shown their respective shortcomings, such as causing serious over-generalization or not effectively improving the class imbalance in data space, while facing the more challenging problem as opposed to the binary class imbalance scenario. The author proposes a synthetic minority oversampling algorithm based on k-nearest neighbors (k-NN), called SMOM,

for handling multi-class imbalance problems [20]. SMOM is a method to prevent over-generalization since safer neighboring directions are more likely to be chosen to produce synthetic instances. It is also suggested that combine sampling be rendered by combining the techniques of SMOTE and Tomek with SVM as the method of binary classification [45]. SMOTE is a useful over-sampling technique for increasing the number of positive classes incorporating sample drawing methods by replicating the data randomly so that the number of positive classes is equal to that of the negative class. [46] has performed multiclass classification with equal distribution of the data among various classes using SMOTE, owing to the introduction of synthetic instances which increased the number of training samples to distribute the data equally among 10 different labels. Tomek links method is under-sampling, which works by decreasing negative class numbers. However, in some extreme cases mixing sampling methods are no stronger than utilizing Tomek link methods.

2. *Algorithms-based decomposition techniques*

The technique must first use decomposition strategies to transform the original multi-class data into binary subsets.

- *One-vs-all* is a strategy that requires training N independent binary classifiers, each programmed to identify a specific class. All those N classifiers are collectively used to classify multiple classes. With multi-class imbalanced data, an algorithm called One-vs-All with Data Balancing (OAA-DB) is built to enhance the classification performance [47]. It is mentioned that the OAA-DB algorithm can boost classification efficiency for imbalanced multi-class data without decreasing the overall classification accuracy. In other words, for every class, One-vs-All trains a single classifier, treating the existing class as the minority one and the remaining classes as a majority.
- *One-vs-One* trains a binary classifier for each potential pair of classes, ignoring examples that are not part of the pair classes. To resolve the multi-class imbalance classification problems, an exhaustive empirical study is proposed to investigate the possibility of improving the one-vs-one scheme through the application of binary ensemble learning approaches [48].
- *One-Against-Higher-Order* (OAHO) is an explicitly designed decomposition process for unequaled sets of data. OAHO first divides class by decreasing the number of samples [49]. OAHO sequentially marks the current class as 'positive class' and all the remaining classes with lower ranks as 'negative classes,' then trains a binary classifier.
- *All-in-One* uses One-vs-All along with One-vs-One, it first uses One-vs-All sub-classifiers to find the top two most probable categories for each test case, and then use the corresponding One-Vs-One sub-classified to decide the final result [50].

3. *Algorithms-based Ensemble techniques*

The main purpose of the ensemble methodology is to improve single classifier efficiency. The method involves constructing from the original data numerous two-stage classifiers and then aggregating their predictions.

a) Boosting-based techniques

One strategy which can be used to increase classification efficiency is boosting. Although several data sampling techniques are explicitly developed to fix the issue of class imbalance, boosting is a technique that can increase the efficiency of any weak classifier. *Ada Boost* iteratively constructs a model ensemble, which is an adaptive boosting strategy that combines many weak and inaccurate rules to build a predictive rule that is highly effective. During each iteration, case weights are changed to properly classify the instances in the next iteration that were wrongly classified during the current iteration. Upon completion, all models developed to take part in a weighted vote to identify unlabeled cases. Such a strategy is especially useful when grappling with class inequality as in successive implementations the minority class instances are more likely to be misclassified and thus assigned larger weights. In other words, it's a binary classification algorithm that combines many weak classifiers to create a stronger classifier [4]. Boosting can be achieved either by "reweighing" or "resampling". At each step, the changed example weights are transferred directly to the base learner while boosting by reweighing. Not all learning algorithms are designed to integrate example weights into their decision-making systems, however. This is a class that uses the AdaBoost Ml method to boost a nominal classifier which can only address nominal class problems. It is given in equation (12):

$$f(x) = sign(\sum_{m=1}^{M} \theta_m f_m(x)) \quad (12)$$

Here, $f(x)$ represents m^{th} weak classifier and θ_m is the corresponding weight.

This often improves the performance dramatically but sometimes overfits [51]. *Gradient boosting* is an approach that generates a set of weak regression trees by introducing iteratively a new one which further strengthens the learning goal by optimizing an arbitrary differentiable loss function [52]. Gradient Boosting builds the first learner to predict the samples on the training dataset and calculates the loss. And use that loss in the second stage to build an improved learner. The recent implementation of this boosting method called *XGBoost* combines the principles of computational efficiency. The paper presents a scalable end-to-end tree boosting system *XGBoost* that is widely used by data scientists to perform state-of-the-art machine learning outcomes [52].

b) Bagging-based techniques

Bootstrap aggregation, also known as *bagging*, is an ensemble meta-algorithm for machine learning that aims to enhance the stability and accuracy of classification algorithms. The standard algorithm requires the development of specific bootstrap training items, 'n' with substitution. Then train the algorithm on each bootstrapped algorithm separately, then aggregate the forecasts at the end. The authors present online bagging and boosting versions that require only one pass through the training data [53]. Random Forests is an ensemble classifier composed of several decision trees and generating the class which is the class output mode for individual trees. In this way, an RF ensemble classifier works more than a single tree from the classification results perspective [54]. The authors suggested ensemble classifiers focused on original principles such as learning cluster boundaries by the base classifiers and mapping cluster confidences to a class decision using a fusion classification [55]. The classified data set is divided into several clusters and is fed into several distinctive base classifiers. Cluster boundaries are identified to base classifiers and cluster confidence vectors are built. A second stage fusion classifier blends class decisions with confidences and maps of the clusters. This ensemble classifier restructured the learning environment for the base classifiers and promoted successful learning.

4. *Other Techniques*

Despite their effectiveness, however, sampling methods add complexity and the selection of required parameters. To address these problems, the author suggests a modern decision tree strategy named *Hellinger Distance Decision Trees* (HDDT), which allows the use of distance from Hellinger as the criteria for splitting. For

probability and statistics, the Hellinger distance is used to measure the correlation of two distributions of probabilities. The authors use a Hellinger weighted ensemble of HDDTs to combat definition drift and improve the accuracy of single classifiers [56].

Error Correcting Output codes, ECOC is a common multi-class learning tool that works by breaking down the multi-class task into a set of binary class subtasks (dichotomies) and creating a binary classifier from each dichotomy. Both the dichotomy classifiers evaluate a test instance and then assign it to the nearest class in code space. A suitable code matrix, an effective learning strategy, and a decoding strategy highlighting minority classes are needed to enable ECOC to tackle multi-class imbalances. The authors propose the imECOC approach that operates on dichotomies to deal with both the imbalance between class and the imbalance within a class [57]. ImECOC assigns dichotomy weights and uses weighted decoding distances where optimum dichotomy weights are derived through reducing weighted loss in terms of minority classes.

The authors suggest merging weighted One-vs-One voting with a Winnow dynamic combiner customized to the program for the data stream. This will allow weights for classifiers to be dynamically modified, boosting the power of those competent in the current state of the stream [17]. *DOVO* simply adjusts the weights for classified objects returned via an active learning approach that enables even more consistent weights and lower processing costs. From those in the perspective of operation recognition, each action shall be taken over a given period. The proposed weighting procedure thereby enables to rapidly increase the significance of qualified classifiers to identify this particular behavior immediately after it has been identified by the active learning methodology and to sustain the significant importance of these related classifiers throughout its length.

C. Existing solutions or software for classification with imbalanced datasets

A program, *KEEL* [58], provides a customized algorithm for the problem of classification with class imbalances. *Multi-IM* draws its basis from the probabilistic relational methodology (PRMsIM), developed to learn from imbalanced data for the problem of two categories [59]. *Imbalanced-learn*; A Python toolbox for resolving imbalanced results [60].

We use the following framework to evaluate the accuracy output of various ML algorithms and to validate our implementations in the classification of multi-class imbalance data on financial news datasets.

III. Framework and Working of Financial News Classification System: Challenges and Solutions of Data Imbalances

Text classification is crucial for information extraction and summarization, text retrieval, and question-answering in general. Using machine learning algorithms, the authors demonstrated the text classification process [19]. Following the approach, we developed a structure shown in Fig. 2. to distinguish the banking and other related sector-oriented news items from financial news posts. It involves three stages, including the data pre-processing phase, the training phase of the classifiers, and a comparative estimation of the performance phase of the classifiers. The phases are discussed in brief in the sub-sections along with certain challenges and solutions are given by researchers.

However, when faced with imbalanced multi-class results, we can drop output on one class quickly when attempting to get output on another class. A clearer analysis of the essence of the issue of class imbalance is required, as one should recognize in what realms class imbalance most impedes the output of traditional multi-class classifiers while developing a system suitable to this topic. Although most of the problems addressed in the preceded section can be applied to these multi-class concerns, the banking and other related news extraction from the financial news domain. We are identifying the following vital research directions for the future.

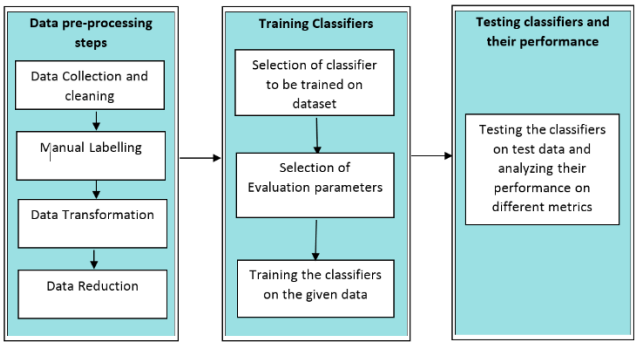

Fig. 2. Multiclass classification of Financial News.

A. Data Pre-processing

Data preprocessing is a method used to transform the raw data into an effective and functional format. Effective pre-processing of text data is critical to achieving an appropriate output and better text classification quality [61].

Challenge-A: The task of preprocessing data here may be much more critical than in the case of binary issues. Possible difficulties can be easily identified: class overlap can occur in more than two classes, class label noise can influence the issue, and class boundaries may not be specific. Therefore, effective data cleaning and sampling techniques must be implemented to take into consideration the various characteristics of the classes and the balanced performance of all of them [62].

Solution-1: The problem of noise present in the data in the case of imbalanced distributions is incredibly difficult. Distortions may dramatically deteriorate classifier efficiency, particularly in the case of minority examples. New data cleaning methods need to be used to manage the existence of overlapping and chaotic samples which can also lead to worsening efficiency of the classifier. We might conceive projections into different spaces where the overlap is alleviated or basic examples are eliminated as mentioned in 3.1.3. However, measures are needed to assess whether a provided overlapping example can be excluded without discrimination to one class. A study of the effect on the real imbalance between classes is quite important in the case of label noise. Measures are therefore required to determine whether a given overlapping example can be discarded without compromising one of the classes. False labeling may lead to increasing the imbalance or disguise actual disproportions. This situation is handled with sustained methods for sensing and filtering noise, as well as handling and relabeling strategies for such examples as mentioned in 1.

Solution-2: Analysis of the kind of examples found in each class and their connections with other classes is interesting. Measuring each sample's difficulty here isn't straightforward, as it may adjust to various classes. For instance, for classes Banking and Governmental, news related to a collective decision on negative GDP outlook and modification on repo rate by RBI may be of borderline type while at the same time being a safe example when considering the remaining classes. Therefore, we have preferred a more flexible classification i.e. SMOTE. SMOTE functions by choosing similar examples in the vector

space, drawing a line through the examples in the vector space, and drawing a new example at a point in the line.

Solution-3: New sampling approaches are needed for issues of multiple classes. Simple re-balancing is not a proper approach towards the largest or smallest class. We need to establish precise methods for adapting the sampling procedures to both the individual class property and their mutual relationships. [6] has provided the ensemble methods to deal with class imbalance classification, ADASYNBagging, and RSYNBagging. The ADASYN and RSYN were based on over-sampling and under-sampling techniques respectively. These were combined with a bagging algorithm to integrate the advantages of both algorithms. Another paper has provided a hybrid model to get a random sample from an unknown population. When compared with a random sample, a non-random sample could not provide better representative inferential statistics. Hence, to overcome this problem, Snoran Sampling Method was developed by [63]. We have not implemented these techniques in our paper. What sampling strategies would function best with the learning of the ensemble to boost class inequality, however, is highly dependent on problem domains.

1. Data Collection

To continue this, we gathered data by *scrapping* news from public news sources such as Bloomberg, Financial Express, Money Control, and Times of India using python-written code. As a result, we have been collected more than 10000 instances of financial news articles from the year 2017 to 2020. The news articles belong to different sectors or market segments. These are then pre-processed such that the machine learning algorithms may learn from the training dataset and adapt them in an acceptable way to the testing data collection. Therefore, these are pre-processed for the machine learning models to be explored from the training sample and implemented in an appropriate format to the test data set.

2. Labeling

The first step in the pre-processing phase is to *label* the news from 4 classes to which they belong to the specific sector. 4-classes are named as Banking, Global, Governmental, and Non-Banking. We prefer manual labeling [64] of the news articles with the help of experts of the financial domain where overlapping examples were preferred to discard without damaging one of the classes. Table I mentions the instance of each class as follows:

TABLE I. SAMPLE OF NEWS ARTICLES FROM DIFFERENT SOURCES AND CLASSES

Source	News article	Class
Source1[1]	The Kolkata-based private sector lender Bandhan Bank surpassed the market capitalization of all listed PSU banks except State Bank of India upon blockbuster stock market debut on Tuesday after floating India's biggest bank IPO earlier this month.	Banking
Source2[2]	For India, the current account deficit is within the comfort zone although it has widened and the GDP growth is heading towards 7.5-7.7 percent.	Governmental
Source3[3]	The U.S. Federal Reserve has cut its benchmark interest rate by a half-point-the biggest reduction, and the first outside of scheduled meetings since the 2008 crisis year.	Global
Source4[4]	The Nifty50 formed a bearish candle for the sixth consecutive day in a row and analysts feel that it will be hard for the index to breach the 200-DEMA in a hurry.	Non-Banking

[1] www.financialexpress.com
[2] www.moneycontrol.com
[3] www.bloombergquint.com
[4] www.moneycontrol.com

3. Data Cleaning

They are then *cleaned* because the data can have several sections that are insignificant and missing. Data cleaning is done to handle that portion. It includes absent managing data, noisy data, etc. It helps the machine learning algorithms to efficiently grasp and operate on them.

4. Data Transformation

The next step, *data transformation*, is taken to turn the data into appropriate forms suited to the mining process, and the text of news articles therein is converted into measures with quantitative values by constructing a vector *set of features.* Since data mining is a technique used for managing enormous quantities of data. In these instances, research became harder when operating with a huge volume of data. To get rid of that, we use the strategy of *data reduction*. This seeks to increase the capacity of storage and reduce the expense of data collection and analysis. In other words, in the last step of this stage, the feature vector is normalized and scaled to prevent an *unbalanced dataset*.

B. Training classifiers

Training is the practice of having text that is considered to belong to the defined classes and creating a classifier based on that known text. The basic concept is that the classifier accepts a collection of *training data* describing established instances of classes and uses the information obtained from the training data to determine the classes other unknown content belongs to, by conducting statistical analysis of training data. We can also use the classifier to derive information on your data based on the statistical analysis carried out during the training process. First, we identify the classes on a collection of training data, and then the classifier uses these classes to evaluate and decide the classification of other data. When the classifier assesses the data, it uses two often contradictory metrics to help decide if the content found in the new data belongs in or outside a class*. Precision*, is the likelihood that what has been labeled as being is actually in that class. High precision may come at the cost of missing certain results whose terms match those of other outcomes in other groups. *Recall*, the likelihood that an object is listed as being in that class in fact in a class. High recall may come at the cost of integrating outcomes from other classes whose terms match those of target class results. We need to find the right balance with high precision and high recall while we are tuning our classifier. The balance focuses on what our priorities and criteria are for implementation. We need to train the classifier with *sample data* that describes members of all the classes to find the best thresholds for our data. Finding good training samples is very critical because the nature of the training can directly influence the quality of the classification. The samples should be statistically valid for each class and should include samples that include both solid class examples and samples near the class boundary.

Challenge-B: The strong potential resides in the complexity of multi-class, distort-insensitive classifiers. They will permit multi-class complications to be handled without referring to strategies for resampling when algorithm-level approaches are used to counter class imbalances. So one may wonder if other prominent classifiers can be adapted in this case [62].

Solution-1: Certain issues should be addressed when constructing multi-level classifiers in the case of class imbalances. A broader study

is required of how numerous unbalanced data sets influence decision boundaries in classifiers. Based on [65] Hellinger distance has proved useful in cases of class imbalance. Since accuracy may offer a distorted picture of success on unbalanced data, current stream classifiers are focused on accuracy that is hampered by minority class output on unbalanced streams, resulting in low recall levels of minority classes. A split based on Hellinger Distance will give a high score to a split separating the classes in the best way relative to the parent population. When utilizing Hellinger, it is possible to obtain a statistically relevant change in the recall level on imbalanced data sources, with a reasonable rise in the false positive rate.

Solution-2: Other solutions with potential robustness to the imbalance, such as methods based on density, need to be explored. [66] have provided a more thorough review of the cluster oversampling based on density and in terms of density-dependent clustering under-sampling techniques. Their findings suggest the strategy will boost the classifier's predictive efficiency. It also yields the best in the precision average.

Solution-3: While modern methods of learning with imbalances are suggested to tackle the question of data imbalances, they have certain limitations; under-sampling methods lose essential details, and cost-sensitive methods are prone to outliers and noise. [67] has provided an efficient hybrid ensemble classifier architecture incorporating density-based under-sampling and cost-effective approaches by investigating state-of-the-art solutions using an algorithm for multi-objective optimization. First, they developed a density-based under-sampling method to select informative samples with probability-based data transformation from the original training data, which enables multiple subsets to be obtained following a balanced class-wide distribution. Second, they have used the cost-sensitive approach of classification to address the problem of information incompleteness by modifying weights in minority groups misclassified, rather than majority samples. Finally, they implemented a multi-objective optimization method and used sample-to-sample relations to auto-modify the classification outcome utilizing an ensemble classification system [68-80].

C. Testing Classifiers and their performance

We run the trained classifier on unknown news articles to check a classifier to decide which classes each news article belongs to. The goal of this stage is to check the performance of the classifiers on the training set and to see if they detect the training correctly. The classifiers considered will be graded according to their effectiveness in detecting the appropriate class. In the later section, we will test various classifiers on the unseen news articles and compare the performance of each.

IV. Working of Financial news classification system

Throughout this section, we describe first the experimental method used to train the classifiers and then demonstrate their success in the classification of news articles into four separate classes. It should be noted here that most text classification algorithms are prone to the form and design of the dataset, depending on factors such as class size, class disparity (number of samples per class), feature scaling, number of training samples, number of features, etc. Besides, different algorithms follow different approaches to solving problems of multi-class classification which also affects their performance. So, we have faced some challenges and, to address these challenges, we have made sure that the available data from which each classifier will learn is distributed equally for each class.

A. Experimental set up to train the classifiers

We used the Tableau prep tool for the data cleaning and preprocessing operations, while the desktop tool was used for the data visualization. The classification tests were performed on Python 3.8 utilizing numerous Python-supported libraries to incorporate machine learning and deep learning algorithms. With a split of 75% and 25% respectively, the total of 10,000 news articles is divided into training and test data. The news articles are related to 4 different classes as mentioned in the introductory section. The data was imbalanced. So, to balance the data various sampling techniques were used. As stated in the introductory section, the news articles are linked to 4 different classes. In nature, the data had been imbalanced. Therefore, different sampling strategies were used to balance the data among classes as discussed in section 4.2. The machine and deep algorithms were further implemented on data for classification using scikit-learn and imblearn libraries of Python. Scikit-learn offers a package named the TfidfVectorizer for the extraction of functionality from text documents. This class is responsible for both vectorizing text documents (news articles) into vectors of word features and transforming them of the term vectors in the scores of TfIdf. We also vectorized the dataset during the experiments using the N-gram approach, with unigrams, bigrams, and tri-grams.

B. Results and Discussion

We have carried out several experiments on our pre-processed data collection utilizing conventional machine learning algorithms detailed in the section preceding. The key purpose of these experiments is to determine the right classifier that gives the best performance. Every classifier's output concerning classification is calculated using the metrics Precision, Recall, and F_1-score. The accuracies are obtained with both train-test split and 5-fold cross-validation for all classifiers. The outcomes of the chosen classifiers are described in the sub-sections that follow.

For the traditional machine learning algorithms, TF-IDF features of 1-gram, 2-gram, and 3-gram were used. The detailed experiments on the financial news datasets were carried out.

1. Results of multiclass classification with data-imbalances

Table II lists the results of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with data imbalances across classes. From the different classifiers Decision Tree {criterion='gini' to measure qualility of split, splitter='best', max_depth=2 for maximum depth of tree, random_state=1 is the seed for random number generator}, Linear SVC {C=1 regularization parameter, multi_class='y'}, Logistic Regression {C=1 regularization parameter, random_state=0}, Multinomial Naïve Bayes {alpha=1.0 smoothing parameter}, Random Forest {n_estmators=100 for number of trees, random_state=1 will always produce same results with same parameters and training data, max_depth=3 for maximum depth of the tree}, and Multilayer Perceptron {solver='lbfgs' for weight optimization, alpha=0.0001 L2 penality, learning_rate='constant', hidden_layer_sizes=(5,2), random_state=1}, Random Forest performed best with accuracy 88% as shown in Table III. The Random Forest achieved the F_1-score 0.96, 0.71, 0.92, 0.38 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is visualized in Fig. 3-6. Table III shows that the accuracy comes out to be 78%-88% range for all machine learning algorithms with train-test split and cross-validation.

TABLE II. RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH DATA IMBALANCES

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Decision Tree	1, 2, 3	0.93	0.96	0.94	0.76	0.68	0.72	0.90	0.95	0.92	0.80	0.31	0.44
Linear SVC	1, 2, 3	1.00	0.77	0.87	0.86	0.61	0.71	0.86	0.98	0.91	1.00	0.38	0.56
Logistic Regression	1, 2, 3	1.00	0.31	0.47	0.88	0.34	0.49	0.76	0.99	0.86	0.00	0.00	0.00
Multinomial NB	1, 2, 3	0.86	0.23	0.36	0.73	0.54	0.62	0.79	0.97	0.87	0.00	0.00	0.00
Random Forest	1, 2, 3	0.93	1.00	0.96	0.89	0.59	0.71	0.88	0.98	0.92	1.00	0.23	0.38
Multi-layer Perceptron	1, 2, 3	1.00	0.69	0.82	0.78	0.68	0.73	0.86	0.96	0.91	1.00	0.31	0.47

TABLE III. ACCURACY OF THE CLASSIFIERS WITH IMBALANCED DATA

Classifier	Accuracy(Train/Test)	Cross-Validation
Decision Tree	0.87	0.87
Linear SVC	0.87	0.87
Logistic Regression	0.78	0.78
Multinomial NB	0.78	0.78
Random Forest	0.88	0.88
Multi-layer Perceptron	0.87	0.87

However, Table II shows that the recall of the minority classes is very less. The Logistic Regression and Multinomial NB has shown 0% precision and recall for the minority class i.e. Governmental. This is visualized in Fig. 5. At the same time, the precision and recall for the other classes have shown high precision and recall. It shows that machine learning models are more biased towards the majority class. So, we need to apply imbalanced data handling techniques.

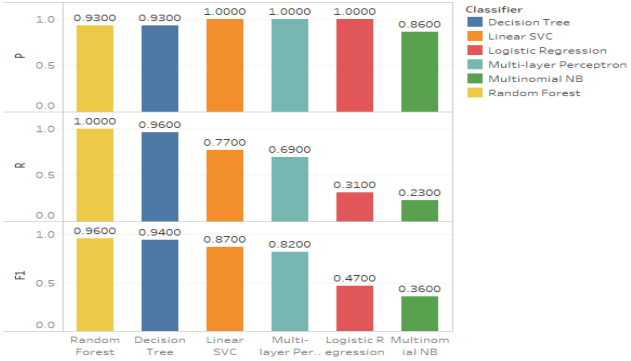

Fig. 3. Performance metrics P, R, F-1 for various classifiers for Banking Class.

Fig. 4. Performance metrics P, R, F-1 for various classifiers for Global Class.

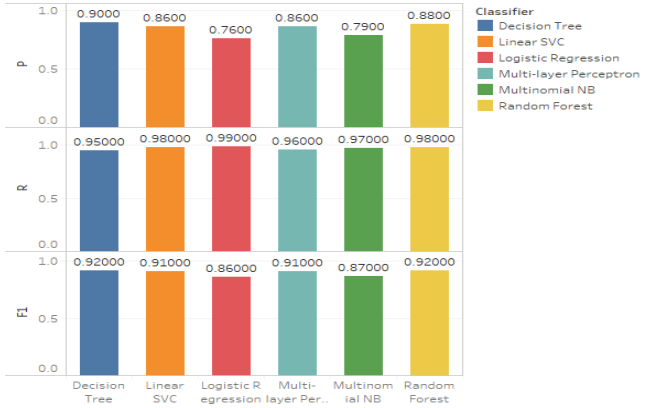

Fig. 5. Performance metrics P, R, F-1 for various classifiers for Non-Banking Class.

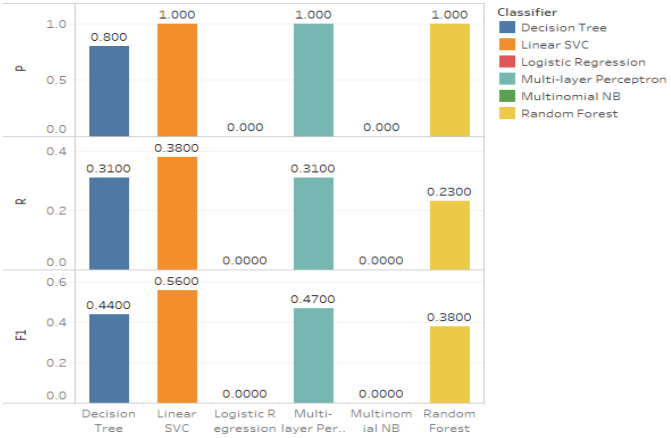

Fig. 6. Performance metrics P, R, F-1 for various classifiers for Governmental Class.

2. *Results of multiclass classification with data balance using Data-level technique: Random Over-Sampling with replacement*

The Resampling takes place with the exclusion of the minority class, increasing the sample number to equal that of the majority class. Tables IV and V lists the results of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with data balanced using random over-sampling technique across classes.

TABLE IV. RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH BALANCED DATA USING UP-SAMPLING

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Decision Tree	1, 2, 3	0.99	1.00	0.99	0.95	0.99	0.97	0.99	0.94	0.97	0.99	1.00	1.00
Linear SVC	1, 2, 3	0.98	1.00	0.99	0.98	0.99	0.98	0.99	0.97	0.98	0.99	1.00	1.00
Logistic Regression	1, 2, 3	0.98	1.00	0.99	0.93	1.00	0.96	1.00	0.91	0.95	0.99	1.00	1.00
Multinomial NB	1, 2, 3	0.97	0.96	0.97	0.93	0.97	0.95	0.93	0.83	0.88	0.94	1.00	0.97
Random Forest	1, 2, 3	0.99	1.00	1.00	0.98	0.99	0.98	0.99	0.98	0.98	1.00	1.00	1.00
Multi-layer Perceptron	1, 2, 3	0.99	1.00	0.99	0.94	0.99	0.97	0.99	0.93	0.96	1.00	1.00	1.00

Table V: Accuracy of the classifiers with balanced data using Up-sampling

Classifier	Accuracy(Train/Test)	Cross-Validation
Decision Tree	0.98	0.983
Linear SVC	0.98	0.982
Logistic Regression	0.98	0.975
Multinomial NB	0.94	0.948
Random Forest	0.99	0.996
Multi-layer Perceptron	0.98	0.985

From the different classifiers Decision Tree, Linear SVC, Logistic Regression, Multinomial Naïve Bayes, Random Forest, and Multilayer Perceptron with accuracy for all classes with balanced datasets using up-sampling, the Random Forest again performed best with accuracy 99% as shown in Table V. The Random Forest achieved the F_1-score 1.00, 0.98, 0.98, 1.00 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is shown in Tables IV and V. It is observed that with data balances the precision and recall have also improved for every classifier. And the accuracy of the classifiers varies between 94% to 100% and it is visualized in Fig. 7.

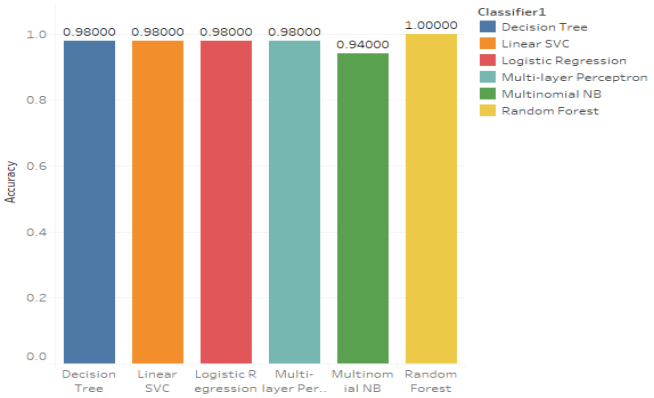

Fig. 7. Accuracy for various classifiers with balanced classes using Up-Sampling.

3. Results of multiclass classification with data balance using Data-level *technique: Random Down-Sampling without replacement.*

This is done by resampling the majority class without replacement, setting the number of samples corresponding to that of the minority class. Table VI, VII lists the results of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with data balanced using down-sampling technique across classes.

From the different classifiers Decision Tree, Linear SVC, Logistic Regression, Multinomial Naïve Bayes, Random Forest, and Multilayer Perceptron with accuracy for all classes with balanced datasets using down-sampling, the Random Forest again performed best with an accuracy of 80% as shown in Table VII.

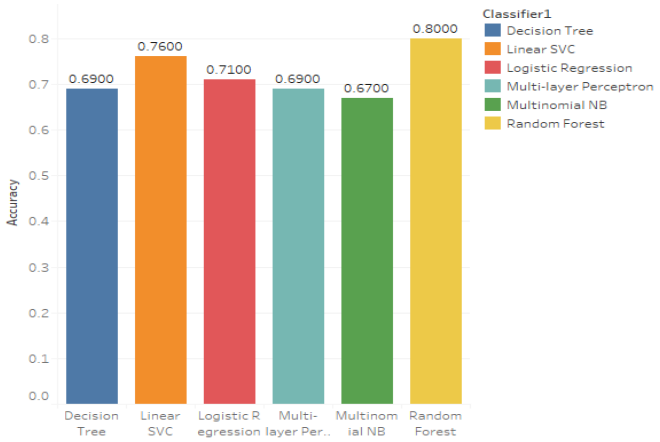

Fig. 8. Accuracy for various classifiers with balanced classes using Down-Sampling.

The Random Forest achieved the F_1-score 0.95, 0.83, 0.73, 0.70 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is shown in Tables VI and VII. The accuracy of the classifiers has degraded with data balances using down-sampling as compared to up-sampling. And the accuracy of the classifiers varies between 67% to 80% and it is visualized in Fig. 8.

4. *Results of multiclass classification with data balance using Data-level technique: hybrid over-sampling technique SMOTE*

SMOTE helps to balance the representation of the classes by replicating randomly through minority class examples. SMOTE synthesizes new instances within existing instances of minority classes. This produces the virtual train records by linear interpolation for the minority class. For each case, these synthetic training records are created by a random selection of one or more k-nearest neighbors in the minority class. The data is reconstructed after the oversampling process, and the classification models are implemented for the processing data. Table VIII lists the results of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with

TABLE VI: RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH BALANCED DATA USING DOWN-SAMPLING

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Decision Tree	1, 2, 3	1.00	0.91	0.95	0.56	0.83	0.67	0.71	0.38	0.50	0.60	0.67	0.63
Linear SVC	1, 2, 3	1.00	0.91	0.95	0.75	0.80	0.77	0.78	0.67	0.72	0.67	0.67	0.67
Logistic Regression	1, 2, 3	1.00	0.82	0.90	0.69	0.75	0.72	0.70	0.54	0.61	0.54	0.78	0.64
Multinomial NB	1, 2, 3	0.82	0.82	0.82	0.69	0.75	0.72	0.67	0.31	0.42	0.53	0.89	0.67
Random Forest	1, 2, 3	1.00	0.91	0.95	0.83	0.83	0.83	0.89	0.62	0.73	0.57	0.89	0.70
Multi-layer Perceptron	1, 2, 3	0.88	0.64	0.74	0.71	0.83	0.77	0.62	0.62	0.62	0.60	0.67	0.63

TABLE VII: ACCURACY OF THE CLASSIFIERS WITH BALANCED DATA USING DOWN-SAMPLING

Classifier	Accuracy(Train/Test)	Cross-Validation
Decision Tree	0.69	0.695
Linear SVC	0.76	0.764
Logistic Regression	0.71	0.720
Multinomial NB	0.67	0.750
Random Forest	0.80	0.803
Multi-layer Perceptron	0.69	0.692

TABLE VIII: RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH BALANCED DATA USING SMOTE

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Decision Tree	1, 2, 3	0.99	0.93	0.96	0.99	1.00	1.00	0.95	0.99	0.97	0.98	1.00	0.99
Linear SVC	1, 2, 3	0.99	0.92	0.96	0.98	1.00	0.99	0.94	0.99	0.97	0.99	1.00	1.00
Logistic Regression	1, 2, 3	1.00	0.91	0.95	0.98	1.00	0.99	0.93	1.00	0.96	0.99	1.00	1.00
Multinomial NB	1, 2, 3	0.92	0.85	0.88	0.97	0.96	0.97	0.94	0.96	0.95	0.94	1.00	0.97
Random Forest	1, 2, 3	0.99	0.99	0.99	0.99	1.00	1.00	0.99	0.99	0.99	1.00	1.00	1.00
Multi-layer Perceptron	1, 2, 3	0.99	0.93	0.96	0.99	1.00	0.99	0.94	0.99	0.97	1.00	1.00	1.00

TABLE IX: ACCURACY OF CLASSIFIERS WITH BALANCED DATA USING SMOTE UP-SAMPLING

Classifier	Accuracy(Train/Test)	Cross-Validation
Decision Tree	0.98	0.981
Linear SVC	0.98	0.948
Logistic Regression	0.98	0.972
Multinomial NB	0.94	0.948
Random Forest	1.00	0.995
Multi-layer Perceptron	0.98	0.986

data balanced using the over-sampling technique SMOTE across classes.

From the different classifiers Decision Tree, Linear SVC, Logistic Regression, Multinomial Naïve Bayes, Random Forest, and Multilayer Perceptron with accuracy for all classes with balanced datasets using up-sampling, the Random Forest again performed best with accuracy 100% as shown in Table IX. The Random Forest achieved the F_1-score 0.99, 1.00, 0.99, 1.00 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is shown in Tables VIII and IX. It is observed that with data balances using SMOTE the precision and recall have also improved for every classifier. And the accuracy of the classifiers varies between 94% to 100% as visualized in Fig. 9.

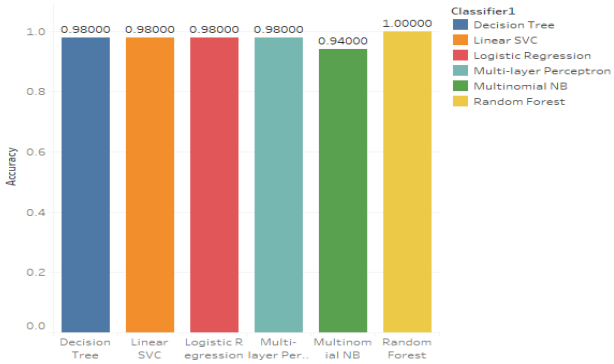

Fig. 9:.Accuracy for various classifiers with balanced classes using SMOTE Up-Sampling.

5. *Results of multiclass classification with data balance using Data-level technique: over-sampling technique ADASYN*

ADASYN (Adaptive synthetic sampling approach) algorithm builds on the methodology of SMOTE. This uses a weighted distribution for specific examples of minority classes due to their degree of learning capacity, whereas more sophisticated data is generated for examples

TABLE X: RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH BALANCED DATA USING ADASYN

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Decision Tree	1, 2, 3	0.89	0.96	0.93	0.82	0.66	0.73	0.91	0.94	0.92	0.42	0.38	0.40
Linear SVC	1, 2, 3	0.96	0.85	0.90	0.76	0.71	0.73	0.89	0.95	0.92	0.62	0.38	0.48
Logistic Regression	1, 2, 3	0.96	0.85	0.90	0.79	0.76	0.73	0.90	0.95	0.92	0.60	0.46	0.52
Multinomial NB	1, 2, 3	0.50	0.85	0.63	0.57	0.93	0.70	0.99	0.65	0.78	0.29	0.77	0.43
Random Forest	1, 2, 3	0.93	0.96	0.94	0.91	0.71	0.79	0.91	0.98	0.94	0.88	0.38	0.53
Multi-layer Perceptron	1, 2, 3	0.94	0.65	0.77	0.73	0.73	0.73	0.88	0.95	0.92	0.88	0.38	0.53

TABLE XI: ACCURACY OF THE CLASSIFIERS WITH BALANCED DATA USING ADASYN UP-SAMPLING

Classifier	Accuracy(Train/Test)	Cross-Validation
Decision Tree	0.87	0.872
Linear SVC	0.87	0.865
Logistic Regression	0.88	0.881
Multinomial NB	0.72	0.725
Random Forest	0.91	0.914
Multi-layer Perceptron	0.86	0.863

TABLE XII: RESULTS FOR THE CLASSIFIERS FOR DIFFERENT CLASSES WITH BALANCED DATA USING NEAR-MISS

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Decision Tree	1, 2, 3	0.68	1.00	0.81	0.23	0.24	0.24	0.82	0.70	0.75	0.25	0.54	0.34
Linear SVC	1, 2, 3	0.41	0.96	0.57	0.30	0.56	0.39	0.88	0.45	0.59	0.27	0.69	0.39
Logistic Regression	1, 2, 3	0.43	0.96	0.60	0.29	0.56	0.38	0.94	0.17	0.28	0.12	0.92	0.22
Multinomial NB	1, 2, 3	0.32	0.88	0.47	0.30	0.59	0.40	0.91	0.32	0.47	0.20	0.77	0.32
Random Forest	1, 2, 3	0.91	0.77	0.83	0.67	0.20	0.30	0.82	0.97	0.89	0.60	0.46	0.52
Multi-layer Perceptron	1, 2, 3	0.46	0.81	0.58	0.57	0.61	0.59	0.91	0.72	0.80	0.28	0.62	0.38

of minority classes that are more difficult to understand. The key idea of the ADASYN algorithm is to use a density distribution as a parameter to automatically calculate the number of synthetic samples that each minority data example requires to be generated. The data is reconstructed after the oversampling process, and the classification models are implemented for the processing data. Table X lists the results of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with data balanced using the over-sampling technique ADASYN across classes.

From the different classifiers Decision Tree, Linear SVC, Logistic Regression, Multinomial Naïve Bayes, Random Forest, and Multilayer Perceptron with accuracy for all classes with balanced datasets using up-sampling, the Random Forest again performed best with accuracy 91% as shown in Table XI. The Random Forest achieved the F_1-score 0.94, 0.79, 0.94, 0.53 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is shown in Tables X and XI. It is observed that with data balances using ADASYN the precision and recall have also downgraded as compared to SMOTE based up-sampling for every classifier. And the accuracy of the classifiers varies between 72% to 91% as visualized in Fig. 10.

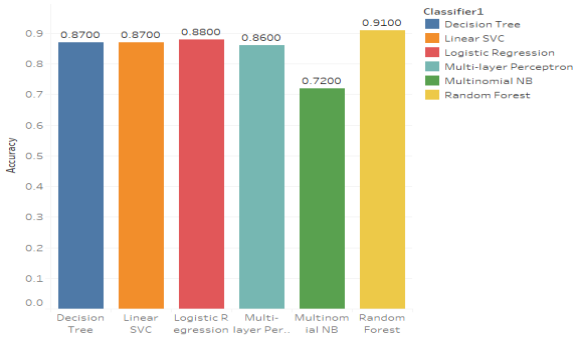

Fig. 10. Accuracy for various classifiers with balanced classes using ADASYN Up-Sampling.

6. *Results of multiclass classification with data balances using Data-level technique: down-sampling technique Near-Miss*

The NearMiss Algorithm under-sampled the majority class's instances and made them equivalent to the minority class. The majority classes, here, were reduced to the minimum number as of minority class so that all classes would have the same number of records. The data is reconstructed after the down-sampling process using the Near-Miss method, and the classification models are implemented for the processing data. Table XII lists the results of each of the classifiers where data is being vectorized using the N-gram TF-IDF feature with data balanced using the down-sampling technique Near-Miss across classes.

ACCURACY OF THE CLASSIFIERS WITH BALANCED DATA USING NEAR-MISS DOWN-SAMPLING

Classifier	Accuracy(Train/Test)	Cross-Validation
Decision Tree	0.65	0.652
Linear SVC	0.53	0.526
Logistic Regression	0.34	0.344
Multinomial NB	0.43	0.431
Random Forest	0.81	0.814
Multi-layer Perceptron	0.70	0.704

TABLE XIV: RESULTS FOR THE ENSEMBLE CLASSIFIERS FOR DIFFERENT CLASSES

Classifier	N-Gram	Banking			Global			Non-Banking			Governmental		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
BalancedBaggingClassifier	1, 2, 3	0.98	0.97	0.97	0.99	1.00	1.00	0.98	0.98	0.98	0.99	1.00	1.00
BalancedRandomForestClssifier	1, 2, 3	0.86	0.64	0.73	0.99	0.90	0.94	0.66	0.88	0.76	0.84	0.87	0.86
RUSBoostClassifier	1, 2, 3	0.33	1.00	0.50	0.94	0.34	0.50	0.08	0.05	0.06	0.00	0.00	0.00
EasyEnsembleClassifier	1, 2, 3	0.87	0.93	0.90	0.57	0.44	0.49	0.26	0.85	0.40	0.92	0.83	0.87

TABLE XV: ACCURACY OF THE ENSEMBLE CLASSIFIERS

Classifier	Accuracy(Train/Test)	Cross-Validation
BalancedBaggingClassifier	0.99	0.991
BalancedRandomForestClssifier	0.82	0.823
RUSBoostClassifier	0.34	0.344
EasyEnsembleClassifier	0.78	0.781

From the different classifiers Decision Tree, Linear SVC, Logistic Regression, Multinomial Naïve Bayes, Random Forest, and Multilayer Perceptron with accuracy for all classes with balanced datasets using down-sampling, the Random Forest again performed best with an accuracy of 81% as shown in Table XIII. The Random Forest achieved the F_1-score 0.83, 0.30, 0.89, 0.52 for classes Banking, Global, Non-Banking, and Governmental respectively. The comparison of all the mentioned classifiers for 4-different classes is shown in Table XII and XIII. The accuracy of the classifiers has degraded with data balances using down-sampling with the Near-Miss approach as compared to all other up-sampling approaches as visualized in Fig. 11.

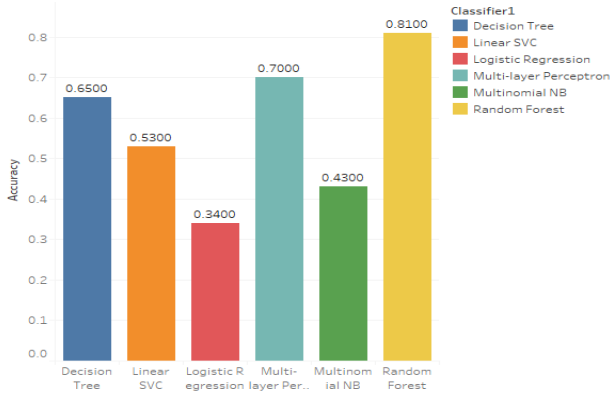

Fig. 11. Accuracy for various classifiers with balanced classes using Near-Miss Down-Sampling.

7. *Results of multiclass classification with data balance using Ensemble classifiers*

Ensemble models are meta-algorithms incorporating many strategies in machine learning into one predictive model to minimize variance (bagging), bias (boosting), or strengthen predictions (stacking). Bagging methods build multiple estimators on various randomly chosen subsets of data in ensemble classifiers. The classifier is called BaggingClassifier in scikit-learn. This classifier, however, does not require a balancing of the data sub-set. So, this classifier would support the plurality groups when training on imbalanced data set. ***BalancedBaggingClassifier*** requires each subset of data to be resampled until any of the ensemble estimators are equipped. In brief, the performance of an EasyEnsemble sampler is paired with an ensemble of classifiers (i.e., BaggingClassifier). Hence the BalancedBaggingClassifier requires the same parameters as the BaggingClassifier scikit-learn. Additionally, there are two additional parameters to monitor the actions of the random under-sampler, sampling strategy, and substitution. ***BalancedRandomForestClassifier*** is another ensemble method that provides a balanced bootstrap sample for each tree in the forest. ***RUSBoostClassifier*** sub-sample the data collection randomly before executing a boosting iteration. A particular method in the bagging classifier which uses AdaBoost as learners is named EasyEnsemble. The ***EasyEnsembleClassifier*** allows AdaBoost learners to be trained on appropriate samples of bootstrap. Table XIV lists the results of each of these ensemble classifiers for the various classes. And Table XV shows the accuracy of these ensemble classifiers.

From the different ensemble classifiers BalancedBaggingClassifier, BalancedRandomForestClssifier, RUSBoostClassifier, EasyEnsembleClassifier with accuracy for all classes, the BalancedBaggingClassifier performed best with an accuracy of 99% as shown in Table XV. The BalancedBaggingClassifier achieved the F_1-score 0.97, 1.00, 0.98, 1.00 for classes Banking, Global, Non-Banking and Governmental respectively. The comparison of all the mentioned ensemble classifiers for 4-different classes is shown in Table XIV and XV. The accuracy of the BalancedBaggingClassifier has resulted in 99% which is quite similar to the result of multiclassification using Random-Forest Classifier with SMOTE sampling i.e. 100%. The accuracy of the Random Forest classifier with a random up-sampling

approach for data balances is also 99%. The comparison of the accuracies of classifiers across all approaches is visualized in Fig. 12. The accuracy of classifiers with down-sampling using the Near-Miss approach is worst amongst all.

The accuracy, precision, recall, and F-1 of Random-Forest Classifier with SMOTE sampling is very good in terms of multiclass news classification. However, under Governmental and Banking classes (minor classes in original), the precision of Random Forest with SOMTE overlapped with the precision of Random Forest with a random up-sampling approach. The comparison of the Precision of classifiers with each approach across all mentioned classes is visualized in Fig. 13. Some of the key explanations for the low performance of some of the classifiers, including Linear SVC and Multinomial naïve Bayes, is that a huge number of features don't fit well for them. Earlier it has been stated that Multinomial Naive Bayes' output is very weak when the dataset faces class imbalance problems. The result has shown that the efficiency of the RUSBoostClaasifer ensemble algorithm is very poor when it comes to the multi-class classification of text with noisy data and class imbalance.

Fig. 12. Comparison of accuracies with Classifiers across different approaches.

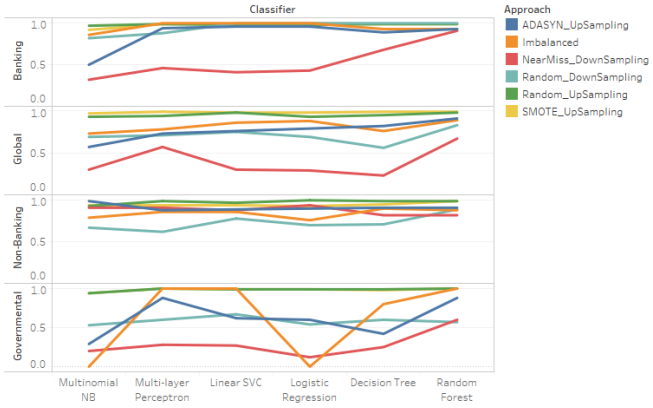

Fig. 13 Comparison of Precision with Classifiers under each class across different approaches.

It is clear from the Fig. 14., the recall of the classifier Random Forest with data balanced across classes using random up-sampling and SMOTE is increased as compared to down-sampling techniques random down-sampling and Near-Miss. The comparison of recall across all approaches with different classifiers under each class is visualized in Fig. 14.

Fig. 14. Comparison of Recall with Classifiers under each class across different approaches.

The accuracy of the ensemble classifiers is compared with Random Forest with SMOTE and it is visualized in Fig. 15. The accuracy of multi-class financial news classification using Random Forest with data balanced using SMOTE is higher as compared to all other ensemble classifiers discussed in the previous section. It is slightly greater than BalancedBaggingClassifier. The precision and recall of Random Forest with data balanced using SMOTE across all classes are higher as compared to all other ensemble classifiers and it is visualized in Fig. 16. and 17 respectively.

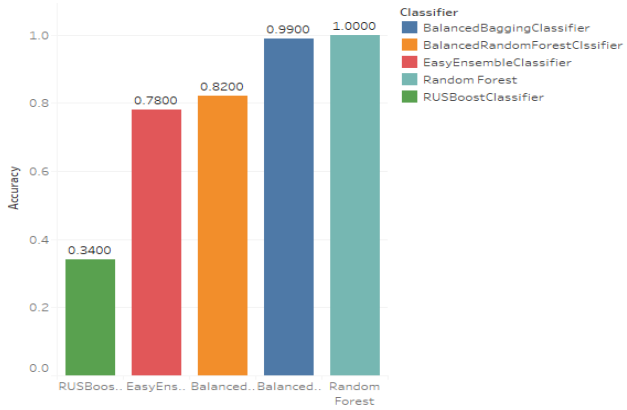

Fig. 15. Ensemble classifiers vs Random Forest (SMOTE).

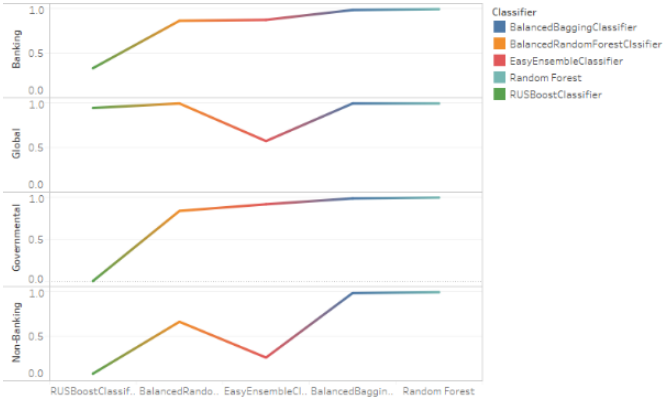

Fig. 16. Precision of ensemble classifiers vs Random Forest (SMOTE).

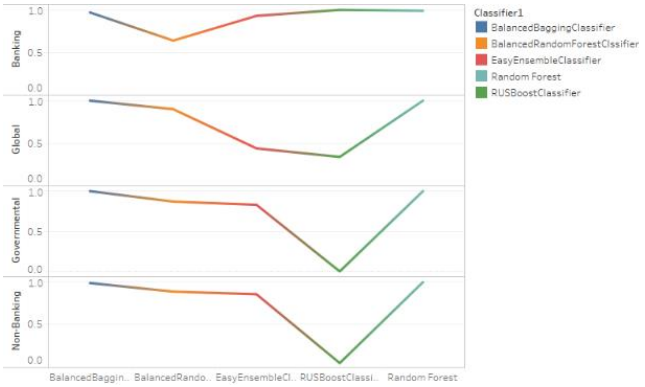

Fig. 17. Recall of ensemble classifiers vs Random Forest (SMOTE).

V. Conclusion and Future Direction

This paper aims to extract banking news from the pool of articles on financial news. This multi-class Financial News classification will help to get news on the banking domain. The development of a system for gathering banking news and other relevant domains is a major and untested problem for the Indian stock market. We're interested in seeking news from Indian banks, the Indian government, and the global. We take a structured approach to divide the news into realms of our choosing, grouping the news articles into 4 classes. The news articles are gathered from numerous online news sources and labeled to derive the banking and other related news to achieve the paper's goal. To automate the classification process, 5 traditional machine learning classifiers, 1 neural network classifier, and 4 ensemble classifiers are used to classify the news articles into 4 classes (Banking, Governmental, Global, and Non-Banking). Since our data set faces the class imbalance issue, we used many methods to align the data set between classes, and the classifier output is evaluated using the original imbalanced and balanced data set. We used precision, recall, F-1, and accuracy parameters to evaluate the classification models. It is evident from results that Random Forest with balanced data using SMOTE achieved the highest accuracy of 100% whereas other models have lower classification accuracy even with 34%. Based on our results, our trained classification model can be used to classify the news into other specific domains by training the model on data-sets of those domains. The labeling of the dataset is done manually at the current stage of our study, with the help of the domain experts. In our future research, including those listed in this paper, we may also use certain recently introduced methods and frameworks for classifying data with a larger volume.

References

[1] Atkins, Adam, Mahesan Niranjan, and Enrico Gerding. "Financial news predicts stock market volatility better than close price," *The Journal of Finance and Data Science* 4, no. 2, pp. 120-137, 2018.

[2] Belainine, Billal, Alexsandro Fonseca, and Fatiha Sadat. "Named entity recognition and hashtag decomposition to improve the classification of tweets," In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pp. 102-111. 2016.

[3] da Costa Albuquerque, Fábio, Marco A. Casanova, Jose Antonio F. de Macedo, Marcelo Tilio M. de Carvalho, and Chiara Renso. "A proactive application to monitor truck fleets," In *2013 IEEE 14th International Conference on Mobile Data Management*, vol. 1, pp. 301-304. IEEE, 2013.

[4] D. McDonald, H. Chen, and R. Schumaker. "Transforming Open-Source Documents to Terror Networks: The Arizona TerrorNet," In *AAAI Spring Symposium: AI Technologies for Homeland Security*, pp. 62-69, 2005.

[5] C.P. Wei, and Y.H. Lee. "Event detection from online news documents for supporting environmental scanning," Decision Support Systems 36, pp. 385-401, 2004.

[6] M.H. Steinberg. "Clinical trials in sickle cell disease: adopting the combination chemotherapy paradigm," *American Journal of Hematology* 83, no. 1, pp. 1-3, 2008.

[7] S. Xiong, K. Wang, D. Ji, B. Wang. "A short text sentiment-topic model for product reviews," *Neurocomputing* 297, pp. 94-102, 2018.

[8] Abbasi, Ahmed, Stephen France, Zhu Zhang, and Hsinchun Chen. "Selecting attributes for sentiment classification using feature relation networks," *IEEE Transactions on Knowledge and Data Engineering* 23, no. 3, pp. 447-462, 2010.

[9] Aggarwal, Charu C. "Machine Learning for Text: An Introduction," In *Machine Learning for Text*, pp. 1-16. Springer, Cham, 2018.

[10] Ahmed, Sajid, Asif Mahbub, Farshid Rayhan, Rafsan Jani, Swakkhar Shatabda, and Dewan Md Farid. "Hybrid methods for class imbalance learning employing bagging with sampling techniques," In *2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)*, pp. 1-5. IEEE, 2017.

[11] Alcalá-Fdez, Jesús, Luciano Sanchez, Salvador Garcia, Maria Jose del Jesus, Sebastian Ventura, Josep Maria Garrell, José Otero *et al.* "KEEL: a software tool to assess evolutionary algorithms for data mining problems," *Soft Computing* 13, no. 3, pp. 307-318, 2009.

[12] Armanfard, Narges, James P. Reilly, and Majid Komeili. "Local feature selection for data classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, no. 6, pp. 1217-1227, 2015.

[13] Bahassine, Said, Abdellah Madani, and Mohamed Kissi. "An improved Chi-sqaure feature selection for Arabic text classification using decision tree," In *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pp. 1-5. IEEE, 2016.

[14] Cao, Peng, Dazhe Zhao, and Osmar Zaiane. "An optimized cost-sensitive SVM for imbalanced data learning," In *Pacific-Asia conference on knowledge discovery and data mining*, pp. 280-292. Springer, Berlin, Heidelberg, 2013.

[15] Chen, Jingnian, Houkuan Huang, Shengfeng Tian, and Youli Qu. "Feature selection for text classification with Naïve Bayes," *Expert Systems with Applications* 36, no. 3, pp. 5432-5435, 2009.

[16] S. Kumar, Ravishankar, and S. Verma. "Context Aware Dynamic Permission Model: A Retrospect of Privacy and Security in Android System," in 2018 International Conference on Intelligent Circuits and Systems , IEEE Xplore, Phagwara, India, pp. 324-329, 2018.

[17] T. Sabbah, A. Selamat, M.H. Selamat, F.S. Al-Anzi, E.H. Viedma, O. Krejcar, and H. Fujita. "Modified frequency-based term weighting schemes for text classification," Applied Soft Computing 58, pp. 193–206, 2017.

[18] B. Vijayalakshmi, K. Ramar, NZ Jhanjhi, S. Verma, M. Kaliappan, et.al. "An Attention Based Deep Learning Model For Traffic Flow Prediction Using Spatio Temporal Features Towards Sustainable Smart City," *International Journal of Communication Systems*, 34, pp. 1-14 ,2020.

[19] S. Schmidt, S. Schnitzer, and C. Rensing. "Text classification based filters for a domain-specific search engine," Computers in Industry 78, pp. 70–79, 2016.

[20] Y. Liu, H.T. Loh, and A. Sun. "Imbalanced text classification: A term weighting approach," Expert System Applications 36, pp. 690–701, 2013.

[21] Ghosh, Samujjwal, and Maunendra Sankar Desarkar. "Class specific TF-IDF boosting for short-text classification: Application to short-texts generated during disasters," In *Companion Proceedings of the The Web Conference 2018*, pp. 1629-1637. 2018.

[22] Dal Pozzolo, Andrea, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. "Credit card fraud detection: a realistic modeling and a novel learning strategy," *IEEE Transactions on Neural Networks and Learning Systems* 29, no. 8, pp. 3784-3797, 2017.

[23] Das, Sanjiv Ranjan. "Text and context: Language analytics in finance," *Foundations and Trends® in Finance* 8, no. 3, pp. 145-261, 2014.

[24] I. Batra, S. Verma and Kavita, and M. Alazab. "A Lightweight IoT based Security Framework for Inventory Automation Using Wireless Sensor Network," *International Journal of Communication Systems* 33, pp.1-16, 2019.

[25] Elagamy, Mazen Nabil, Clare Stanier, and Bernadette Sharp. "Stock market random forest-text mining system mining critical indicators of stock market movements," In *2018 2nd International Conference on*

Natural Language and Speech Processing (ICNLSP), pp. 1-8. IEEE, 2018.

[26] García, Salvador, and Francisco Herrera. "Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy," *Evolutionary Computation* 17, no. 3, pp. 275-306, 2009.

[27] Ghanem, Amal S., Svetha Venkatesh, and Geoff West. "Multi-class pattern classification in imbalanced data," In *2010 20th International Conference on Pattern Recognition*, pp. 2881-2884. IEEE, 2010.

[28] Gomez, Juan Carlos, and Marie-Francine Moens. "PCA document reconstruction for email classification," *Computational Statistics & Data Analysis* 56, no. 3, pp. 741-751, 2012.

[29] Granitto, Pablo M., Cesare Furlanello, Franco Biasioli, and Flavia Gasperi. "Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products," *Chemometrics and Intelligent Laboratory Systems* 83, no. 2, pp. 83-90, 2006.

[30] He, Haibo, and Edwardo A. Garcia. "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering* 21, no. 9, pp. 1263-1284, 2009.

[31] Jeatrakul, Piyasak, and Kok Wai Wong. "Enhancing classification performance of multi-class imbalanced data using the OAA-DB algorithm," In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8. IEEE, 2012.

[32] Jin, Xin, Anbang Xu, Rongfang Bie, and Ping Guo. "Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles," In *International Workshop on Data Mining for Biomedical Applications*, pp. 106-115. Springer, Berlin, Heidelberg, 2006.

[33] 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 (CSUR)* 52, no. 4, pp. 1-36, 2019.

[34] L. Khreisat. "Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study," In *Conference on Data Mining* (*DMIN* 2006), pp. 78-82, 2006.

[35] S.B. Kotsiantis. "Decision trees: a recent overview," *Artificial Intelligence Review* 39, no. 4, pp. 261-283, 2013.

[36] B. Krawczyk. "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence* 5, no. 4, pp. 221-232, 2016.

[37] I. Batra, S. Verma, Kavita, U. Ghosh, J. J. P. C. Rodrigues, *et al.* "Hybrid Logical Security Framework for Privacy Preservation in the Green Internet of Things," MDPI-Sustainability 12, no. 14, pp. 5542, 2020.

[38] J. Lee, I. Yu, J. Park, D.W. Kim. "Memetic feature selection for multilabel text categorization using label frequency difference," *Information Sciences* 485, pp. 263-280, 2019.

[39] G. Lemaître, F. Nogueira, and C.K. Aridas. "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *The Journal of Machine Learning Research* 18, no. 1, pp. 559-563, 2017.

[40] Jing, Li-Ping, Hou-Kuan Huang, and Hong-Bo Shi. "Improved feature selection approach TFIDF in text mining," In *Proceedings. International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 944-946. IEEE, 2002.

[41] G. Liang, C. Zhang. "A comparative study of sampling methods and algorithms for imbalanced time series classification," In *Australasian Joint Conference on Artificial Intelligence*, pp. 637-648. Springer, Berlin, Heidelberg, 2012.

[42] M. A. Jan, B. Dong, S. R. U. Jan, Z. Tazzn, S. Verma, *et al.* "A Comprehensive Survey on Machine Learning-based Big Data Analytics for IoT-enabled Smart Healthcare System," *Mobile Networks and Applications* 26, pp.234-252, Springer, 2021,.

[43] P. Liu, X. Qiu, and H. Xuanjing. "Recurrent neural network for text classification with multi-task learning," In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence,* pp.2873-2879, 2016.

[44] X. Liu, Q. Li, and Z. Zhou. "Learning imbalanced multi-class data with optimal dichotomy weights," In *2013 IEEE 13th International Conference on Data Mining*, pp. 478-487. IEEE, 2013.

[45] R.J. Lyon, J.M. Brooke, J.D. Knowles, and B.W. Stappers. "Hellinger distance trees for imbalanced streams," in *2014 22nd International Conference on Pattern Recognition*, pp. 1969-1974. IEEE, 2014.

[46] D. Fatta, Giuseppe, A. Fiannaca, R. Rizzo, A. Urso, M. R. Berthold, and S. Gaglio. "Context-Aware Visual Exploration of Molecular Datab," In *Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06)*, pp. 136-141. IEEE, 2006.

[47] A. Makazhanov, and D. Rafiei, "Predicting the political preference of Twitter users," *Social Network Analysis and Mining* - ASONAM '13, pp. 298–305, 2013.

[48] K. Mathew, and B. Issac. "Intelligent spam classification for mobile text message," In *Proceedings of 2011 International Conference on Computer Science and Network Technology*, vol. 1, pp. 101-105. IEEE, 2011 .

[49] C. Zhang, J. Bi, S. Xu, E. Ramentol, G. Fan, B. Qiao, and H. Fujita. "Multi-Imbalance: An open-source software for multi-class imbalance learning," *Knowledge Based System* 174, pp. 137–143, 2019.

[50] A. Mazyad, F. Teytaud, and C. Fonlupt. "A comparative study on term weighting schemes for text classification", in *Lecture Notes in Computer Science*, Springer Verlag, pp. 100–108, 2018.

[51] A. Moreo, A. Esuli, and F. Sebastiani. "Distributional random oversampling for imbalanced text classification," In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 805-808, 2016.

[52] A. Onan, S. Korukoğlu, and H. Bulut. "Ensemble of keyword extraction methods and classifiers in text classification," *Expert System Applications* 57, pp. 232–247, 2016.

[53] N.C. Oza, and S. J. Russell. "Online bagging and boosting," In *International Workshop on Artificial Intelligence and Statistics*, pp. 229-236., 2001.

[54] A. Özçift. "Random forests ensemble classifier trained with data resampling strategy to improve cardiac arrhythmia diagnosis," *Computers in Biology and Medicine* 41, no. 5, pp. 265-271, 2011.

[55] V.N. Phu, V.T.N. Tran, V.T.N. Chau, N.D. Dat, and K.L.D. Duy. "A decision tree using ID3 algorithm for English semantic analysis," *International Journal of Speech Technology* 20, no. 3, pp. 593-613, 2017.

[56] T. Pranckevičius, and V. Marcinkevičius. "Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification," *Baltic Journal of Modern Computing* 5, no. 2, pp. 221, 2017.

[57] M. Raza, F.K. Hussain, O.K. Hussain, M. Zhao, and Z. ur Rehman. "A comparative analysis of machine learning models for quality pillar assessment of SaaS services by multi-class text classification of users' reviews," *Future Generation Computer Systems* 101, pp. 341–371, 2017.

[58] F. Khan, A. Shahnazir, N. Ayazsb, S. Khan, S. Verma, and Kavita. "A Resource Efficient hybrid Proxy Mobile IPv6 extension for Next Generation IoT Networks," *IEEE Internet of Things Journal*, 2021, 10.1109/JIOT.2021.3058982

[59] A. P. Singh, A. K. Luhach, S. Agnihotri, N. R. Sahu, D. S. Roy, NZ Jhanjhi, S. Verma, Kavita, and U. Ghosh. "A Novel Patient-Centric Architectural Framework for Blockchain-Enabled Healthcare Applications," IEEE-Transaction on Industrail Informatics 17, no. 8, pp. 5779 – 5789, 2020, 10.1109/TII.2020.3037889.

[60] R.E. Schapire, Y. Singer, and A. Singhal. "Boosting and Rocchio applied to text filtering," In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 215-223. 1998.

[61] R.P. Schumaker, and H. Chen. "Textual analysis of stock market prediction using breaking financial news: The AZFin text system," *ACM Transactions on Information Systems (TOIS)* 27, no. 2, pp. 1-19, 2009.

[62] R.A. Stein, P.A. Jaques, and J.F. Valiati. "An analysis of hierarchical text classification using word embeddings," *Information Sciences* 471, pp. 216–232, 2019.

[63] S. Tan. "Neighbor-weighted K-nearest neighbor for unbalanced text corpus," *Expert System Applications* 28, pp. 667–671, 2005.

[64] H. Tayyar Madabushi, E. Kochkina, and M. Castelle. "Cost-Sensitive BERT for Generalisable Sentence Classification on Imbalanced Data," *arXiv preprint arXiv:2003.11563*, pp. 125–134, 2020.

[65] C.F. Tsai, W.C. Lin, Y.H. Hu, and G.T. Yao. “Under-sampling class imbalanced datasets by combining clustering analysis and instance selection,” *Information Sciences* 477, pp. 47–54, 2019.

[66] A.K. Uysal, and S. Gunal. “A novel probabilistic feature selection method for text classification,” *Knowledge-Based Systems* 36, pp. 226–235, 2012.

[67] B. Verma, and A. Rahman. “Cluster-oriented ensemble classifier: Impact of multicluster characterization on ensemble classifier learning,” *IEEE Transactions on Knowledge and Data Engineering* 24, no. 4, pp. 605–618, 2012.

[68] M.K. Verma, D.K. Xaxa, and S. Verma. "DBCS: density based cluster sampling for solving imbalanced classification problem," In *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, vol. 1, pp. 156-161. IEEE, 2017.

[69] G. Yang, M. A. Jan, A. U. Rehman, M. Babar, and M. M. Aimal. “Interoperability and Data Storage in Internet of Multimedia Things: Investigating Current Trends, Research Challenges and Future Directions,” *IEEE Access* 8, pp. 124382 – 124401, 2020.

[70] V. Dogra. "Banking news-events representation and classification with a novel hybrid model using DistilBERT and rule-based features," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12, no. 10, pp. 3039-3054, 2021.

[71] J. Yan, B. Zhang, N. Liu, S. Yan, Q. Cheng, W. Fan, Q. Yang, W. Xi, and Z. Chen. “Effective and efficient dimensionality reduction for large-scale and streaming data preprocessing,” *IEEE Transactions on Knowledge and Data Engineering* 18, no. 3, pp. 320–332, 2006.

[72] J. Yang, Y. Liu, X. Zhu, Z. Liu, and X. Zhang. “A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization,” *Information Processing Management* 48, pp. 741–754, 2012.

[73] K. Yang, Z. Yu, S. Member, X. Wen, W. Cao, S. Member, C.L.P. Chen, H. Wong, and J. You. “Hybrid Classifier Ensemble for Imbalanced Data,” *IEEE Transactions on Neural Networks and Learning Systems* 31, no. 4, pp. 1–14, 2019.

[74] H. Zhang, and M. Li. “RWO-Sampling: A random walk over-sampling approach to imbalanced data classification,” *Information Fusion* 20, pp. 99–116, 2014.

[75] A. S. Ashour, S. Beagum, N. Dey, A. S. Ashour, D. S. Pistolla, , G. N. Nguyen,... and F. Shi. “Light microscopy image de-noising using optimized LPA-ICI filter,” *Neural Computing and Applications* 29, no. 12, pp. 1517-1533, 2018.

[76] S. Doss, J. Paranthaman, S. Gopalakrishnan, A. Duraisamy, S. Pal *et al.* "Memetic optimization with cryptographic encryption for secure medical data transmission in iot-based distributed systems," *Computers, Materials & Continua* 66, no.2, pp. 1577–1594, 2021.

[77] D. N. Le. “A new ant algorithm for optimal service selection with end-to-end QoS constraints,” *Journal of Internet Technology 18*, no.5, pp. 1017-1030, 2017.

[78] Z. Zhang, B. Krawczyk, S. Garcìa, A. Rosales-Pérez, and F. Herrera, “Empowering one-vs-one decomposition with ensemble learning for multi-class imbalanced data,” *Knowledge Based System* 106, pp. 251–263, 2016.

[79] Z. Sabir, K. Nisar, M. A. Z. Raja, M. R. Haque, M. Umar, A. A. A. Ibrahim, and D. N. Le. “IoT technology enabled heuristic model with Morlet wavelet neural network for numerical treatment of heterogeneous mosquito release ecosystem,” *IEEE Access 9*, pp. 132897-132913, 2021.

[80] T. Zhu, Y. Lin, and Y. Liu. “Synthetic minority oversampling technique for multiclass imbalance problems,” Pattern Recognition 72, pp. 327–340, 2017.

Varun Dogra has been pursuing a Ph.D. in Computer Applications at Lovely Professional University, Phagwara, Punjab, India. He has Bachelor in Science and Masters in Computer Applications. He has also been working as Assistant Professor in the School of Computer Science and Engineering, Lovely Professional University. He has to have 14 years of experience in teaching/ industry. He has published papers in reputed journals and presented papers in International conferences. He has also been reviewed research papers of Scopus/ WoS indexed journals. His area of research covers Artificial Intelligence, Natural Language Processing, Data Science, and Financial Markets.

Sahil Verma (Senior Member IEEE, ACM, IAENG) is Ph. D in Computer Science and Engineering. He is an Associate Professor and (A.) Directior in Chandigarh University, Mohali, India. He has published many research articles in reputed journals/publishers like IEEE, Wiley, Springer, ACM, Elsevier, MDPI etc. He has published papers in reputed top-cited journals like IEEE Transaction in Industrial Informatics, IEEE Transaction on Network Science and Engineering, IEEE Internet of Things Journals, ACM Transaction on Internet Technology, CMC, IEEE Access, MONET Elsevier, HCIS Springer, MTAP Springer, MDPI Sensors, Symmetry and many more. He is reviewer of top-cited journals like IEEE Transaction on Intelligent Transport Systems, IEEE Transactions on Network Science and Engineering, IEEE Access, Neural Computing and Applications Springer, Human-centric Computing and Information Sciences Springer, Mobile Networks and Applications Springer, Journal of Information Security and Applications Elsevier, Mobile Information Systems Hindawi, International Journal of Communication Systems Wiley, Security and Communication Networks Hindawi etc. Dr. Verma is also had professional membership of many reputed organisations like IEEE, ACM, IAENG. His tenure led to an overall Excellence in Education, Research, Infrastructure and Systemic Development of Organization. His current focus is to enhance the Quality of Education through Strategic Quality Initiatives. He has visited many countries like: Austria, Czech Republic, Germany, Switzerland, France, Italy and Thailand for exploring research and development, establishment of labs and for the collaboration with foreign universities (students exchange programs, faculty exchange programs etc.

Kavita Verma is Ph. D in Computer Science and Engineering. She is an Associate Professor at Chandigarh University, Mohali, India. She has published papers in reputed journals like IEEE Transaction in Industrial Informatics, IEEE Transaction on Network Science and Engineering, IEEE Internet of Things Journals, ACM Transaction on Internet Technology, CMC, IEEE Access, MONET Elsevier, HCIS Springer, MTAP Springer, MDPI Sensors, Symmetry and many more. She is also a reviewer of top-cited journals like IEEE Transaction on Intelligent Transport Systems, IEEE Transactions on Network Science and Engineering, IEEE Access, Neural Computing, and Applications

Springer, Human-centric Computing and Information Sciences Springer, Mobile Networks and Applications Springer, Journal of Information Security and Applications Elsevier, Mobile Information Systems Hindawi, International Journal of Communication Systems Wiley, Security and Communication Networks Hindawi, etc. Dr. Kavita Verma has professional membership of many reputed organizations like SMIEEE, MACM, MIAENG, MISCA.

Noor Zaman Jhanjhi (NZ Jhanjhi) is currently working as Associate Professor, Director Center for Smart society 5.0 [CSS5], and Cluster Head for Cybersecurity cluster, at School of Computer Science and Engineering, Faculty of Innovation and Technology, Taylor's University, Malaysia. He is supervising a great number of Postgraduate students, mainly in cybersecurity for Data Science. The cybersecurity research cluster has extensive research collaboration globally with several institutions and professionals. Dr Jhanjhi is Associate Editor and Editorial Assistant Board for several reputable journals, including IEEE Access Journal, PeerJ Computer Science, PC member for several IEEE conferences worldwide, and guest editor for the reputed indexed journals. Active reviewer for a series of top tier journals has been awarded globally as a top 1% reviewer by Publons (Web of Science). He has been awarded as outstanding Associate Editor by IEEE Access for the year 2020. He has high indexed publications in WoS/ISI/SCI/Scopus, and his collective research Impact factor is more than 350 points as of the first half of 2021. He has international Patents on his account, edited/authored more than 30 plus research books published by world-class publishers. He has great experience supervising and co-supervising postgraduate students. An ample number of PhD and Master students graduated under his supervision. He is an external PhD/Master thesis examiner/evaluator for several universities globally. He has completed more than 22 international funded research grants successfully. He has served as Keynote speaker for several international conferences, presented several Webinars worldwide, chaired international conference sessions. His research areas include Cybersecurity, IoT security, Wireless security, Data Science, Software Engineering, UAVs.

Uttam Ghosh is currently working as Associate Professor of Cybersecurity in Meharry School of Applied Computer Science, Nashville, TN, USA. He has been over 10 years of research and development experience in secure wireless and wired communications, Software defined networking, CPS Security. His area of research covers multiple domains like Cyber Physical system Security, Mobile Ad hoc Networks, Wireless Sensor Networks, Software-Defined Networking, Cloud Computing, Distributed Algorithms, and Internet of Things(IoT). He has published many research articles in reputed journals/publishers. He is also a reviewer of top-cited journals. He has a professional membership of reputed organizations like SMIEEE, Sigma Xi, ACM, IEEE, AAAS, ASEE.

Dac-Nhuong Le has an MSc and PhD in computer science from Vietnam National University, Vietnam in 2009, and 2015, respectively. He is an Associate Professor on Computer Science, Deputy Head of the Faculty of Information Technology, Haiphong University, Vietnam. He has a total academic teaching experience of 20+ years in computer science. He has more than 80+ publications in the reputed international conferences, journals, and book chapter contributions (Indexed by SCIE, SSCI, ESCI, Scopus). His areas of research are in the field of intelligence computing, multi-objective optimization, network security, cloud computing, virtual reality/argument reality. Recently, he has been on the technique program committee, the technique reviews, the track chair for international conferences under Springer-ASIC/LNAI/CISC Series. Presently, he is serving on the editorial board of international journals and edited/authored 20+ computer science books published by Springer, Wiley, CRC Press, Bentham Publishers.