

Review on class imbalance techniques to strengthen model prediction

Putta Hemalatha, Geetha Mary Amalanathan

School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

Article Info

Article history:

Received Feb 13, 2024

Revised Feb 28, 2025

Accepted Mar 15, 2025

Keywords:

Class imbalance

Data mining

Machine learning

Model performance

SMOTE techniques

ABSTRACT

Data is a fundamental component in various fields, including science, business, health care, and technology. It is often processed, stored, and analyzed using computer systems and software applications. The importance of data lies in its ability to provide valuable insights, drive innovation, and improve decision-making processes. However, it's essential to handle and manage data responsibly to address privacy and ethical considerations. Data mining (DM) involves discovering patterns, trends, correlations, or useful information from large datasets. Data dredging or DM and machine learning (ML) are closely related fields that both involve the analysis of data to discover patterns and make predictions. DM focuses on extracting knowledge from data; ML emphasizes the development of algorithms that can do analysis. The two fields are interconnected, and the techniques from one state of art integrated into the processes of the other. In ML the class imbalance problem occurs due to the class distribution in the training data is not equal. Imbalanced classification refers to a condition where a particular class (minority class) is under represented paralleled to another class (majority class) in a dataset. This paper furthermore emphasizes on the synthetic minority oversampling technique (SMOTE) variants employed by the researchers, and highlights the limitations the work.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Geetha Mary Amalanathan

School of Computer Science and Engineering, Vellore Institute of Technology

Vellore Campus, Tiruvalam Rd, Katpadi, Vellore, Tamil Nadu 632014, India

Email: geethamary.a@gmail.com

1. INTRODUCTION

In machine learning (ML), class imbalance is a scenario where we see highly skewed distribution of classes in a particular dataset, with one or more classes being much more common than others. Classifiers may become biased towards the majority class as a result, which emphasis on poor performance on the minority classes. In a real-world application, where the minority class represents significant or critical instances, such as medical diagnosis, fraud detection, and anomaly identification, addressing class imbalance is essential. Methods like cost-sensitive learning, ensemble approaches, and resampling are frequently used to lessen the impact of class imbalance and enhance ML model performance. This work discusses the different techniques to handle the class imbalance problem, different synthetic minority oversampling technique (SMOTE) based methods, different classification models, various evaluation metrics which is used to measure the model performance, and challenges associated with a class imbalance in the form of a literature review. The analysis of existing different categorization approaches helps us to better understand the difficulties that the imbalance data set presents. ML techniques have been widely used in a variety of applications, including finance, clinical diagnostics, marketing, banking, epidemiology, and bioinformatics [1]. Most of the datasets in the real-world seem to be imbalanced. The nature of imbalance and importance of

the minority samples and its wide range of data mining (DM) application fields are discussed in detailed by [2], which also delivers a synopsis of the evaluation metrics and current procedures for assessing and resolving imbalance problems. One over-sampling technique that addresses this issue is the SMOTE. In essence, an imbalanced database has minority and majority information. DM is one of the key approaches for handling voluminous imbalanced data unfolding hidden patterns through its techniques such as classification, association analysis, clustering, and pattern recognition [3]. Setting up a real-time data processing, datasets frequently consist of a greater quantity of data belonging to one class, known as the majority class, and a lesser number belonging to the minority class. Almost majority of the cases in these issues are labelled as belonging to one class, whereas much fewer instances are labelled as belonging to the other class, which is typically the more significant class. Imbalance is a situation where there are biased number of samples in one class when compared to other classes. In these circumstances, the classifier created may predict in favor of the dominant class. In earlier research investigations, several strategies were developed to address the issue of class imbalance. Therefore, the author aims to review various techniques for addressing the class imbalanced problem.

2. BACKGROUND ON DATA MINING, MACHINE LEARNING, AND CLASS IMBALANCE

2.1. Data mining play's vital role to enhance class imbalance problems

Class imbalance is a very important problem in real life scenarios, and such applications include; health care, fraud analysis, information retrieval, and text mining. This imbalance in datasets can significantly reduce the efficiency of conventional ML algorithms that have been based on fairly balanced data. This problem is addressed effectively using DM by applying various methods and techniques that identify complex structures and correlations between classes in sporadically imbalanced datasets. New techniques include synthetic sampling, cost-sensitive learning, ensemble techniques, and hybrid architecture to ensure that the models solve the problem of under representation of the minority classes while maintaining the merits of balanced models. This paper proposes the use of DM techniques in building reliable and fair models for effective handling of imbalance as the volume and complexity of data increases.

- Improved model performance: when the data is imbalanced, it can lead to the creation of imbalanced models that have poor performance when it comes to the minority class. Preprocessing techniques such as resampling helps to correct imbalance in the number of instances that belongs to the minority class thereby increasing the ability of the model in correctly identifying this class [4].
- Better generalization: imbalanced datasets can result in models that overfit to the majority class. DM methods, like feature selection and dimensionality reduction, can help create a more balanced and representative dataset, allowing the model to generalize better.
- Anomaly detection: in scenarios like fraud detection, identifying rare events (minority class) is crucial. DM methods can help detect anomalies or rare cases effectively, even in imbalanced datasets.
- Cost reduction: in certain applications, misclassifying instances from the minority class can have significant costs or consequences. DM helps in building models that reduce these costs by addressing the class imbalance issue.
- Ensemble methods: ensemble learning techniques, like bagging and boosting, are widely used in DM to improve model performance on imbalanced datasets. These methods combine multiple models to make predictions, and they can be adapted to give more weight to the minority class [5].
- Resampling techniques: DM offers several methods of resampling including oversampling (generating additional instances for the minority class), under-sampling (reducing the number of instances, belonging to the majority class), and synthetic data generation (creating artificial samples in the context of the minority class). These techniques can balance the dataset and improve model training [6].
- Cost-sensitive learning: DM allows for the incorporation of cost-sensitive learning techniques, where misclassifications of different classes are assigned different costs. This is particularly useful in scenarios where misclassifying the minority class is costlier [7], [8].
- Feature engineering: feature engineering techniques in DM can help create informative features that are particularly useful for distinguishing between classes in imbalanced datasets.
- Active learning: DM techniques can be applied to active learning scenarios where the model selects instances for labelling that are likely to improve its performance on the minority class, thus optimizing the learning process.
- Evaluation metrics: DM provides a wide range of evaluation metrics beyond accuracy, such as precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC), which are more informative for imbalanced datasets. These metrics give a better understanding of a model's performance on both majority and minority classes [9].

2.2. Machine learning approaches for class imbalance problems

ML is a subset of artificial intelligence (AI) that enables on developing models, compute, and learn from evidences and become better at certain tasks [10]. In essence, DM focuses on the discovery of patterns and relationships within data, while ML focuses on building algorithms that can learn from data to perform tasks such as classification, regression, and clustering. These two fields are often intertwined, with techniques and insights from one field informing and enhancing the other. ML can be further been categorized into supervised learning, unsupervised learning, and reinforcement learning as sketched in Figure 1. In supervised learning, each data point has a target label, thus the system learns from labelled data. The objective of any model is to develop a correlation among the input attributes and output labels so that the model can effectively predict the class accurately from unseen data. In contrast, the method of unsupervised learning uses unlabeled data [11]. Supervised learning looks for structures, correlations, and patterns in the data, for example: clustering or dimensionality reduction or grouping comparable data points together. In reinforcement learning, a system assumes the role of an agent and interacts with its surroundings to discover how to maximize reward signals. It is frequently employed in situations where an agent develops the ability to make a series of decisions. ML is a continuously evolving discipline that has seen remarkable progress across multiple fields. However, there are still several research challenges that researchers and experts are actively working to address. Some of these challenges include: interpretable and explainable models, bias and fairness, handling multi class or class imbalance problems, adversarial attacks and defenses, scalability and efficiency, and causal inference.

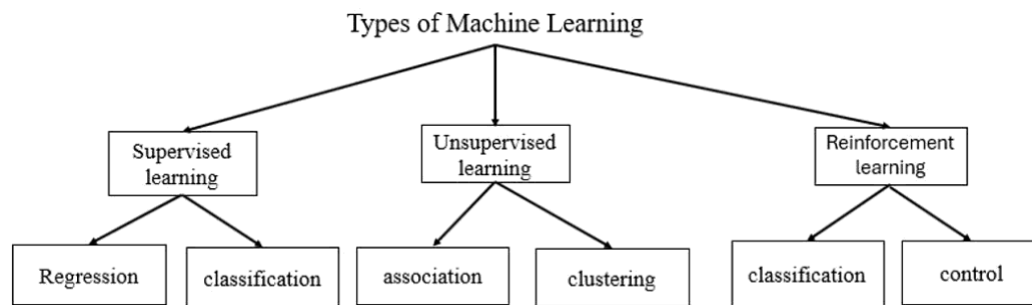


Figure 1. General overview of different types of ML tasks

2.2.1. Multi-label class vs. multiclass classification

Multi-label class and multiclass classification are the common problems in ML and affect the field of DM and statistics. Numerous approaches have generally been put forth to balance the unbalanced data sets [12], [13]. However, due to the larger data size, it remains a challenging task as shown in Figure 2.

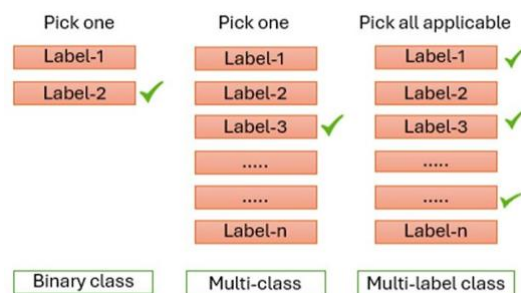


Figure 2. Different types of class learner challenges in ML

2.2.2. Multiclass data

Multiclass data refers to a form of dataset frequently utilized in ML and data analysis applications. Each data point in a multi-class dataset corresponds to one of several preset classes or groups. To put it another way, each data point can be classified into one class out of three or more classes. The multiclass method paves a path to deal with data points that belong to numerous classes and perform ML and data analysis.

2.2.3. Applications of multiclass data

There are several uses for multiclass data in numerous disciplines and sectors. Following will be a few examples of how multiclass data is applied in various fields [14]: i) image processing: consider a dataset of images where the aim is to categorize the images into multiple object classes, for example cats, dogs, birds, and cars. The challenge is one of multiclass categorization because each image belongs to one of these various classes; ii) medical diagnosis: multiclass classification refers to the process of classifying different diseases based on information about the patient, such as symptoms and test findings; iii) text categorization: in natural language processing, you might have a dataset of text documents that need to be categorized into various topics or themes. Each document can belong to one of many possible categories, such as sports, politics, entertainment, and technology; iv) speech recognition: it is the process of identifying words used in speech, phrases, or commands and categorizing them, allowing voice assistants to comprehend various user commands; v) manufacturing and quality control: it involves inspecting and categorizing products on an assembly line according to their quality or flaws; and vi) financial data analysis: financial data analysis involves examining and interpreting data related to financial transactions, investments, markets, and economic indicators to gain insights and make informed decisions.

2.3. Class imbalance

Accurately estimating the class distribution becomes an issue when dealing with data where certain classes have few samples than others, also known as class imbalance. There is a tendency to increase the dependence on models trained on such data and they may be also prejudiced to the majority class, leading to inaccurate predictions and misclassification. This issue frequently arises in fields like anomaly detection, risk assessment, crime detection, medical analysis, and spam filtering, where one class appears significantly less often than the others.

- The challenges of class imbalance are follows: i) bias towards majority class: models tend to perform well in the majority class while struggling to predict the minority class due to the abundance of majority class samples; ii) reduced generalization: the model might generalize poorly to new data, especially for the minority class, leading to poor performance on unseen data; and iii) biased evaluation: traditional accuracy might be misleading since a naive model that predicts the majority class for everything could still achieve high accuracy in imbalanced datasets.
- Strategies to address class imbalance are follows: i) resampling techniques: in ML, resampling is a typical method for addressing class imbalance. By choosing instances from the original dataset, a new training dataset with a different class distribution will be created. Resampling techniques can be divided into the following categories; ii) oversampling: increasing the number of instances in the minority class by duplicating or generating synthetic data points [15]; iii) under sampling: reducing the number of instances in the majority class by randomly removing samples [16]; iv) SMOTE: its engenders fabricated or synthetic samples for the minority class based on incorporating among the existing data points [17]; v) cost-sensitive learning: this approach in ML accounts for the varying costs related with multiple kinds of misclassification errors in a classification model. Conventional ML models aim to decrease overall error, but they do not account for the particular costs connected with false positives and false negatives, in scenarios where misclassification costs are imbalance. Cost-sensitive learning can be easily integrated with classification regression tree (CART), support vector machine (SVM), and naïve Bayes [18]; vi) anomaly detection: the identification of patterns or occurrences within a dataset that deviate from expected behavior is called anomaly detection, sometimes referred to as outlier detection. The choice of anomaly detection technique, such as statistical, density- or distance-based methodologies, based on the nature of the dataset and the explicit necessities of the application [19]; vii) data augmentation: it is a procedure of performing various modifications on the training data and its widely applied on computer vision and ML to synthetically raise the volume of a training dataset. Several well-liked methods for enhancing data include random swapping, translating, cropping, pitch shifting, time warping, scaling, and deformation [20]; viii) ensemble methods: ensemble methods are a powerful approach that handles the class imbalance in ML. Some ensemble methods that can be effective for addressing class imbalance like SMOTE based approaches, easy ensemble, AdaBoost, and XGBoost with scale pose weight [21].
- SMOTE: It is an important technique used in ML and DM to deal with imbalance problem in classification problems [22]. It specifically addresses this issue since it produces synthetic samples of the minority classification. The core principle of SMOTE involves creating synthetic instances that resemble existing minority class samples, thereby enhancing the representation of the minority samples and improving class distribution balance through these synthesized samples, SMOTE opens the ability of the model to learn most of the characteristics of the minority class rather than just focusing on the majority class thus improving on the classification of the model. This leads to improved classification

performance on the underrepresented class and overall better generalization of the model. There are several variations of SMOTE, each designed to address specific challenges or scenarios. The different types of SMOTE methods are as follows [23]: i) SMOTE: it is one of the methods used in ML and DM to deal with the challenge of class imbalance in the classification problem. Class imbalance refers to a situation whereby the instances belong to one class are very few than those of the other class. In order to solve the problem, SMOTE create new samples based on the existing instances of the minorities class. In order to create minority samples SMOTE, applies arbitrary selection of a minority class instance and identifying a group of its k-nearest neighbors (KNN), and then creating new synthetic instances based on these relationships [24]; ii) Borderline-SMOTE: this algorithm belongs to the extended SMOTE algorithm family which is a method that originated from the basic SMOTE algorithm but with an aim of handling borderline instances whereby they are constructed from instances in the minority class nearby the decision border. Instead of using all minority class instances, borderline-SMOTE only selects those instances that are near the decision boundary. This approach aims to generate synthetic instances where they are most needed, near the borderline between classes [25]; iii) adaptive synthetic (ADASYN) sampling approach: it is an adaptive extension of SMOTE. It aims on crating more fabricated/synthetic samples for minority class instances that are difficult to classify, such as those close to the decision boundary. ADASYN uses a density distribution to determine the number of synthetic samples to create for each minority class instance, making it adaptive to the data distribution; iv) SMOTE with edited nearest neighbors (ENN): combines SMOTE with the ENN algorithm. It first oversamples the minority class using SMOTE and then uses ENN to clean the synthetic samples by removing noisy ones and borderline instances from both the minority and majority classes [26]; v) SMOTE-Tomek links: combination of SMOTE and the Tomek links under sampling method. It uses SMOTE to oversample the minority class and then identifies Tomek links (pairs of instances from different classes that are close to each other and act as mutual nearest neighbors). The Tomek links are removed, leading to a cleaner dataset [27]; vi) safe-level SMOTE: designed to address the issue of overgeneralization when generating synthetic samples. It calculates a “safe level” for each minority class instance based on the density of the majority class instances in its neighborhood. Synthetic samples are generated for instances with safe levels above a certain threshold [28]; vii) evolutionary multi-distribution oversampling (EMDO)-SMOTE: an extension of SMOTE that can handle datasets with multiple minority class distributions. It adapts the oversampling strategy based on the distribution of the minority class instances, allowing it to generate synthetic samples more effectively; viii) Gaussian SMOTE: it is an advanced variation of the original SMOTE algorithm used in assessing the imbalanced class problem in ML. It builds upon the concept of SMOTE by introducing a more sophisticated approach to synthesizing additional samples to improve the personification of the minority class labels. This approach enhances process by considering the distribution of feature values using Gaussian distributions. This outcomes in fabricated samples that are not only located on line segments between instances but are also influenced by the underlying distribution of the data [29]; and ix) fuzzy based Gaussian (FG)-SMOTE: it is oversampling technique employed by the researcher to address the class imbalance problem where fuzzy Gaussian methodology was used. The results proven to be more satisfying than the existing methods of SMOTE variants [30].

Each of these variations of SMOTE has its strengths and is suited to different scenarios and datasets. The preference of which SMOTE variant to use based on the particular characteristics of the dataset and the challenges posed by class imbalance in the ML task.

3. SURVEY ON CLASS IMBALANCE PROBLEMS

A key challenge in the imbalanced class problem is the reliability of accurateness and error proportion as metrics for evaluating classifier performance. Figure 3 shows the unequal distribution between the two classes, where green rings represent the majority classes, and the blue color star represents the minority classes. Figure 4 shows the sample generation over imbalanced datasets with respect to the synthetic samples (yellow diamond).

Consider a dataset where the 99% of the data with majority class accounts and 1% of the minority class of the data. Then it's called a dataset with a 99:1 imbalance. A classifier like naive, which consistently predicts the majority class, will have a 99% accuracy rate under such circumstances. Therefore, the majority classifier will be 99.99% accurate if a dataset has a balance ratio of 9999:1. Consider the effectiveness of utmost conventional classifiers when used in the imbalanced class problem to illustrate one effect of this constraint. Such cases classifiers can achieve great accuracy, but it is worthless in most real time applications then the minority class is frequently the class of interest (otherwise, for this a classifier need not be required, as the class of interest occurs practically always) [31]. As a result, many strategies have remained created to address the issue of imbalance class. These techniques can be divided into two categories: sampling

techniques and skew-insensitive classifiers. A minority class is typically ignored or missed by a standard classifier since there aren't many examples of the class. One over-sampling method that addresses this issue is SMOTE.

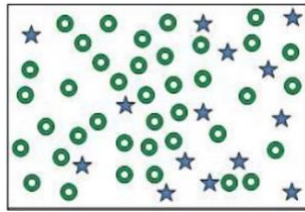


Figure 3. Example of imbalanced dataset

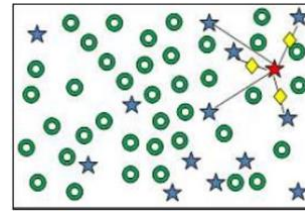


Figure 4. Generating sample over imbalanced dataset

Imbalanced class algorithms can be divided into three types such as ensemble-based model, data-level, and algorithm level [32]. Among them, data-level methods which include the three sub types of oversampling, under-sampling, and hybrid schemes are widely used. The ensemble classifiers can improve the accuracy of a single classifier by integrating multiple classifiers together, the class imbalance problem cannot be resolved easily by conventional learning strategies. Instead, ensemble learning algorithms must be specially created to address this issue. By attempting to restore the equilibrium between the majority and minority categories, these techniques alter the information itself. Different types of class imbalanced algorithms have developed based on the data-level approach in the past decades as shown in Figure 5. They achieve this by increasing the number of minority category samples (oversampling) or by decreasing the number of majority category samples (under-sampling) [33]. Under-sampling involves selecting samples at random or employing heuristic methods to exclude the less important patterns. However, under-sampling is sometimes regarded as dangerous due to the increased likelihood of losing potentially important data. Oversampling is the other, possibly more effective choice. Creating new minority category patterns or randomly reproducing minority category patterns are the two main ways that oversampling is performed. The crucial benefit of this data-level approach is that it is simple to generate new data to make up for the minority category's data deficiency. As a result, oversampling is frequently favored in data-level approaches.

Synthetic samples remain generated for the minority class in oversampling techniques to attain the balanced dataset. A well-known oversampling method frequently used to address class imbalance issues is SMOTE [34]. It is quite easy to use and very effective in practice. It produces synthetic samples precisely where the minority class joins. However, oversampling with SMOTE does not really account for the distribution of samples from the majority class during the development of synthetic samples, which results in the generation of unnecessary minority samples around the desired samples [35].

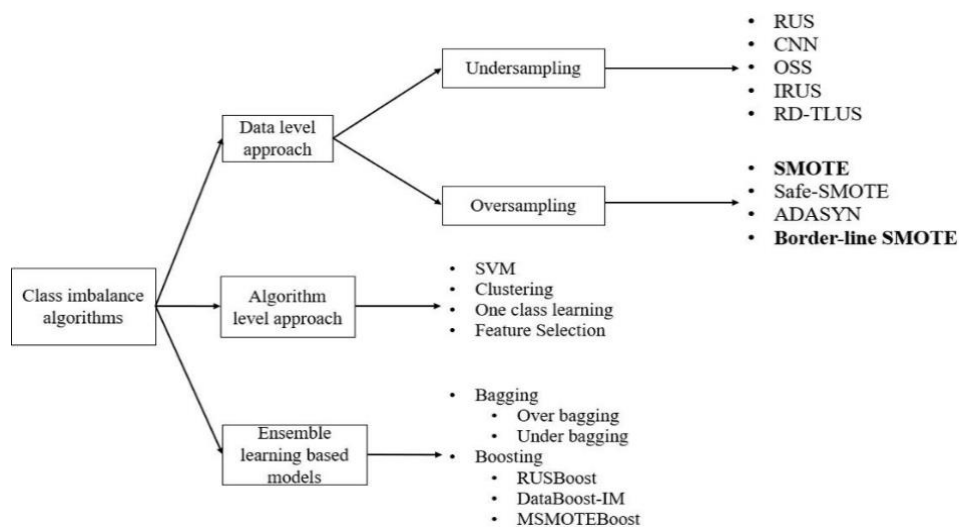


Figure 5. Different types of class learner challenges in ML

3.1. Related work

Huge amounts of unprocessed data are produced in the ML area because of the tremendous progress in research and technology. Each dataset has a unique distribution and informational structure. Due to the skewed distribution of classes in some datasets, an imbalanced determination is presented. In other words, classifiers should be forced to treat minor cases as big examples when handling imbalanced information sets. DM is the process of discovering hidden knowledge from the data. The primary methods are pattern mining, association analysis, grouping, and classification. ADASYN is a ML-based oversampling technique to solve the issue of imbalanced datasets [36]. The fundamental idea behind ADASYN is to employ a weighted distribution for minority class instances depending on their learning exertion. Harder-to-learn minority class examples receive more synthetic samples associated to those that are simpler to study. Additionally, ADASYN dynamically adjusts the classifier's decision boundary, focusing more on challenging instances to improve learning effectiveness. This is accomplished through adaptive weight adjustments and a learning process that considers the data distribution, ensuring a more balanced and effective classification.

With more representative training data for the minority class, ADASYN enhances the quality of ML representations when handling with imbalanced datasets by generating fabricated samples for the minority class, ensuring a more balanced representation and improving classification accuracy. It should be used wisely, though, particularly in conjunction with other methods for dealing with class imbalance, such as using various assessment metrics or utilizing other sampling techniques like under-sampling or combining oversampling and under sampling. In the case of multi-class data sets, it is much more difficult to define the majority and minority classes proposed and addressed imbalanced data comprising many classes based on clustering and SMOTE. The strategy put into practice, and the results of the experiments demonstrate that it is a useful tool for handling multiclass imbalanced data sets. It can also enhance the minority class's classification performance both within and throughout the entire data set. Using several balancing techniques created a SMOTE framework based on a genetic algorithm for learning healthcare data that has an imbalanced distribution. In healthcare data analytics, finding infrequent but important healthcare events in enormous unstructured datasets has become a typical problem. However, because most classification algorithms implicitly assume an equal occurrence of classes and are created to improve overall classification accuracy, uneven class distribution in many practical datasets significantly hinders the detection of uncommon occurrences.

Fuzzy support vector machine (FSVM) is widely acknowledged as a noteworthy enhancement to soft margin SVMs, such as C-SVM, as the latter produces suboptimal outcomes when outliers are present. Fuzzy-SMOTE broadens the minority decision boundary by producing more minority class training instances, which enhances classifier performances on imbalanced datasets. The fuzzy-SMOTE algorithm assigns a membership degree to each minority class instance based on the original class distribution in the training dataset. It then generates a higher number of synthetic samples for instances with lower membership degrees, ensuring a more balanced representation of the minority class. In order to produce new training dataset, the old training dataset is combined with fresh synthetic instances from the minority class [37].

A ML method-based class imbalance technique is proposed which uses a cost-sensitive modification of the least mean square (LMS) algorithm to penalize errors of various samples with various weights and some general guidelines to choose those weights. Following the balancing stage, employs various classification algorithms. When compared to the other classes, these applications frequently have one or more minority classes with very few samples. Numerous studies have demonstrated that DM techniques fall short of desired performance when the training data is imbalanced. Traditional ML approaches rely on balanced category distributions for training data and, in addition, are likely to accurately distinguish the majority category while misclassifying the minority category. This is primarily due to the disparity in class. The data-level approach is often used in handling class imbalance due to its effective performance and classifier independence. By attempting to restore the equilibrium between the majority and minority categories, these techniques alter the information itself. As a result, oversampling is frequently favored in data-level approaches. Synthetic samples are created for the minority group in oversampling procedures to balance an imbalanced database.

Class imbalance learning using intuitionistic fuzzy twin support vector machine (CIL-FART-IFTSVM) and fuzzy adaptive resonance theory (ART) is proposed to overcome the challenges posed by noise imbalance. This approach effectively addresses class imbalance in datasets containing noise outliers and large-scale data, enhancing model robustness and classification performance. Resampling techniques are frequently used to create synthetic data. Generative adversarial networks (GANs) can also be used as an alternative to information gain address the issue of class imbalance by producing tabular data, despite their primary purpose being the generation of image data. In this work, resampling techniques and GAN-based techniques are compared. When resampling methods are used to balance the data, ML methods perform 27% better. One popular algorithm for balancing train data that adds synthetic data on minor class data is called SMOTE. One of the key steps in the SMOTE process is identifying the KNN using Euclidean distance to

generate synthetic data. However, relying solely on Euclidean distance without considering attribute correlations may lead to the selection of unrepresentative neighbors, especially when certain attributes exhibit higher correlation values than others. To address this limitation, this study developed attribute weighted and KNN hub on SMOTE (AWH-SMOTE), it incorporates attribute weighting and KNN hub selection to enhance the quality of synthetic data generation., which improves SMOTE by using occurrence data in the KNN hub to improve the selective sampling method and by using attribute weighting to improve neighbor's and noise identification. The methods of information gain are applied to attribute weighting, borderline-SMOTE1 and borderline-SMOTE2 are variants of the SMOTE method that specifically oversample minority class samples located near the decision boundary. Experimental results indicate that these approaches outperform both SMOTE and random oversampling in terms of true positive (TP) rate and F-value for the minority class. Since minority class instances near the boundary are more susceptible to misclassification, these techniques enhance their representation and improve classification performance. Thus, while SMOTE and random over-sampling bring all examples from the minority class or a random subset of the samples of the minority class, the proposed approaches only oversample the samples of the minority class that is close to the decision surface. Therefore, this paper develops a fuzzy-based SMOTE algorithm with the purpose of adding the fuzzy set theory to the SMOTE. The fuzzy-SMOTE produces additional fabricate samples for minority class instances, particularly in the fuzzy region, where minority examples exhibit lower membership degrees to the minority class and are more prone to misclassification. As the sample size of the majority class increases relative to the minority class, fuzzy-SMOTE ensures a more balanced distribution by creating synthetic instances that expand the decision boundary of classifiers. By incorporating fuzzy-based sampling, fuzzy-SMOTE reduces classifier bias toward the majority class, allowing models to construct broader decision regions with enhanced representation of minority cases. Comparative analysis demonstrates that fuzzy-SMOTE outperforms traditional SMOTE and borderline-SMOTE in terms of accuracy, effectively improving classification performance for both the minority class and the overall dataset. CIL-FART-IFTSVM and fuzzy ART is proposed as an effective approach for addressing class imbalance, particularly in the presence of noise, outliers, and large-scale datasets. To tackle the imbalance issue, fuzzy ART is utilized as a clustering technique to modify the dataset distribution, ensuring a more balanced representation of minority and majority classes. Following this, IFTSVM is applied to construct optimal non-parallel hyperplanes on the newly transformed dataset after data reduction. Finally, to enhance computational efficiency, a coordinate descent system with shrinking via an active set is employed, significantly reducing the computational complexity of the learning process. This approach ensures improved classification performance in challenging imbalanced data scenarios.

3.2. SMOTE

SMOTE is a popular pre-processing method for correcting class imbalance in ML datasets, especially in binary classification problems. SMOTE creates synthetic samples for the minority class to balance the distribution of the classes. This helps avoid model bias in favor of the dominant class and can enhance classification performance. The SMOTE process's goal was to interpolate data from diverse minority class samples. Generating novel minority class samples in the nearby location, will increase the number of minority class occurrences and help classifiers become more generic. The SMOTE model is a notable pioneer in imbalanced classification. The SMOTE methodology in the data pre-processing has grown in prominence among academics. To improve the performance of SMOTE algorithms in various situations, numerous alternatives and technological advancements were provided. SMOTE was widely used and accepted, making it one of the potential mechanisms for sampling and data pre-processing in DM and ML applications.

To balance the preliminary training set, the SMOTE method applying an oversampling strategy by generating synthetic samples instead of merely duplicating minority class instances. The core idea behind SMOTE is to create new-fangled data points through exclamation between existing minority class samples within a defined neighborhood. Unlike traditional methods that focus on individual data points, SMOTE operates in the feature space, leveraging the values and relationships of features to synthesize new samples. This distinction highlights the need for a complete assessment of the theoretical association between real and synthetic instances, considering the impact of data dimensionality on the efficiency of the generated samples.

The link between the distributions of training and test instances, as well as other features like variance and correlation in the data and feature space, must be considered as illustrated in Figure 6. To produce new synthetic data points, minority class instance x_i is chosen as the foundation. Several nearest neighbors of the same class x_{i1} - x_{i4} are designated from the training set based on a distance measure. To obtain fresh instances r_1 - r_4 , a randomized interpolation is then performed.

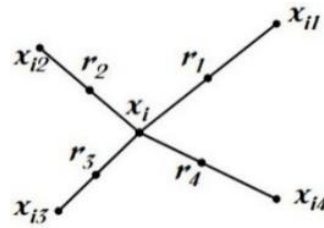


Figure 6. A simple illustration of SMOTE algorithm

3.3. General workflow of SMOTE

SMOTE is a widely used oversampling technique that generates synthetic samples rather than simply duplicating minority class instances, effectively mitigating class imbalance. A detailed workflow in shown in Figure 7. It is important to note that while SMOTE is a useful technique for addressing class imbalance, it may not be suitable for all types of datasets or imbalance scenarios. Additionally, there are variations and extensions of SMOTE, such as borderline-SMOTE, ADASYN, fuzzy-based, and Lorentzian based SMOTE which address specific challenges in handling imbalanced data is discussed in Table 1.

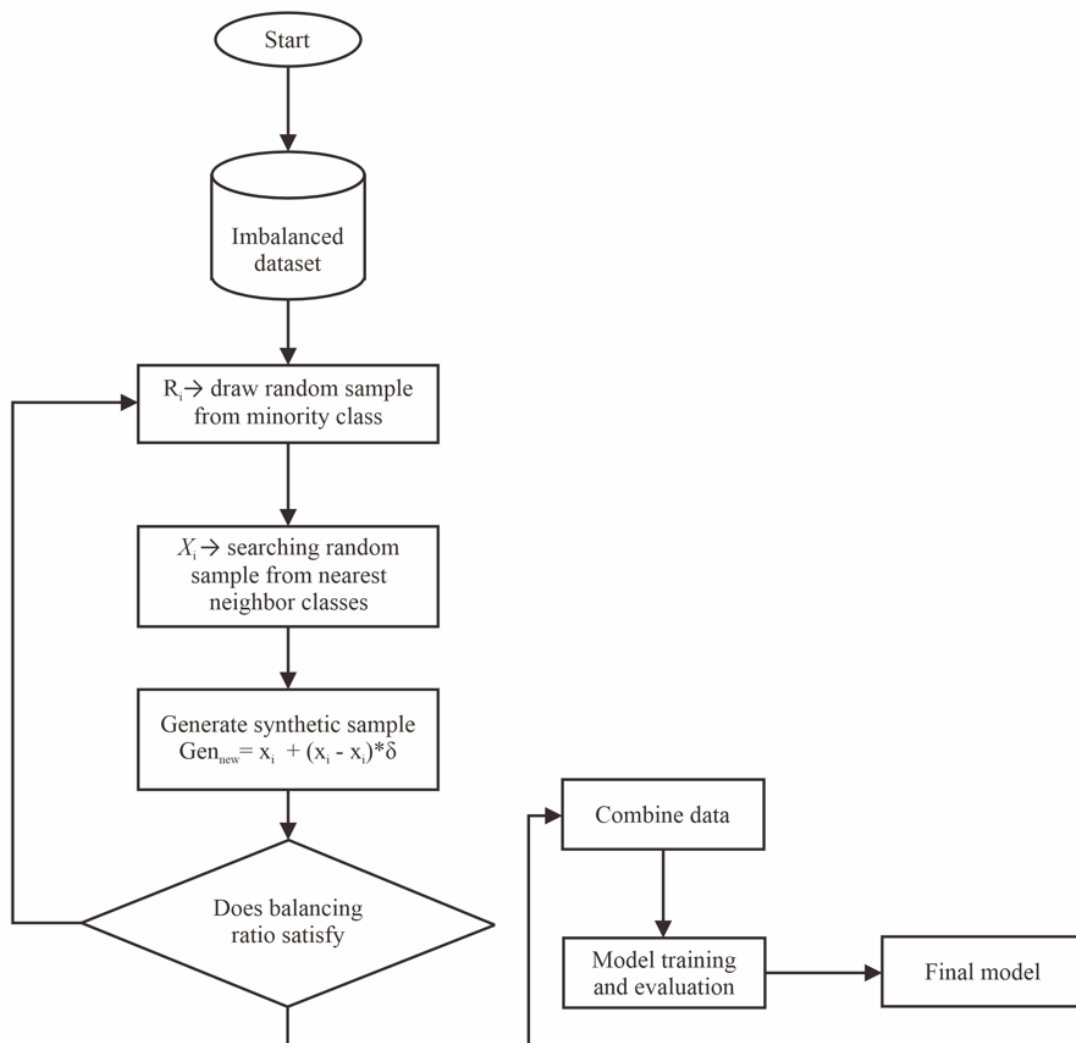


Figure 7. General Workflow of SMOTE Model

Table 1. Summary of SMOTE based techniques

Related work	Approaches	Metrics used	Remarks	Data sets used
[38]	A-SMOTE	F-measure, accuracy	Achieved greater result in binary classification problems.	44 highly imbalanced
[39]	AMDO	AUC	Outperformed in handling multiclass imbalanced data	10 sets
[14]	SMOTE	General metrics	Exhibited better performance in handling multiclass problems than conventional method.	15 UCI repository data sets
[18]	SMOTE and k-means	Accuracy metric	It prevents the introduction of noise and effectively addresses both inter-class and intra-class imbalances.	LUCAS highly imbalanced data set
[8]	D-SMOTE	Recall, f-value, precision	Evolved based on DB-SCAN and SMOTE for multi-class imbalance problem.	16 High dimensional data sets
[40]	Dynamic SMOTE	Minimum recall, Mc Nemar's Test, G-Mean	Outperformed the borderline SMOTE and traditional SMOTE best algorithm for multi-class imbalance problem	29 data sets from different software repositories
[41]	Cluster oriented SMOTE	G-mean, f-measure, accuracy	Cluster Based Synthetic Oversampling obtained greater result on other methods	20 real world data sets
[12]	Oversampling scheme based on weight setting and classification ranking	Accuracy	Achieved 90% accuracy for handling multiclass samples.	12 data sets from UCI repository are used
[31]	Synthetic over-sampling scheme	General model	Reliable speed for binary class data	26 bench mark data sets are used
[4]	G-SMOTE	p-value, G-metric, Accuracy	Novel data generation procedure outperformed in classification	13 UCI data sets
[28]	Safe-level- SMOTE	G-mean, AUC, f- measure	Achieved greater result in social media data analysis	Raw data sets are used
[34]	Reverse-SMOTE	G-mean, AUC, f- measure	Efficient for classification problem	3 imbalanced data sets from KEEL repository are used.

4. ROLE OF CLASSIFIER ON IMBALANCE DATA

This section explores the various classifiers used in this study to address the challenge of class imbalance. These classifiers are trained on datasets specifically designed to handle skewed data distributions. Research has consistently highlighted that imbalanced or unstructured data poses a significant challenge in DM, particularly when the class of interest has far fewer samples than the others. This imbalance can lead to biased predictions and reduced model performance, making it crucial to implement strategies that effectively mitigate these issues. Different classifiers are used to handle the class imbalance dataset. However, for the study purpose we were utilized some different classifiers as described as follows:

4.1. K-nearest neighbors

A well-liked and adaptable classification approach for imbalanced data is KNN [42]. To successfully manage the dataset's imbalance, nevertheless, a few factors and strategies must be used. Here are a few methods for applying KNN to data that is imbalanced: i) class balancing techniques: under sampling: extract instances belonging to the majority class to decrease the number of instances and increase the number of instances of the minority class. However, do not throw away the baby with bath water to avoid important data disappearing same way. Oversampling: more instances from the minority class should be synthesized by methods like SMOTE and ADASYN. Sampling techniques: there are a number of techniques such as giving more weights to the classes which are less represented, then making the model to focus more on the minority class; ii) distance metric selection: choose appropriate distance metrics that are sensitive to the nature of the data. Experiment with different distance metrics (e.g., Euclidean and Manhattan) to find the most suitable one for your dataset; iii) tune K value: experiment with different values of K (the number of nearest neighbors) to find the optimal value for your dataset. Too small a K may lead to overfitting, while too large a K may cause underfitting; iv) probabilistic output: obtain probabilities for each class prediction instead of direct class labels. This allows for adjusting the classification threshold to improve sensitivity or specificity based on the problem's requirements; v) algorithm tuning: experiment with different KNN variants or extensions that are suitable for imbalanced datasets, such as ENN or condensed nearest neighbors; vi) ensemble approaches: combine KNN with other classifiers or ensemble methods like bagging or boosting to enhance performance, especially for imbalanced datasets; vii) cross-validation and evaluation metrics: in order to reduce the risk of a high variance the following methods should be employed, for instance, stratified k-fold cross validation ensures that classes in the data division are proportionately split. In evaluating the

accuracy of a model, the best options are precision, recall, F1-score, AUC-ROC or area under the precision recall (AUC-PR) since they were developed to deal with issues related to class imbalance; and viii) post-processing: apply post-processing techniques to the KNN predictions, such as thresholding or cost-sensitive learning, to further optimize the performance based on the class imbalance.

By incorporating these strategies and adapting the KNN algorithm to may handle imbalanced data, you can improve the performance and robustness of your classification model for such datasets. Experimentation is important to find the best approach that fits your data. Careful consideration of your dataset's characteristics is key to achieving optimal results.

4.2. Deep belief network classifier

A deep belief network (DBN) is a type of artificial neural network made up of numerous layers of stochastic latent variables, or nodes, that have been generatively trained using unsupervised learning techniques. It is a specific type of deep learning model that has been used in computer vision, binary classification, class imbalance, natural language processing, and other ML applications [43]. A stack of restricted Boltzmann machines (RBMs) and a supervised learning layer (such as a SoftMax classifier) are the two primary parts of a DBN. An overview of the typical construction and training of a DBN for classification tasks is given as follows: i) stacking RBM: the initial stage in creating a DBN is to train several RBMs one after another. A visible layer (input layer) and a concealed layer make up each RBM. The probabilistic representations of the data at each layer are taught to the RBMs using unsupervised learning techniques like contrastive divergence; ii) greedy-layer wise training: each RBM is trained independently and in a "greedy" layer-wise manner for the RBMs. One RBM's output serves as the input for the following RBM in the stack; iii) fine tuning: after all the layers (RBMs) have been trained, supervised learning is used to adjust the DBN. To optimize a particular goal, such as classification, the weights and biases of the entire network are changed using a supervised learning technique, such as backpropagation; iv) adding classifier: it is typical to modify the top layer of the DBN (the final RBM or an additional layer) to act as a classifier. To allow the network to predict outcomes for various classes, a SoftMax layer is frequently added to this layer for classification tasks; and v) training the classifier: the classifier is then trained using the labelled data and supervised learning methods (for example, the SoftMax layer). To reduce the classification error, backpropagation is frequently employed to adjust the weights and biases towards the layer.

DBNs are quite robust models that are accustomed for learning DN architecture of the input data and for this reason DBNs are suitable for applications such as classification. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are mainly used for directions such as image recognition and sequential data respectively. However, it is pertinent to emphasize that the newer types and designs of deep learning structures and methodologies came after DBNs.

4.3. Decision tree

Decision tree is a fundamental machine-learning method used for both classification and regression problems. Recursively dividing the feature space according to feature values creates a model in the form of a tree structure. The goal is to categorize the data into homogeneous subgroups and base predictions on these subgroups. Dealing with class imbalance is crucial whether utilizing decision trees or any other ML algorithm. Imbalanced data can have an impact on decision trees, resulting in biases in favor of the dominant class [44]. They handle category and numerical data and are simple to use and understand. However, if the tree is allowed to grow too deep or the dataset is noisy, they can easily be overfit to the training data. Techniques for pruning are frequently employed to reduce overfitting. Some key aspects and considerations when using decision trees: i) pruning: two types of pruning are there, where, pre-pruning refers to particular kinds of halting the growth of the tree at a certain point and the post-pruning where branches are omitted during the construction of the tree; ii) feature importance: decision trees can provide information about feature importance, which can be useful for feature selection and understanding the impact of different features on the predictions; iii) handling imbalanced data: as discussed earlier, handling class imbalance is crucial when using decision trees. Strategies like class weights, cost-sensitive learning, and ensemble methods can be effective; iv) hyperparameters: adjusting hyperparameters like maximum depth, minimum samples per class, and minimum samples per training is crucial for optimizing the balance between model complexity and performance; and v) visualization: decision trees can be visualized, such that it leads to a clear understanding of the decision-making process. Decision trees are often used in the creation of other more accurate models such as the random forests and gradient boosting machines since decision trees have ability to handle challenges such as class imbalance, classification, feature selection, among others.

4.4. C4.5 classifier

C4.5 another popular decision tree-based classification technique can be applied to issues involving binary and many classes. However, C4.5 is susceptible to class imbalance, just like many other common

classification techniques. The model typically favors the majority class when handling the imbalanced datasets, which results in subpar performance for the minority class [45]. When utilizing the C4.5 classifier it's easy to understand address class imbalance by applying several strategies to lessen the impact of the inequity on the model's performance. Few methods to define the C4.5 classifier as follows: i) class weights: adjust the class weights through the training process to give higher weights to the minority class. This way, the classifier will pay more attention to correctly predicting the minority class instances [46]; ii) cost-sensitive learning: modify the algorithm to be cost-sensitive, where misclassifications of the minority class are regularized more heavily. This encourages the model to focus on minimizing errors for the minority class; iii) resampling techniques: employ oversampling (e.g., SMOTE) or under sampling to balance the class distribution. This can help in training a more balanced model; iv) ensemble methods: utilize ensemble techniques like bagging or boosting with appropriate modifications to handle imbalanced data. Bagging can be particularly effective in reducing the bias towards the majority class; v) threshold adjustment: adjust the classification threshold to increase sensitivity to the minority class. This can be done post-model training by evaluating the model on validation data and finding an optimal threshold; and vi) hybrid approaches: combine oversampling of the minority class with under sampling of the majority class to create a more balanced dataset for training. Advanced decision tree-based algorithms like random forests or XGBoost, which are enhancements of C4.5 and can handle imbalanced data more effectively.

5. EVALUATION METRICS

In ML, handling class imbalance is critical, especially in classification task when the classes are not distributed equally. It's possible that traditional precision is not a good assessment metric when working with imbalanced datasets. However, this study uses different evaluation metrics to measure the proposed techniques that are frequently applied to measure class imbalance techniques [47], [48], as follows.

5.1. Accuracy

Most common used performance metric in classification task is accuracy, but it may not always be appropriate for imbalanced datasets. It estimates the percentage ratio of points belonging to the class as well as to its complement that were correctly classified to the total number of points in a dataset. However, in scenarios with severe class imbalance, accuracy can be misleading, as a classifier can achieve a high accuracy score simply by predicting the majority class while failing to correctly identify minority class instances. This limitation highlights the need for alternative metrics, such as precision, recall, F1-score, and AUC-ROC, to provide a more reliable assessment of model performance in imbalanced classification problems. The formula of accuracy can be expressed as (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

5.2. Geometric mean

When it comes into the question of the method of dealing with imbalanced data, it is necessary to note that a frequent candidate for its analysis is the geometric mean, in particular, geometric mean is often used, for example, to calculate the imbalanced performance measure. This is often done to determine the geometric mean of the sensitivity (true positive rate) and specificity (true negative rate), which provides a balanced assessment of model performance across classes. The geometric can be expressed as (2).

$$Geometric\ mean = \sqrt{sensitivity * Specificity} \quad (2)$$

5.3. AUC

AUC is a popular evaluation metric applied in binary classification problems, particularly for assessing the performance of ML models in terms of discrimination and ranking ability. AUC quantifies how well a certain model separates instances of one class from those of another by depicting the true positive rate sensitivity against the false positive rate. AUC is a measure that has values within the in the range [0,1], where value of 1 means that the model is reliable and accurate. It evaluates how efficiently the model performs between two different classes, the positive and the negative one, as illustrated in Figure 8. Interpretation of AUC as follows: AUC=0.5: this score indicates the model exhibits equivalent predictions; AUC≥0.5: the generated model discriminates between the classes, indicating improved distinction; and AUC=1: the trained model perfectly identifies between the positive and negative classes.

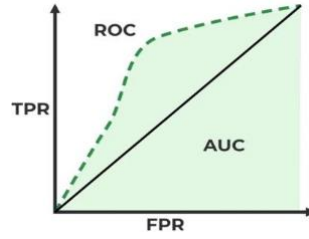


Figure 8. Example of AUC curve

5.4. Precision

Precision is a crucial evaluation metric used to evaluate the ML models in binary classification and multiclass classification. It is especially helpful when minimizing false positives is the main goal. The formula of precision can be expressed as (3).

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

Where, TP represent the cases where the model correctly identifies instances belonging to the positive class. False positives occur when the model incorrectly predicts an instance as positive when it actually belongs to the negative class. A precision score of 1.0 (or 100%) signifies that the model did not produce any false positive predictions. A precision score of 0.0 (or 0%) means that the model exclusively made false positive predictions.

5.5. Recall

Another key evaluation parameter in binary and multiclass classification problems is recall, which is sometimes referred to as sensitivity or true positive rate. The formula of recall can be expressed as in (4).

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

A recall of 1.0 (or 100%) indicates that the model identified all actual positive instances. A recall of 0.0 (or 0%) indicates that the model missed all actual positive instances.

5.6. F1-score

F1-score is a crucial measure for the binary classification models when the dataset is imbalanced. It provides the overall performance reporting of a model in which false positive rate and false negative rate are included through the average of the precision and recall statistics. The F1-score expresses the harmonic mean of simple accuracy and recall equation, and it's defined as (5)

$$F1\ Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (5)$$

The F1-score ranges between 0 and 1, where a higher value indicates a better balance between precision and recall.

5.7. Time complexity analysis

It is utilized as metric to analyze the time complexity of the proposed methods. Time complexity analysis is a key component to measure the algorithm design and analysis is time complexity analysis, which aims to comprehend how an algorithm's runtime grows with the quantity of the input. It offers useful information about an algorithm's effectiveness and scalability [49], [50]. In time complexity, Big O notation is used to estimate an upper bound of the running time of an algorithm.

6. RESEARCH GAPS

The research gaps for examining imbalanced dataset classification problems are provided as: i) employing the best data level strategy to effectively handle skewed data distribution to increase classification performance; ii) high redundancy dispute resolution; iii) reassigning overlapping instances, incorrect class tags, and outliers effectively; iv) oversampled and randomly generated examples that are

inaccurate; v) categorization of multi-class imbalanced data sets effectively; and vi) efficient handling of several inherent data traits that subtly influence how well imbalanced data sets are classified.

7. CONCLUSION

This paper summarizes the detailed survey on different class imbalanced techniques. A deep survey was conducted to show the different categorization strategies and their advantages and disadvantages in terms of improving performance in imbalanced data sets. The key contribution of the work is to understand the broad trends that affect classification or other ML tasks. A detailed investigation on SMOTE-based approaches, classifier-based methods, active learning-based methods, distribution-based methods, and cost-based methods creates particular attention on the problems that arise with imbalanced data sets. The work conducted a study on different classifier approaches were used to handle the imbalanced class problem. In parallel, the different evaluation metrics used to analyze the methods. The review of literatures reveals that classification of imbalanced data has received more attention in recent years since real-world data is frequently left out. When compared to the other classes (minority classes), the majority class (which contains the most samples) is not feasible to classify more accurately by using the traditional methods.

ACKNOWLEDGEMENTS

The authors would like to thank all the reviewers and editors for considering our work which is an appreciable worth and encouragement for conducting better research in near future assignments.

FUNDING INFORMATION

The authors declare that no funding was received for conducting this research.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration. This paper is solely based on the analysis and study conducted by the first author, with the guidance and supervision of the corresponding author, who provided necessary insights and revisions throughout the research process.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Putta Hemalatha	✓	✓	✓	✓	✓	✓		✓		✓			✓	
Geetha Mary Amalanathan	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nvestigation

R : **R**esources

D : **D**ata Curation

O : Writing - **O**riginal Draft

E : Writing - Review & **E**ditng

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

There are no financial or personal relationships that could have influenced the findings or interpretations presented. The authors have no conflicts of interest to disclose.

DATA AVAILABILITY

This study is a comprehensive review of existing research, relying exclusively on publicly available studies, datasets, and literature. All relevant sources and data utilized in this work are appropriately cited within the manuscript, ensuring transparency and reproducibility. No new data were generated or analyzed as part of this study.

REFERENCES




- [1] B. S. Arasu, B. J. B. Seelan, and N. Thamaraiselvan, "A machine learning-based approach to enhancing social media marketing," *Computers & Electrical Engineering*, vol. 86, Sep. 2020, doi: 10.1016/j.compeleceng.2020.106723.

- [2] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalanced data: a review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, Jun. 2009, doi: 10.1142/S0218001409007326.
- [3] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, "A survey on data preprocessing for data stream mining: current status and future directions," *Neurocomputing*, vol. 239, pp. 39–57, 2017, doi: 10.1016/j.neucom.2017.01.078.
- [4] W. Xie, G. Liang, Z. Dong, B. Tan, and B. Zhang, "An improved oversampling algorithm based on the samples' selection strategy for classifying imbalanced data," *Mathematical Problems in Engineering*, vol. 2019, no. 1, 2019, doi: 10.1155/2019/3526539.
- [5] C. Yang, G. Liu, C. Yan, and C. Jiang, "A clustering-based flexible weighting method in AdaBoost and its application to transaction fraud detection," *Science China Information Sciences*, vol. 64, no. 12, 2021, doi: 10.1007/s11432-019-2739-2.
- [6] T. Wongvorachan, S. He, and O. Bulut, "A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining," *Information*, vol. 14, no. 1, Jan. 2023, doi: 10.3390/info14010054.
- [7] S. Zhu and J. Wan, "Cost-sensitive learning for semi-supervised hit-and-run analysis," *Accident Analysis & Prevention*, vol. 158, Aug. 2021, doi: 10.1016/j.aap.2021.106199.
- [8] Z. Zojaji and B. Tork Ladani, "Adaptive cost-sensitive stance classification model for rumor detection in social networks," *Social Network Analysis and Mining*, vol. 12, no. 1, Dec. 2022, doi: 10.1007/s13278-022-00952-2.
- [9] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 1–11, Mar. 2015, doi: 10.5121/ijdkp.2015.5201.
- [10] S. Angra and S. Ahuja, "Machine learning and its applications: a review," in *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDACI)*, IEEE, Mar. 2017, pp. 57–60, doi: 10.1109/ICBDACI.2017.8070809.
- [11] G. Battineni, G. G. Sagaro, N. Chinatalapudi, and F. Amenta, "Applications of machine learning predictive models in the chronic disease diagnosis," *Journal of Personalized Medicine*, vol. 10, no. 2, 2020, doi: 10.3390/jpm10020021.
- [12] M. Deng, Y. Guo, C. Wang, and F. Wu, "An oversampling method for multi-class imbalanced data based on composite weights," *PLOS ONE*, vol. 16, no. 11, 2021, doi: 10.1371/journal.pone.0259227.
- [13] L. Yao and T.-B. Lin, "Evolutionary mahalanobis distance-based oversampling for multi-class imbalanced data classification," *Sensors*, vol. 21, no. 19, 2021, doi: 10.3390/s21196616.
- [14] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, 2018, doi: 10.1613/jair.1.11192.
- [15] X. Xu, W. Chen, and Y. Sun, "Over-sampling algorithm for imbalanced data classification," *Journal of Systems Engineering and Electronics*, vol. 30, no. 6, pp. 1182–1191, 2019, doi: 10.21629/JSEE.2019.06.12.
- [16] R. Sikora and S. Raina, "Controlled under-sampling with majority voting ensemble learning for class imbalance problem," in *Intelligent Computing*, Springer, Cham, 2019, pp. 33–39, doi: 10.1007/978-3-030-01177-2_3.
- [17] G. A. Pradipta, R. Wardoyo, A. Musdholifah, I. N. H. Sanjaya, and M. Ismail, "SMOTE for handling imbalanced data problem : a review," in *2021 Sixth International Conference on Informatics and Computing (ICIC)*, IEEE, Nov. 2021, pp. 1–8, doi: 10.1109/ICIC54025.2021.9632912.
- [18] X. W. Liang, A. P. Jiang, T. Li, Y. Y. Xue, and G. T. Wang, "LR-SMOTE — an improved unbalanced data set oversampling based on k-means and SVM," *Knowledge-Based Systems*, vol. 196, May 2020, doi: 10.1016/j.knsys.2020.105845.
- [19] S. Subha and J. G. R. Sathiaselalan, "Anomaly detection and oversampling approach for classifying imbalanced data using clubs technique in IoT healthcare data," *International Journal of Intelligent Engineering Informatics*, vol. 11, no. 3, pp. 255–271, 2023, doi: 10.1504/IJIEI.2023.133074.
- [20] D. Liu, S. Zhong, L. Lin, M. Zhao, X. Fu, and X. Liu, "Deep attention smote: data augmentation with a learnable interpolation factor for imbalanced anomaly detection of gas turbines," *Computers in Industry*, vol. 151, 2023, doi: 10.1016/j.compind.2023.103972.
- [21] A. A. Khan, O. Chaudhari, and R. Chandra, "A review of ensemble learning and data augmentation models for class imbalanced problems: combination, implementation and evaluation," *Expert Systems with Applications*, vol. 244, Jun. 2024, doi: 10.1016/j.eswa.2023.122778.
- [22] D. Elreedy, A. F. Atiya, and F. Kamalov, "A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning," *Machine Learning*, vol. 113, no. 7, pp. 4903–4923, Jan. 2024, doi: 10.1007/s10994-022-06296-4.
- [23] R. Kumari, J. Singh, and A. Gosain, "Empirical review of oversampling methods to handle the class imbalance problem," in *Evolution in Computational Intelligence*, Springer, Singapore, 2023, pp. 35–48, doi: 10.1007/978-981-99-6702-5_3.
- [24] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," *International Journal of Computer Science and Network*, vol. 2, no. 1, Feb. 2013.
- [25] S. Rezvani and X. Wang, "A broad review on class imbalance learning techniques," *Applied Soft Computing*, vol. 143, Aug. 2023, doi: 10.1016/j.asoc.2023.110415.
- [26] S. Dhanalakshmi, S. Das, and R. Senthil, "Speech features-based parkinson's disease classification using combined SMOTE-ENN and binary machine learning," *Health and Technology*, vol. 14, no. 2, pp. 393–406, Mar. 2024, doi: 10.1007/s12553-023-00810-x.
- [27] E. F. Swana, W. Doorsamy, and P. Bokoro, "Tomek link and smote approaches for machine fault classification with an imbalanced dataset," *Sensors*, vol. 22, no. 9, Apr. 2022, doi: 10.3390/s22093246.
- [28] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-SMOTE: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem," in *Advances in Knowledge Discovery and Data Mining*, Springer, Berlin, Heidelberg, pp. 475–482, 2009, doi: 10.1007/978-3-642-01307-2_43.
- [29] H. Lee, J. Kim, and S. Kim, "Gaussian-based SMOTE algorithm for solving skewed class distributions," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 4, pp. 229–234, Dec. 2017, doi: 10.5391/IJFIS.2017.17.4.229.
- [30] P. Hemalatha and G. M. Amalanathan, "FG-SMOTE: fuzzy-based Gaussian synthetic minority oversampling with deep belief networks classifier for skewed class distribution," *International Journal of Intelligent Computing and Cybernetics*, vol. 14, no. 2, pp. 270–287, Apr. 2021, doi: 10.1108/IJICC-12-2020-0202.
- [31] A. Gosain and S. Sardana, "Handling class imbalance problem using oversampling techniques: a review," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, Sep. 2017, pp. 79–85, doi: 10.1109/ICACCI.2017.8125820.
- [32] J. H. Joloudari, A. Marefat, M. A. Nematollahi, S. S. Oyelere, and S. Hussain, "Effective class-imbalance learning based on smote and convolutional neural networks," *Applied Sciences*, vol. 13, no. 6, Mar. 2023, doi: 10.3390/app13064006.
- [33] R. Mohammed, J. Rawashdeh, and M. Abdullah, "Machine learning with oversampling and undersampling techniques: overview study and experimental results," in *2020 11th International Conference on Information and Communication Systems (ICICS)*, IEEE, pp. 243–248, Apr. 2020, doi: 10.1109/ICICS49469.2020.239556.




- [34] R. Das, S. K. Biswas, D. Devi, and B. Sarma, "An oversampling technique by integrating reverse nearest neighbor in smote: reverse-SMOTE," in *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, IEEE, pp. 1239–1244, Sep. 2020, doi: 10.1109/ICOSEC49089.2020.9215387.
- [35] N. A. A. Khleel and K. Nehéz, "A novel approach for software defect prediction using CNN and GRU based on SMOTE tomed method," *Journal of Intelligent Information Systems*, vol. 60, no. 3, pp. 673–707, Jun. 2023, doi: 10.1007/s10844-023-00793-1.
- [36] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, IEEE, pp. 1322–1328, Jun. 2008, doi: 10.1109/IJCNN.2008.4633969.
- [37] Y. Xu, C. Wu, K. Zheng, X. Niu, and Y. Yang, "Fuzzy-synthetic minority oversampling technique: oversampling based on fuzzy set theory for android malware detection in imbalanced datasets," *International Journal of Distributed Sensor Networks*, vol. 13, no. 4, Apr. 2017, doi: 10.1177/1550147717703116.
- [38] A. S. Hussein, T. Li, C. W. Yohannese, and K. Bashir, "A-SMOTE: a new preprocessing approach for highly imbalanced datasets by improving SMOTE," *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, 2019, doi: 10.2991/ijcis.d.191114.002.
- [39] X. Yang, Q. Kuang, W. Zhang, and G. Zhang, "AMDO: an over-sampling technique for multi-class imbalanced problems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1672–1685, Sep. 2018, doi: 10.1109/TKDE.2017.2761347.
- [40] W. Feng *et al.*, "Dynamic synthetic minority over-sampling technique-based rotation forest for the classification of imbalanced hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2159–2169, Jul. 2019, doi: 10.1109/JSTARS.2019.2922297.
- [41] N. U. Niaz, K. M. N. Shahariar, and M. J. A. Patwary, "Class imbalance problems in machine learning: a review of methods and future challenges," in *Proceedings of the 2nd International Conference on Computing Advancements*, New York, NY, USA: ACM, pp. 485–490, Mar. 2022, doi: 10.1145/3542954.3543024.
- [42] H. A. A. Alfeilat *et al.*, "Effects of distance measure choice on k-nearest neighbor classifier performance: a review," *Big Data*, vol. 7, no. 4, pp. 221–248, Dec. 2019, doi: 10.1089/big.2018.0175.
- [43] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, p. 27, Dec. 2019, doi: 10.1186/s40537-019-0192-5.
- [44] J. Ren and F. Liu, "A novel approach for software defect prediction based on the power law function," *Applied Sciences*, vol. 10, no. 5, Mar. 2020, doi: 10.3390/app10051892.
- [45] J. Zhai, S. Zhang, and C. Wang, "The classification of imbalanced large data sets based on mapreduce and ensemble of ELM classifiers," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 3, pp. 1009–1017, Jun. 2017, doi: 10.1007/s13042-015-0478-7.
- [46] S. Belarouci and M. A. Chikh, "Medical imbalanced data classification," *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, no. 3, pp. 116–124, Apr. 2017, doi: 10.25046/aj020316.
- [47] T. Le, M. T. Vo, B. Vo, M. Y. Lee, and S. W. Baik, "A hybrid approach using oversampling technique and cost-sensitive learning for bankruptcy prediction," *Complexity*, vol. 2019, no. 1, Jan. 2019, doi: 10.1155/2019/8460934.
- [48] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, May 2017, doi: 10.1016/j.eswa.2016.12.035.
- [49] A. Anggrawan, H. Hairani, and C. Satria, "Improving SVM classification performance on unbalanced student graduation time data using SMOTE," *International Journal of Information and Education Technology*, vol. 13, no. 2, pp. 289–295, 2023, doi: 10.18178/ijiet.2023.13.2.1806.
- [50] G. Douzas, F. Bacao, J. Fonseca, and M. Khudinyan, "Imbalanced learning in land cover classification: improving minority classes' prediction accuracy using the geometric SMOTE algorithm," *Remote Sensing*, vol. 11, no. 24, 2019, doi: 10.3390/rs11243040.

BIOGRAPHIES OF AUTHORS



Putta Hemalatha    is a research scholar in the School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India, and pursuing Ph.D. degree in the field of data mining. Her areas of interest include, artificial intelligence, deep learning, machine learning, and soft computing. She authored various research papers in the field of machine learning and data mining. Her publications in journals and conference are above than 6. She is life time member of professional bodies such as Indian Science Congress (ISC) and Indian Society for Technical Education (ISTE). She can be contacted at email: hema.keshav10@gmail.com.



Geetha Mary Amalanathan    received her Ph.D. from Vellore Institute of Technology, Vellore, India. She has completed M.Tech. in computer science and engineering from VIT and B.E. from University of Madras, Tamil Nadu, India. She is working for VIT as Professor. She was awarded Merit Scholarship for her best academic performance for the year 2007–2008 during her M.Tech. study. She has authored more than 20 journal and conference papers. She has authored book chapters in the area of data mining and artificial intelligence. Her research interests include security for data mining, databases, and intelligent systems. She works to empower healthcare management using computer science. She is associated with many professional bodies like IACSIT, CSTA and IAENG. She can be contacted at email: geethamary.a@gmail.com.