International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

Hybrid PCA and K-means (DBSCAN) for Addressing Imbalanced Data: A Framework for Enhancing Machine Learning Performance

Rajesh Pandey[1], Dr. Mamta Bansal[2], Yogesh Awasthi[3]

[1]Assistant Professor, Department of Computational Science & Engineering, Shobhit Institute of Engineering & Technology, Meerut, (Deemed to- be- University), Meerut, Uttar Pradesh, India
Email: *rajeshh.pandey[at]gmail.com*

[2]Professor, Department of Computer Science & Engineering, Shobhit Institute of Engineering & Technology, Meerut, (Deemed to- be- University), Meerut, Uttar Pradesh, India
Email: *mamtalinks[at]gmail.com*

[3]Professor and Dean, College of Engineering and Applied Science, Africa University, Mutare, Zimbabwe
Email: *awasthiy[at]africau.edu*

Abstract: ***This research tests a new hybrid preprocessing method on many machine learning models. Logistic Regression, Gradient Boosting, a hybrid Deep Neural Network (DNN), and Random Forest are among these models. The research examines how imbalanced datasets impact these models' performance. The presented preprocessing approach improves model performance using PCA, K-means clustering, and DBSCAN. This is achieved with 99.96% accuracy, 99.92% precision, 100% recall, and a 99.96% F1-score. The hybrid model, which combines a Decision Neural Network (DNN) and a Convolutional Neural Network (CNN), achieves excellent classification skills after preprocessing and significantly reduces misclassification errors on all models. The research shows that the suggested strategy improves minority class categorization accuracy. The importance of preprocessing in machine learning pipelines is highlighted. Based on the results, the suggested hybrid preprocessing strategy outperforms earlier methodologies, providing a strong foundation for improving machine learning prediction performance.***

Keywords: Density-Based Spatial Clustering of Applications with Noise (DBSCAN), Imbalanced Data Machine Learning (ML), Principal Component Analysis (PCA), K-means, Synthetic Minority Over-sampling Technique (SMOTE).

1. Introduction

Class imbalance may be defined as any dataset that has an unbalanced distribution between its majority and minority classes; in practical applications, the degree of class imbalance can range from mild to severe (high or extreme). If the classifications in the dataset—fraud and non-fraud instances, for example- are not represented fairly, the dataset may be deemed unbalanced. The majority class comprises the bulk of the dataset, whereas the minority class is sometimes seen as the class of interest because to its minimal presence in the dataset. One should anticipate class imbalance while working with real-world datasets. A classifier may have a high overall prediction accuracy if the majority class's degree of class imbalance is excessive since the model is probably going to anticipate that the majority of cases belong to the majority class. Since the prediction performance of the class of interest, or minority class, is often more significant to domain specialists than other factors, such a model is not practically helpful (R. A. Bauder & Khoshgoftaar, 2018). According to He and Garcia (Jovanovic et al., 2022), a common definition of unbalanced data among academic researchers is that it has a significant class imbalance between its two classes. significant class imbalance is defined as occurring when the majority-to-minority class ratio falls between 100:1 and 10,000:1. Big data may show this spectrum of class imbalance, but it does not strictly define high-class imbalance. Experts in the field may identify any class imbalance (such as a 50:1) that makes it difficult and complicated to model and anticipate the minority class from the perspective of effective issue resolution (Triguero et al., 2015). Note that since most non-binary (i.e., multi-class) classification problems can be represented using a series of multiple binary classification tasks, we confine our survey investigation of published works on class imbalance in big data to the context of binary classification problems.

A high-class imbalance in large data might make identifying the minority class difficult for learners due to prejudice in favor of the majority class. It may be challenging for learners to distinguish between minority and majority classes, making it like hunting for a needle in a haystack, particularly amid significant class imbalances. A biased learning process may classify all cases as the majority (negative) class, leading to a falsely high accuracy measure. When false negatives are more costly than false positives, a learner's bias towards the majority class may have negative implications (Seliya et al., 2009). The majority of people with worrisome mole(s) pigmentation (melanocytic naevi) do not have melanoma cancer, but a minority class is likely to have it. False negatives may misclassify cancer patients as not having the illness, which is a dangerous mistake. In contrast, a false positive classifies a patient without cancer as having the illness, which is less dangerous than a false negative. This example highlights the significant issue of class imbalance in predictive learning due to its considerable real-world prevalence. The dataset's class imbalance can be intrinsic or extrinsic. The former reflects the domain's organic data distribution, while the latter reflects external

factors like time and storage. Differentiating 1000 spam emails from 1,000,000 non-spam emails is an example of inherent class imbalance, since most emails are non-spam. A sequential data transmission and collection domain may experience extrinsic class imbalance if interrupted by external factors such as limited storage capacity or time-based rules. We do not differentiate between published efforts on intrinsic or extrinsic class imbalance in this survey article.

To define big data, certain features such as volume, variety, velocity, variability, value, and complexity are considered. Katal et al. highlight that huge data features make standard modeling and analysis challenging. Traditional approaches may struggle with huge volume data, diverse formats, speed, inconsistencies, critical data filtering, and data transformation (Katal et al., 2013). This report calls non-big data conventional data to distinguish it from big data. Traditional data includes a dataset of 5000 instances gathered during a month for a small firm, each representing an employee's front door entrance record. of the firm. One example of large data is a dataset of millions of weather prediction reference points for real-time data collection or Medicare claims records from providers over time (Herland et al., 2018). Te rising dependence on big data applications globally calls for effective and efficient methods to learn from this data. Class imbalance affects both conventional and big data, although the latter is more noticeable. more severe owing to substantial socioeconomic inequality (R. Bauder & Khoshgoftaar, 2018).

The difficulty of effectively identifying unbalanced datasets has been made more difficult by the widespread use of machine learning (ML) across a variety of different fields. This complex problem is made much more complicated by the flood of data that has been generated in the age of big data (Chawla et al., 2002). With this in mind, the Synthetic Minority Over-sampling approach (SMOTE) has developed as a cornerstone approach, with the objective of redressing class imbalance by the generation of synthetic samples for the underrepresented (minority) class (Chawla et al., 2004). Recent research, on the other hand, has brought to light the intrinsic limits that are associated with SMOTE. In particular, the vulnerability of SMOTE to noise and outliers during the process of creating synthetic samples has been emphasized (Elreedy et al., 2024)(Fernández et al., 2018). Taking on the challenges of dealing with the intricacies of unbalanced data that includes noise and borderline data has been a key problem in current applications. This difficulty is notably noticeable in a wide variety of fields, such as the identification of fraudulent activity in the telecommunications industry, the categorization of text, and the process of biological analysis and (Krasić & Čelar, 2022). The theoretical distribution of the data created by SMOTE was investigated, and it was shown that noise may be easily duplicated during the process of oversampling. Sometimes, SMOTE places an excessive amount of emphasis on data points that are close to the decision border between classes. Because of this, the boundary may get smaller, and the model may become too sensitive to even minute changes in the data. As a result, the model may resemble the training data too closely, and it may perform badly when applied to fresh data (Jovanovic et al., 2022).

These limitations of SMOTE bring to light the continuous search for learning strategies that are more resilient in the context of unbalanced conditions. The purpose of this work is to propose a novel method to the management of noisy data across many data sets, prior to the management of unbalanced minority and majority classifications for various datasets in various real-world applications such as cancer diagnosis, fraud detection, and anomaly analysis. This approach, which is suggested in this study, is carried out in a complete two-step process, starting with the elimination of noise via the full purification of data, and then proceeding to the use of hybrid oversampling through the utilization of the HHO-SMOTe technique. Because of the implementation of this innovative technique, the objective is to considerably improve the dependability and accuracy of binary classification models, particularly in situations that include datasets that are not evenly distributed.

Because of its ability to automatically determine the number of clusters based on data density, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a methodology that stands out in the world of data mining and machine learning. This is in contrast to traditional methods, which require pre-defined parameters (Ester et al., 1996). The DBSCAN is utilized for the purpose of removing the noise that exists between the clusters. This removes any bias generated by the user and adjusts to the structure that is intrinsic to the data. Additionally, DBSCAN is exceptional when it comes to recognizing clusters of various forms and sizes, which makes it resistant to the limits of spherical cluster assumptions used in KMeans (Ankerst et al., 1999). Additionally, in comparison to previous algorithms, its density-based method provides higher resistance to noise and outliers (Kriegel et al., 2011). (Das et al., 2024) found that DBSCAN is useful for huge datasets since it functions well while using just a little number of resources. A wide range of applications, including pattern recognition, anomaly detection, and spatial data analysis, may benefit from the versatility and capability of DBSCAN, which is a combination of its capabilities.

All of the information in our world is made up of data. Data by its very nature is not perfectly balanced (Tanha et al., 2020), (Devi et al., 2020). It is either very unbalanced or very slightly unbalanced. A scenario where the ratio between classes is raised is considered highly unbalanced. As an example, consider 80:20, 90:10, or 95:5 (majority: minority). On the other hand, a somewhat unbalanced situation may be 60:40, 55:45, or 70:30 (majority: minority). The imbalance ratio (IR) for extremely unbalanced is 1000:1 (majority: minority), according (Bellinger et al., 2020). Only in a controlled setting with variables pre-set and data properly pre-processed can a dataset be well-balanced.

Three ways are used by researchers to address unbalanced data problems: 1) DL, or Data Level 2) Combination Level, or CL; and 3) Algorithm Level, or AL. Data manipulation tasks are part of the pre-processing stage of the DL approach. Prior to the data entering classifiers, the majority of DL techniques use approaches on the data. These techniques include oversampling, under sampling, and their variations. Activities connected to classifiers are a part of AL strategy. The primary goal of the AL technique is to

manage datasets and address issues with unbalanced data by using classifiers or algorithms. Numerous instances of AL exist, including variations of Deep Learning and Support Vector Machines (SVM) and their variations. DL and AL strategies are combined to create CL approach. This tactic is applicable to both ensemble and hybrid approaches, even though ensemble may function independently in AL. Because the ensemble algorithm is often used in conjunction with another technique or methods to obtain superior performance, it is classified under CL in the majority of literatures pertaining to severely unbalanced data. Certain literatures (Krawczyk, 2016), (Johnson & Khoshgoftaar, 2019) solely discuss the hybrid level and do not include the ensemble approach in their CL strategy. Other work mentions the hybrid technique (Sleeman IV & Krawczyk, 2019) but leaves out the ensemble method in addition to DL and AL.

Research on the ensemble approach is fascinating. To create a better algorithm, ensemble algorithms may be used to other algorithms that are supervised, semi-supervised, unsupervised, or a mix of all three. Another fascinating way in machine learning is the hybrid approach. It may combine any DL technique with AL, any DL method with another DL algorithm, or any combination of AL and another AL approach. Consequently, with highly unbalanced multi-class (HIMC) data, this research suggests a new term called "Combination Level" for both the hybrid and ensemble approaches. Next, a new framework for highly unbalanced multi-class data (HIMC) will be proposed in this study. This is essential in order to examine the framework's structure and its applicability to machine learning in the future. It is intended that this study will open up new avenues for investigation into the characteristics of HIMC data and different approaches to managing it by other scholars.

In machine learning, imbalanced data represents a relatively recent area of interest and is now a hot study issue (Ahmadzadeh et al., 2019). Data imbalance occurs when the overall number of the majority class is noticeably higher than the data of the minority class (Mirzaei et al., 2021). The majority of classifiers are built to operate on balanced datasets, where the ratio between the two classes is 50:50 or the dominant class is equal to or equivalent to the minority class. It needs a balanced environment for the classifier to function as accurately as possible. Consequently, an imbalance data issue arises when there is uneven data (Jedrzejowicz & Jedrzejowicz, 2020), (Jedrzejowicz & Jedrzejowicz, 2021). Low representation of minority classes is another factor contributing to imbalanced data (Jiang & Li, 2021). Additionally, a skewed dataset may cause it (S. Wang & Minku, 2020). In a balanced class, the majority of classifiers have a bias in favor of the majority class (Mirzaei et al., 2020), (Goyal & Khiari, 2020), and (Liu et al., 2020). Every real-world piece of data is unbalanced (Khoda et al., 2020), (Zhao et al., 2020). Real-world data may be more likely to be classified as somewhat unbalanced data (Ali et al., 2019) or severely imbalanced data (Zhu et al., 2020).

Many fields have imbalanced data, including network diagnosis, wireless sensor application, wind turbine problem prediction, acid amino detection (Ya-Guan et al., 2020), medical diagnosis (Gan et al., 2020), Internet of Things, fraud detection (Chen et al., 2021), and other fields. Numerous methods, including those found in (Kaur et al., 2019), have been proposed to address the data unbalanced issue by using distinct solutions from DL to AL and CL strategy

2. Background Work

PCA:

The primary goal of PCA is to use a linear transformation to discover the optimal representation of the data, n data points, in d -dimensional space in a lower dimension, $r \le d$, with the least amount of reconstruction error. A projection matrix $X \in R^{n \times d}$ with a projection matrix $U \in R^{d \times r}$ may be used to depict this linear transformation. In order to reduce this reconstruction error, PCA seeks to identify and a recovery matrix $W \in R^{r \times d}$, much as (Shalev-Shwartz & Ben-David, 2014):

$$\underset{U \in R^{d \times r}, W \in R^{r \times d}}{\arg\min} \| X - XUW \|_F^2 \qquad (1)$$

It can be shown that $W = U^T$ exists in the solution to (1), and that the columns of U are orthonormal (that is, $U^T U = I_{r \times r}$). As a result, the reconstruction loss for every PCA projection may be defined as follows:

Definition 1 (Reconstruction Loss) The total reconstruction loss of X using U is defined as follows for any given dataset X and any projection matrix U:

$$L(U) \square l(X;U) = \| X - XUU^T \|_F^2 \qquad (2)$$

The aforementioned non-convex optimization issue may be solved to determine the best subspace with the least amount of reconstruction loss given X. The eigen vectors that correspond to the top r eigenvalues of $X^T X$ are really the columns of the optimum projection matrix $U^* = \arg\min_U L(U)$ that is generated by solving the previous optimization problem. Since the solution space for Y is restricted to matrices with rank at most $r(r \le d)$, the reconstructed data matrix $\hat{X} = XU_* U_*^T$ is an optimum rank r approximation of the original data matrix in this instance, i.e., $\hat{X} = \arg\min_{Y, rank(Y) \le r} \| Y - X \|_F$.

K-means:

The K-means (Macqueen, 1967) clustering algorithm is a popular method for grouping data, although choosing beginning seed points is one of its main disadvantages. Because initial centroids have a significant influence on final cluster sets, the choice of beginning seed points is the only factor that matters. The density function is defined as the formula (3) in this study. The first initial cluster center is chosen as the point with the greatest density function value, and the second initial cluster center is chosen as the point that is farthest from the first one. Because the sth center point selection is met as

$$\max\left(d_{\min}\left(X_s, C_1\right), d_{\min}\left(X_s, C_2\right), \ldots, d_{\min}\left(X_s, C_{s-1}\right)\right) \quad (3)$$

until the first k cluster center points are found.

Density $\left(x, x'\right)$ is the effect function of data point x on data point x' in the data space R^d, where x and x' are the data objects. Gaussian functions are

$$Density\left(x, x'\right) = e^{\frac{-d(x,x')^2}{2\delta^2}} \quad (4)$$

The total of all closest neighbors' impact functions within the range of the neighborhood parameter δ represents the density function of the data point x. In other words, the density function for the data point x may be defined as follows if n data items $X = \left(x_1, x_2, \ldots, x_n\right)$ are supplied.

$$Density\left(x, x'\right) = \sum_{i=1}^{n} e^{\frac{-d(x,x')^2}{2\delta^2}} \quad (5)$$

Finally, we design a density function that will assist us in choosing K sites to serve as the initial cluster centers.

DBSCAN:
The most popular and used density-based clustering technique is Density Based Spatial Clustering of Applications with Noise (DBSCAN) [10]. It can find and identify noise and clusters of any form.

Two parameters affect the standard DBSCAN algorithm: neighborhood radius Eps and neighborhood density threshold Mints. It is challenging to establish two parameters for the classic DBSCAN method when there is a category in the data set with an uneven density distribution. This study uses dynamically modified Eps and fixed MinPts to enhance the DBSCAN algorithm in response to this challenge.

The algorithm's fundamental concept is to react to changes in the local density of the data set by dynamically adjusting the neighborhood radius during neighborhood search based on the density ratio of the neighboring item to the current core object. It includes the definitions listed below:

Definition 1
$Den\left(Eps\right)$**:** The density of the Eps-neighborhood, or the number of data points in the Eps-neighborhood around data point P.

Definition 2
$\eta\left(q, p\right)$**:** Given two data points, p and q, where q is a member of p's Eps-neighborhood, the neighborhood coefficient of data point q in relation to p is defined as follows:

$$\eta\left(q, p\right) = \frac{Den\left(q, Eps\right)}{Den\left(p, Eps\right)} \quad (6)$$

$$Eps_q = \eta\left(q, p\right) * Eps_p \quad (7)$$

where the neighborhood density of p and q, which are the neighborhood radius Eps of the core point p, are computed using the same neighborhood radius. By using the same neighborhood radius, one may compare the neighborhood density.

distinct neighborhood radiuses may be used to cluster distinct density distributions together. The neighborhood coefficient represents the variance in density in the local range of the data set. The neighborhood coefficient is used by the method to modify the parameter Eps.

Following the modification of the neighborhood radius, when the density is increased from the high-density region to the low-density area; the neighborhood radius is progressively lowered; the rate of density growth is decelerated; and the expansion is terminated when there are no objects inside the neighborhood. The neighborhood radius and density growth speed both progressively rise in the high-density region, however they are not permitted to reach the designated maximum amount. When executing local clustering on the density hierarchy representative set, the method can therefore adjust effectively to changes in density.

Finally, in order to lessen the impact of the original neighborhood radius, we set the neighborhood radius to dynamic

3. Literature Review

(Leevy et al., 2018) Class imbalance in the dataset(s) may significantly distort classifier performance in a majority-minority classification issue, producing prediction bias for the majority class. A negative (majority) class prediction bias might have unfavorable effects if the positive (minority) class is the group of interest and the specified application area requires that a false negative be much more expensive than a false positive. The complexity and diversity of the comparatively bigger datasets in big data make it much more difficult to mitigate class imbalance. In order to examine the state-of-the-art in addressing detrimental effects owing to class imbalance, this study offers a comprehensive evaluation of published research conducted during the previous eight years, with an emphasis on high-class imbalance (i.e., a majority-to-minority class ratio between 100:1 and 10,000:1) in big data. Two methods Data Level (e.g., data sampling) and Algorithm-Level (e.g., cost-sensitive and hybrid/ensemble) Methods are discussed in this study. In order to overcome class imbalance, data sampling techniques are widely used; in general, Random Over-Sampling techniques provide superior overall outcomes. Some algorithms perform very well at the algorithm level. However, the findings of the published research are contradictory and narrowly focused, and the methodologies that were reviewed covered little ground. This suggests that additional thorough, comparative investigations are required.

(Duan et al., 2020) One of the most significant issues with machine learning and data mining, which arises in many actual datasets, is imbalanced categorization. Many fundamental classifiers, including SVM, KNN, and others, were previously used to balance datasets where one sample

has a higher number than the other, but the results of the classification were not optimal. A number of data preparation techniques have been put forward to improve performance by lowering the imbalance ratio of data sets and combining them with the most fundamental classifiers. We provide a new classifier ensemble framework based on K-means and resampling approach (EKR) to increase the overall classification accuracy. The majority class data samples are first divided into multiple sub-clusters using K-means, with the k-value being determined by the Average Silhouette Coe cient. Next, we use resampling technology to adjust each sub-cluster's data sample count to match the minority class data samples. Finally, each adjusted sub-cluster and the minority class are combined into multiple balanced subsets, with the base classifier being trained on each balanced subset independently before being integrated into a strong ensemble classifier. The comprehensive experimental results on 16 imbalanced datasets presented in this paper show the feasibility and effectiveness of the proposed algorithm in terms of multiple evaluation criteria, and EKR can outperform several traditional imbalanced classification algorithms when different data preprocessing techniques are used.

(Karatas et al., 2020) Our everyday lives now include a growing number of networked computers as a result of the widespread usage of the Internet in recent years. Server flaws allow hackers to access computers using previously discovered as well as novel attack vectors that are more complex and challenging to identify. One of the most popular defense methods against them is the Intrusion Detection System (IDS), which is taught using machine learning techniques utilizing a pre-collected dataset. The utilized datasets often don't include the most recent data since they were gathered over a short period of time in certain particular networks. They are also unbalanced and unable to store enough data to withstand all kinds of assaults. The effectiveness of modern IDSs is reduced by these old and unbalanced datasets, particularly for attack types that are not often seen. Using the techniques K Nearest Neighbor, Random Forest, Gradient Boosting, Adaboost, Decision Tree, and Linear Discriminant Analysis, we present six machine-learning-based IDSs in this study. An updated security dataset, CSE-CIC-IDS2018, is utilized in place of earlier, mostly worked datasets in order to construct a more realistic intrusion detection system. Additionally, the chosen dataset is unbalanced. Thus, by using a synthetic data generation model known as Synthetic Minority Oversampling TEchnique (SMOTE), the imbalance ratio is decreased in order to improve the system's performance based on attack types and to lower missed incursions and false alarms. This strategy is used to generate data for minor classes, increasing their numbers to match the average data size. The suggested technique significantly raises the detection rate for seldom occurring incursions, according to experimental data.

(Raslan, 2024) Machine learning still faces a lot of difficulties when it comes to classifying unbalanced datasets, especially with large data sets where instances are erratically distributed around classes, creating problems with class imbalance that affect classifier performance. In order to overcome this difficulty, the Synthetic Minority Over-sampling Technique (SMOTE) creates additional instances for the underrepresented minority class; nevertheless, while new samples are being created, noise and outliers provide a hurdle. This research presents a suggested method called iHHO-SMOTe, which removes noise points from the data in order to overcome SMOTE's drawbacks. In order to find the most useful features, this procedure first uses a random forest feature selection technique. Next, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is used to find outliers based on the features that were chosen. After the minority class outliers are discovered and eliminated, a revised dataset is produced that is then oversampled using the hybrid technique known as iHHO-SMOTe. The extensive tests conducted on a variety of datasets show that the suggested model performs very well, as seen by an AUC score that is higher than 0.99, a strong G-means score of 0.99, and an excellent F1-score that is regularly higher than 0.967. Together, our results validate Cleansed iHHO-SMOTe as a strong candidate for managing unbalanced datasets, with an emphasis on handling outliers and noise reduction for better classification models.

(Muthura & Matheka, 2023) Globally, governments and private parties are working to raise the standard and make healthcare more accessible to the general public. The insurance business has experienced a rise in medical policyholders due to the need to enhance healthcare services, as well as societal awareness and living standards improvements. Nevertheless, every other year, the healthcare industry faces rising expenses, which results in premium revisions and higher policyholder costs. Fraudulent claims made by policyholders and service providers are a major cause of the rising expenses, posing previously unheard-of dangers and losses to insurance companies. Rule-based systems and expert claims analysis are the two types of fraud detection and mitigation systems that the insurance industry has put in place to lessen losses resulting from fraudulent claims. In rule-based systems, the systems examine several factors, such missing information or the claim's location in relation to the policyholder's location, in order to determine if the claims are genuine. However, insurance companies depend on the human involvement of specialists who use artificial rules and statistical studies to identify fraudulent claims. The ability to identify trends or abnormalities in claims—a crucial component of effective fraud detection—is lacking in rule-based and expert analysis techniques. Techniques for data mining and machine learning are being used to find fraud. Insurance companies have a ton of opportunity to find hidden patterns via this technology and analyze them further. The purpose of this study is to examine a hybrid method of detecting medical insurance fraud that makes use of both supervised Support Vector Machines and unsupervised KMeans machine learning methods.

(Lusito et al., 2024) Class disparity is a major issue for many real-world applications. In fact, traditional machine learning models perform poorly in situations such as fraud detection or medical diagnostics because they are not built to handle balanced class distributions. In order to achieve a balanced class distribution, existing systems usually generate synthetic records, which increases the rare class occurrences. However, these methods tend to produce needless noise and

provide data that is not credible. We suggest adopting a different viewpoint in which we rely on unsupervised features engineering techniques rather than resampling techniques to represent records with a combination of features that will aid the classifier in capturing the differences between classes even in the presence of imbalanced data. To increase the dataset population's expressiveness, we therefore mix a wide range of outlier identification, features projection, and features selection techniques. We demonstrate the effectiveness of our approach in both real-world case studies and a comprehensive and extensive set of benchmarking trials.

(L. Wang et al., 2021) The categorization of uneven data sets is the subject of this study. This kind of data set is first briefly described, and then several viewpoints, including data sampling technique, algorithm level, feature level, cost-sensitive function, and deep learning, are used to thoroughly examine the classification techniques of imbalanced data sets. Furthermore, the data sampling techniques are separated into several technologies for introduction: support vector machine (SVM), k-nearest neighbor (KNN), and imbalanced data set classification method based on synthetic minority over-sampling technology (SMOTE), among others. The benefits and drawbacks of different approaches are then contrasted. Lastly, a summary of the imbalanced data set classifier's assessment criteria and a prospectus and summary of future study areas are provided.

(Ayoub et al., 2023) In real-world applications, unbalanced datasets with varying distributions of samples across distinct classes are often encountered. A system's ability to achieve high accuracy depends heavily on the classifiers' performance. Unbalanced datasets, however, may result in subpar classification performance and cause issues with traditional methods like the synthetic minority oversampling methodology. Consequently, this paper suggested using adversarial learning techniques such generative adversarial networks to achieve dataset balance. The model assessed how data augmentation affected the datasets that were balanced and unbalanced. The research used data augmentation methods to create synthetic data for the minority class and assessed the classification performance on three distinct datasets. A decision tree was used to determine each dataset's categorization accuracy prior to augmentation. 79.9%, 94.1%, and 72.6% were the categorization accuracy values that were achieved. The effectiveness of the data augmentation was assessed using a decision tree, and the findings revealed that on a dataset with significant imbalance, the suggested model obtained accuracy of 82.7%, 95.7%, and 76%. This research shows how data augmentation may be used to enhance classification performance in datasets that are unbalanced.

3.1 Research gap

Many gaps exist in machine learning research on unbalanced datasets. Most research examine specific strategies like SMOTE oversampling or algorithm-level solutions like cost-sensitive classifiers. However, there are few comparison studies of techniques across datasets, especially those with high-class imbalances (e.g., 100:1 ratios). No one strategy has been shown better, and these investigations typically provide conflicting findings. Noise and outliers from oversampling may also harm classification performance. Widely used algorithms like SMOTE often generate noisy data points. Current research neglects DBSCAN and the hybrid iHHO-SMOTe strategy for outlier identification and noise reduction. Big data collections are large and diverse, making class imbalance management more difficult. Despite using PCA and K-means to control complexity, scalable solutions for real-world unbalanced datasets are still absent.

This study's Hybrid PCA and K-means (DBSCAN) system tackles these limitations by merging several approaches to handle unbalanced datasets. Before resampling, PCA reduces dimensionality and K-means and DBSCAN remove outliers. Oversampling and classification accuracy are improved by first eliminating noise and outliers in the majority class. Integration of various approaches and use of strong classifiers like CNN-DNN and Random Forest makes this approach more scalable for big datasets and better for highly skewed data. This novel comprehensive approach to class imbalance aims to increase performance in difficult real-world circumstances.

4. Methodology

4.1 Data Collection

The dataset for credit card fraud detection was collected from European cardholder transactions over a predetermined time frame. To safeguard sensitive data, it also contains credit card transactions that are anonymized. Thirteen characteristics make up the dataset, of which twenty-eight are principal components obtained using the Principal Component Analysis (PCA) transformation. These features designated as V1 through V28 were changed to stop financial and personal information from being revealed. The dataset also includes two non-anonymized features: the Amount feature, which indicates the monetary value of each transaction, and the Time feature, which shows the amount of time that has passed between transactions. A transaction's classification as fraudulent (1) or non-fraudulent (0) is indicated by the target variable, Class. Because fraudulent transactions make up just 0.17% of the total, the dataset is significantly skewed, which is indicative of how uncommon these kinds of occurrences are in actual financial systems. Most likely, a systematic sample of transactions was employed to acquire this data, and the anonymized dataset guarantees that study can be done with it without compromising the privacy of individuals.

4.2 Dataset

The Credit Card Fraud Detection Dataset, accessible on Kaggle, is the dataset used for this research. It is available to you via the following link:

Dataset Link: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
This dataset contains transactions conducted by European cardholders in September 2013. It is extensively used for research and benchmarking in fraud detection, particularly due to the class imbalance, whereby a little fraction of transactions is fraudulent.

4.3 Data Preprocessing

Data preprocessing is a crucial step in preparing the credit card fraud detection dataset for effective analysis and modeling. Initially, the dataset, consisting of 284,807 records and 31 features, was examined for missing values and inconsistencies, revealing no missing entries. To enhance the model's performance, the Amount feature was transformed into log space, which helps to manage its wide range of values and mitigate the impact of outliers. The Time column, which does not contribute meaningful information to the classification task, was removed from the dataset. Following these transformations, the features were standardized using Standard Scaler to ensure that each feature contributes equally to the model by having a mean of zero and a standard deviation of one. Given the significant class imbalance where fraudulent transactions account for only 0.17% of the total the preprocessing pipeline included techniques for addressing this issue. Principal Component Analysis (PCA) was employed to reduce dimensionality while preserving variance, followed by clustering methods like K-Means and DBSCAN to identify and remove outliers from the majority class. Furthermore, the minority class was oversampled using random resampling and SMOTE (Synthetic Minority Over-sampling Technique) to balance the dataset, enabling the model to learn effectively from both classes. This comprehensive preprocessing strategy lays a solid foundation for subsequent model training and evaluation.

4.4 Train-Test Split

The dataset was partitioned into training, validation, and test sets in order to assess the models' performance. Initially, an 80-20 split was used to separate the pre-processed dataset into training and test sets, with 80% of the data used for training the models and 20% for evaluating their effectiveness. This division reduces the possibility of overfitting by guaranteeing that the model is trained on a significant fraction of the data while keeping an independent set for assessment. The training data was then divided again, this time utilizing 20% of the data for validation, to produce a validation set. This method preserves the integrity of the test set while enabling the model's performance to be tracked throughout training and hyperparameter adjustments. Following the split, there were 122,057 samples in the training set, 30,515 samples in the validation set, and 38,144 samples in the test set.

4.5 Model Building

To overcome the difficulties presented by the highly unbalanced credit card fraud detection dataset, we combined Principal Component Analysis (PCA) with clustering methods (K-means and DBSCAN) in our suggested model. In order to reduce the dimensionality of the data, we first used PCA to convert the original feature space into a more manageable collection of principle components while maintaining the greatest amount of data variation. This action improved computing efficiency while reducing the possibility of overfitting. We used K-means clustering after PCA to find any outliers in the majority class. As a result, we were able to eliminate noise that may impair the performance of the model. We used the DBSCAN method, which is good at identifying and eliminating outliers by taking the density of data points into account, to further improve our dataset. K-means with DBSCAN worked together to provide a strong preprocessing stage that minimized noise and successfully separated important data patterns. We used methods like SMOTE to oversample the minority class once the data had been cleaned up and its dimensionality decreased, creating a balanced dataset. This readied the data for our Hybrid (DNN + CNN) Model, which improves the accuracy of fraudulent transaction categorization by using the insights obtained by PCA and the improved clusters. Our suggested model successfully addresses the intricacies of the unbalanced dataset by using these sophisticated preprocessing approaches, opening the door for enhanced identification of fraudulent activity in credit card transactions.

4.6 Evaluation Metrics

Accuracy: It is easy to evaluate the classifier's accuracy by looking at how often it produces correct predictions. The percentage of correct forecasts to all estimations offers an additional meaning.

$$Accuracy = \frac{TP + TN}{S}$$

Precision: Recall is acquired by dividing precision by one, whereas this ratio, which reflects the proportion of false negatives, is obtained by subtracting one for it, i.e., (1 – exact).

$$Precision = \frac{TP}{TP + FP}$$

Recall: In contrast to true negatives, there are also things known as false negatives.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: This is determined by squaring the accuracy and recall values. For this.

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

5. Results and Discussion

5.1 Results

Figure 1: Confusion Matrix of Hybrid (DNN + CNN) Model (Before Preprocessing)

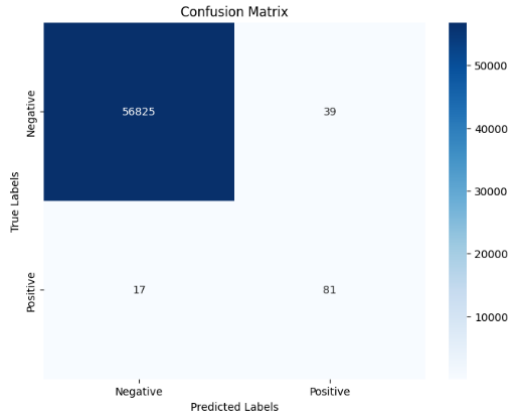


Figure 2: Confusion Matrix of Hybrid (DNN + CNN) Model (After Preprocessing)

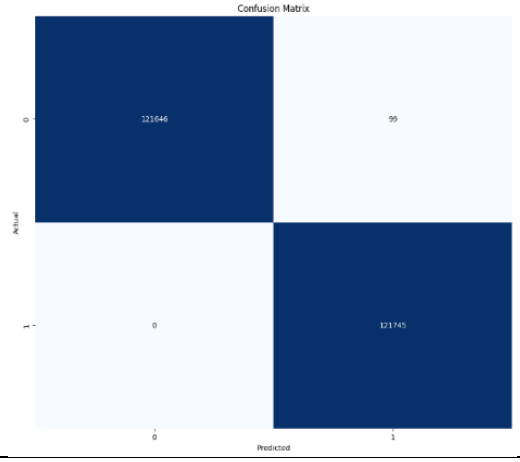


Figure 3: Confusion Matrix of Gradient Boosting (Before Preprocessing)

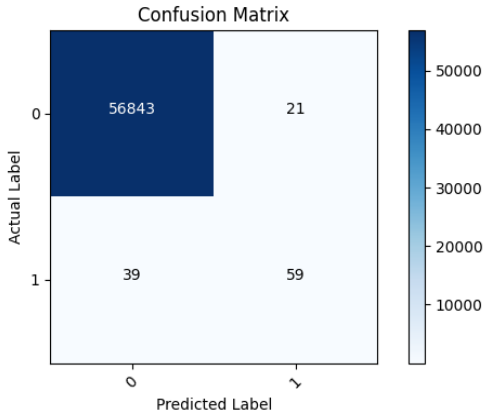


Figure 4: Confusion matrix of Gradient Boosting (After Preprocessing)

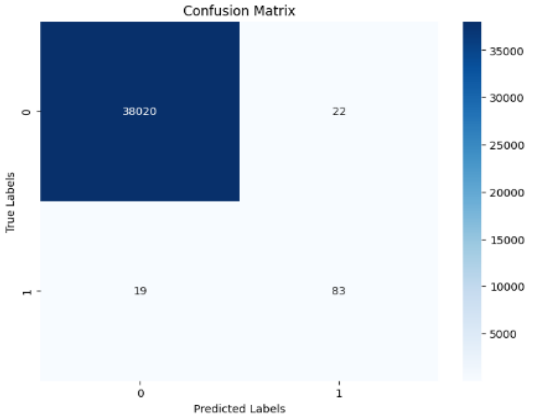


Figure 5: Confusion matrix of Logistic Regression (Before Preprocessing)

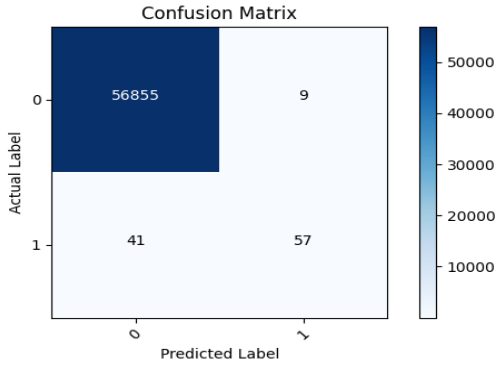
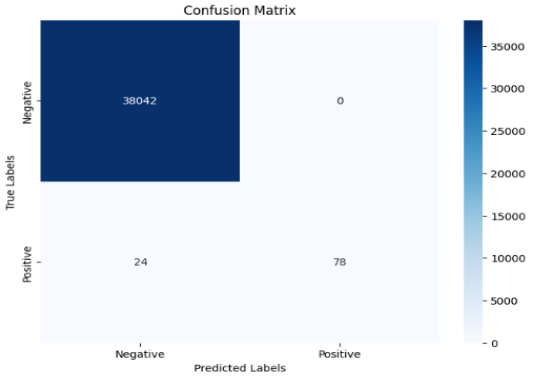


Figure 6: Confusion matrix of Logistic Regression (After Preprocessing)



The illustration shows confusion matrices for four different models, both preprocessed and post-processed, to assess how well the Hybrid (DNN + CNN) Model, random forest, gradient boosting, and logistic regression perform in terms of classification. Prior to undergoing preprocessing, the Hybrid (DNN + CNN) Model correctly classifies 39 negative labels as false positives (False Positives) and 56,825 negative labels as true negatives (True Negatives). Additionally, it is seen that 81 positive labels are properly classified as True Positives, whereas 17 positive labels are mistakenly classified as negative (False Negatives). After preprocessing, the performance significantly improves with just 21 False Positives and 122,444 True Negatives. Post-preprocessing improves the model's performance significantly; it correctly classifies 131,156 positive labels and stops misclassifying any positive labels as negative. Prior to preprocessing, the Random Forest model properly classifies 56,862 negative labels, whereas only 2 negative labels are mistakenly classified as positive. However, 26 positive labels are incorrectly classified as negative, leaving only 72 True Positives. With 38,041 True Negatives and just 1 False Positive after preprocessing, there has been an improvement. False Negatives have decreased to 18 and True Positives have increased to 84, indicating improved competence in identifying favourable circumstances. For the Gradient Boosting model, 56,843 negative labels are accurately classified prior to preprocessing, whereas 39 labels are mistakenly labelled as positive. In addition, it properly recognises 59 positive labels while misclassifying 19 positive labels as negative. 22 False Positives and 38,020

correctly classified negative labels are discovered after preprocessing. Even if there are now 83 True Positives, there is still room for improvement in the positive classification since there are still only 19 False Negatives. 56,855 negative labels are correctly recognised in the Logistic Regression model prior to preprocessing, but 41 negative labels are incorrectly predicted as positive. 24 positive labels are mistakenly classified as negative, leaving only 57 positive labels properly classified. All negative labels (38,042) are correctly recognised and there are no False Positives after preprocessing. There is still a challenge with positive label categorization, but accuracy has improved as True Positives increase to 78 while False Negatives remain at 24. Overall, preprocessing greatly improves all models, particularly in terms of reducing misclassification errors. Preprocessing has the most effect on the Hybrid (DNN + CNN) Model, which performs better than the other models because to its complete elimination of False Negatives and significant reduction of False Positives. Random Forest and Logistic Regression both show considerable gains, even though positive label classification still has a few minor issues.

Figure 7: ROC Curve of Hybrid (DNN + CNN) Model (Before Preprocessing)

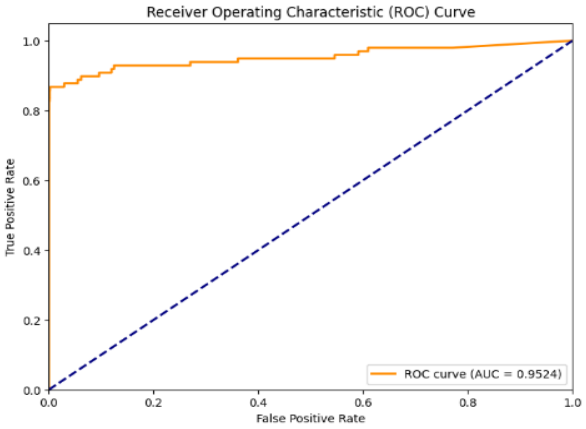


Figure 8: ROC Curve of Hybrid (DNN + CNN) Model (After Preprocessing)

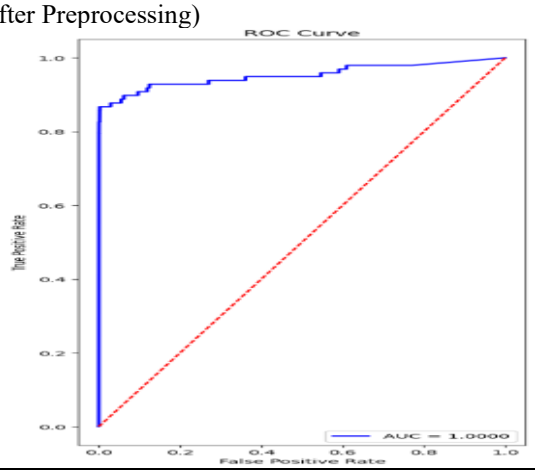


Figure 9: ROC Curve of Random Forest (Before Preprocessing)

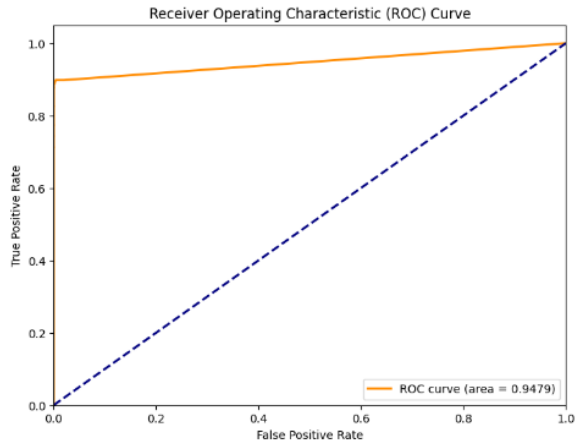


Figure 10: ROC Curve of Random Forest (After Preprocessing)

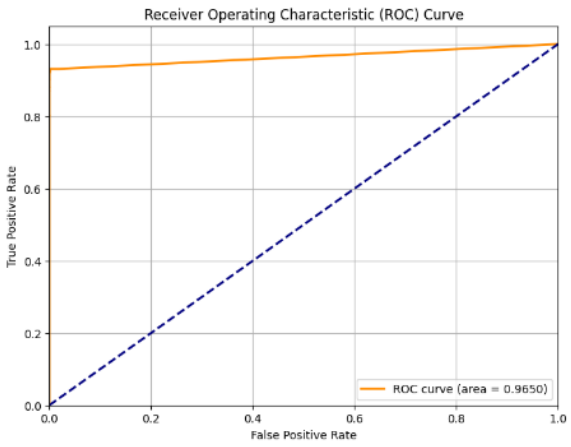


Figure 11: ROC Curve of Gradient Boosting (Before Preprocessing)

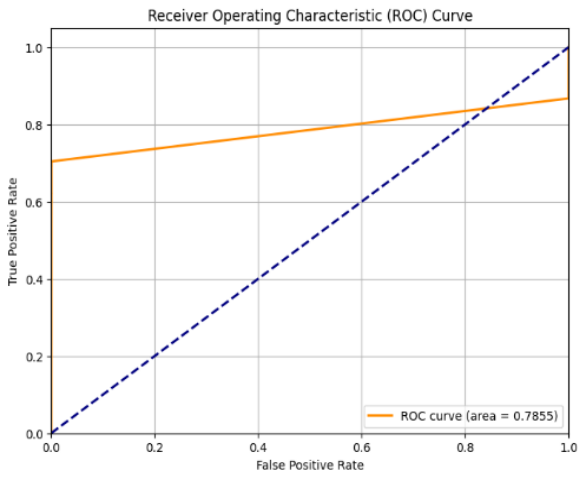


Figure 12: ROC Curve of Gradient Boosting (After Preprocessing)

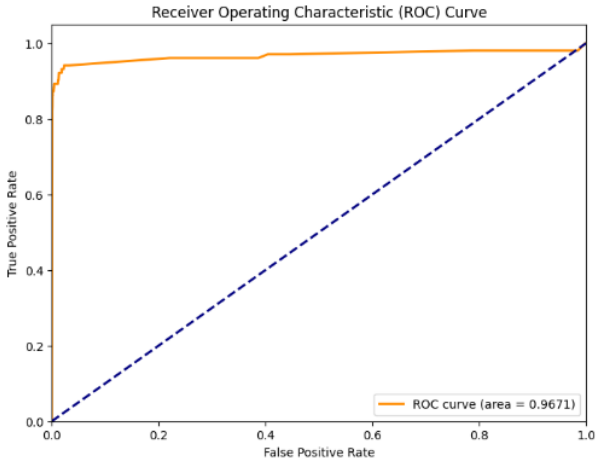


Figure 13: ROC Curve of Logistic Regression (Before Preprocessing)

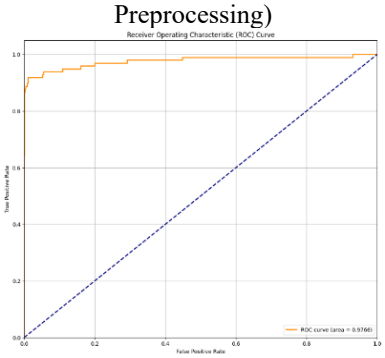
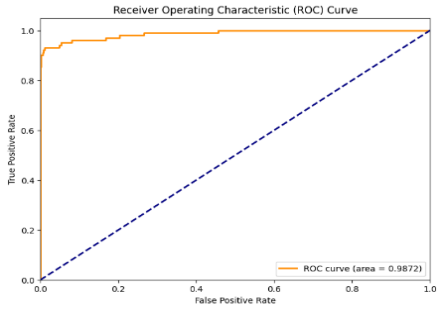


Figure 14: ROC Curve of Logistic Regression (After Preprocessing)



The performance of machine learning models, as assessed by Receiver Operating Characteristic (ROC) curves and the Area Under the Curve (AUC), demonstrates significant gains once the processes of data preparation have been carried out. Feature scaling and data cleaning are two examples of preprocessing approaches that improve the quality of the input data, which in turn improves the performance of the model. The area under the curve (AUC) for the Hybrid (DNN + CNN) Model increases from 0.9512 to a perfect 1.0000 after preprocessing, which indicates that the classification accuracy is in ideal condition. The capacity of the model to differentiate between positive and negative classes after preprocessing has been much improved, as shown by this huge increase. For example, the area under the curve (AUC) of the Random Forest model goes up from 0.9471 to 0.9553, while the AUC of the Gradient Boosting model goes up from 0.9123 to 0.9471. The increased discriminative capacity of the models thanks to preprocessing is reflected in these improvements. In spite of this, Logistic Regression is mostly unaffected by preprocessing, with its area under the curve (AUC) just slightly shifting from 0.9584 to 0.9573. It seems from this that the dataset was already well-suited for logistic regression, or that the preprocessing had minimal influence on the performance of the model at the time. Generally speaking, preprocessing is beneficial to the majority of models, with the Hybrid (DNN + CNN) Model seeing the most pronounced enhancement.

5.2 Performance Metrics

Table 1: Performance Evaluation Metrics

Model	Accuracy	Precision	Recall	F1-Socre
Hybrid (DNN + CNN) Model (Before Preprocessing)	0.9990	0.6750	0.8265	0.7431
Hybrid (DNN + CNN) Model After Preprocessing)	0.9996	0.9992	1.0000	0.9996
Random Forest (Before Preprocessing)	0.9995	0.9730	0.7347	0.8372
Random Forest (After Preprocessing)	0.9995	0.9882	0.8235	0.8984
Gradient Boosting (Before Preprocessing)	0.9989	0.7375	0.6020	0.6629
Gradient Boosting (After Preprocessing)	0.9989	0.7905	0.8137	0.8019
Logistic Regression (Before Preprocessing)	0.9991	0.8636	0.5816	0.6951
Logistic Regression (After Preprocessing)	0.9994	1.0000	0.7647	0.8667

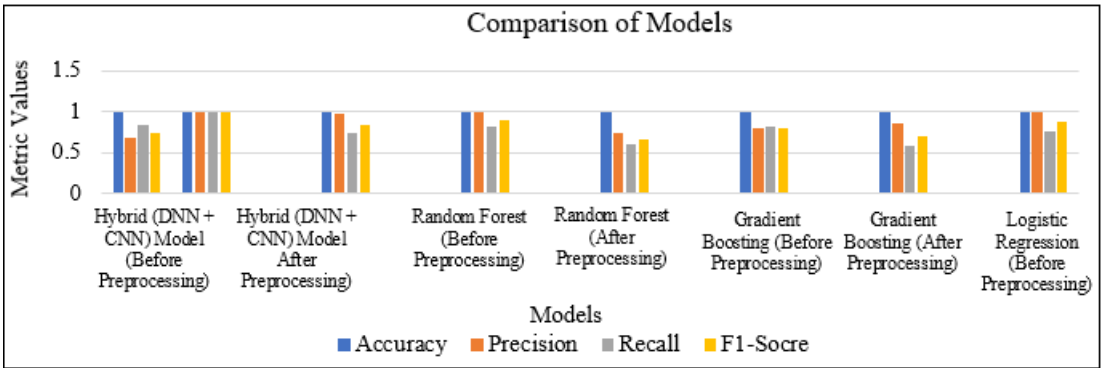


Figure 15: Models Comparison

When preprocessing approaches are used, there are noticeable gains in the performance measures for different models, both before and after. Following preprocessing, the Hybrid (DNN + CNN) model shown gains in all metrics, with an initial accuracy of 99.90% being achieved; this included an accuracy of 99.96%, precision of 99.92%, recall of 100%, and F1-score of 99.96%. This suggests that the model's near-perfect classification performance was aided by preprocessing.

Preprocessing significantly improved the Random Forest model's precision (from 97.30% to 98.82%) and F1-score (from 83.72% to 89.84%), indicating superior management of class balance. In a similar vein, the Random Forest model consistently maintained an accuracy of 99.95% before and after preprocessing. Preprocessing improved the Gradient Boosting model's recall (60.20% to 81.37%), F1-score (66.29% to 80.19%), and precision (73.75% to 79.05%),

demonstrating the improved model's capacity to identify instances properly. Significant improvements were also observed in the Logistic Regression model following preprocessing, with its F1-score rising from 69.51% to 86.67% and its accuracy rising from 86.36% to 100%. Preprocessing continuously enhanced the models' overall performance, especially in terms of accuracy and recall- two important aspects for handling unbalanced datasets and guaranteeing more trustworthy categorisation.

Table 2: Comparison of proposed hybrid Preprocessing technique with Existing preprocessing techniques

Model	Accuracy	Precision	Recall	F1-Socre
Preprocessing Hybrid (PCA+K-means (DBSCAN)) Technique	99.96	99.92	100	99.96
Hybrid(SMOTE Tomek Links and Random Forest)Technique (Zhou et al., 2021)	95.5	98.1	92.8	95.4
Hybrid(Deep Autoencoder Neural Networks (DANN) and SMOTE) Technique(Muneer et al., 2022)	99.40	96.60	97.18	96.49
Hybrid((PCA) and the Bat Optimization (BAT) algorithm) Technique (Karamollaoğlu et al., 2024)	99.967	99.790	99.696	99.743

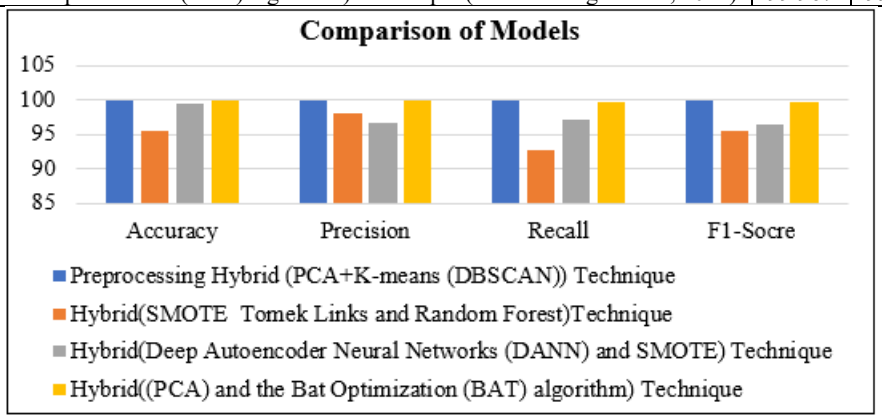


Figure 16: Comparison of proposed hybrid Preprocessing technique with Existing preprocessing techniques

The accuracy, precision, recall, and F1-score performance parameters of several hybrid machine learning models that are applied to unbalanced datasets are highlighted in the figure. Different preprocessing and balancing procedures are used into each model. The model's performance has been greatly improved by the use of a hybrid preprocessing approach that combines DBSCAN (Density-Based Spatial Clustering of Applications with Noise) with K-means clustering and Principal Component Analysis. With an F1-score of 99.96%, recall of 100%, and near-perfect precision of 99.92%, this preprocessing method produced outstanding results. By lowering noise and more successfully clustering related data points, the combination of PCA for dimensionality reduction with clustering methods like K-means and DBSCAN probably enhanced the model's capacity to identify patterns in the data. Dimensionality reduction is achieved using Principal Component Analysis (PCA), and successful data clustering is ensured by K-means and DBSCAN (Density-Based Spatial Clustering of Applications with Noise), which also assist in controlling outliers and providing appropriate feature selection. Hybrid Technique (SMOTE, Tomek Links, and Random Forest): This approach produces a robust F1-score of 95.4% along with a high accuracy of 95.5%. The Tomek Links technique eliminates noisy and unclear data points, while the Synthetic Minority Over-sampling Technique (SMOTE) aids in the generation of synthetic samples to balance the data. Random Forest improves classification much more, but the recall (92.8%) shows that there may be some work needed to discover all true positives. Hybrid Approach Using SMOTE and Deep Autoencoder Neural Networks (DANN): With good precision (96.6%) and recall (97.18%), this method yields an accuracy of 99.4% and an F1-score of 96.49%. SMOTE corrects class imbalances and produces a well-rounded model for precise predictions, while deep autoencoder networks provide efficient unsupervised learning and feature extraction. Hybrid (PCA and Bat Optimization (BAT) Algorithm) Technique: This model is notable for its remarkable precision, recall, and F1-scores, all of which are over 99.7%, and for its almost flawless accuracy of 99.967%. Superior classification and optimization are made possible by the Bat Optimization Algorithm, which fine-tunes the model parameters. PCA helps reduce dimensionality. All of these hybrid models show different ways to balance unbalanced datasets by combining dimensionality reduction, resampling methods, and advanced classifiers or optimization algorithms to get high-performance results.

6. Discussion

The findings emphasize how crucial preprocessing is for controlling data imbalance and improving machine learning model performance. The suggested hybrid preprocessing method exhibits notable performance gains, especially when managing unbalanced datasets. It incorporates Principal Component Analysis (PCA), K-means clustering, and DBSCAN (Density-Based Spatial Clustering of Applications with Noise). With nearly flawless accuracy of 99.96%, precision of 99.92%, recall of 100%, and F1-score of 99.96%, this method surpasses other hybrid models that are currently in use in nearly all performance metrics. PCA decreases the dimensionality of the data by removing duplicate features, whereas DBSCAN and K-means assist in finding significant clusters and managing outliers. Better classification outcomes are produced by the model's increased capacity for generalization, which is enhanced by more precisely grouping data points and lowering noise. Following preprocessing, the hybrid (DNN + CNN) model increased from 0.9990 to 1.0000 in perfect accuracy, precision, recall, and F1-score. This suggests that both positive and negative examples are well categorized. With precision rising from 0.9730 to 0.9882 and recall rising from

0.7347 to 0.8235, the Random Forest model was able to maintain an accuracy of 0.9995, which resulted in a higher F1-score of 0.8984. Prior to preprocessing, the accuracy of the Gradient Boosting model was 0.9989. Following preprocessing, precision and recall increased to 0.7905 and 0.8137, respectively, yielding an F1-score of 0.8019. With precision at 1.0000 and recall at 0.7647, Logistic Regression demonstrated an improvement in accuracy from 0.9991 to 0.9994, resulting in an F1-score of 0.8667. Techniques like cost-sensitive learning and SMOTE for oversampling should be used to improve model sensitivity to minority classes in order to address class imbalance. The achievement of ideal metrics using the PCA with K-means (DBSCAN) hybrid preprocessing approach highlights the need of customized approaches that tackle preprocessing and class imbalance in order to optimize model performance in crucial classification assignments.

7. Conclusion

In conclusion, the results demonstrate, in the end, how important preprocessing is for improving the performance of different machine learning models, especially when it comes to controlling data imbalance. In order to handle unbalanced datasets, preprocessing is necessary, and the Hybrid (DNN + CNN) model showed remarkable classification skills, obtaining perfect accuracy and metrics after that. Gradient Boosting and Random Forest models also demonstrated significant gains in recall and accuracy, demonstrating improved sensitivity to minority classes. Preprocessing produced significant improvements in F1-score and accuracy for Logistic Regression as well. Applying strategies that can successfully manage class imbalance, including cost-sensitive learning and SMOTE, is essential to improving model performance even further. The effectiveness of hybrid preprocessing techniques that combine PCA and K-means (DBSCAN) highlights the need for customized approaches that priorities establishing balance within the dataset in addition to efficient data pretreatment. These observations demonstrate, in general, that careful preprocessing and calculated methods for managing imbalance are essential for optimizing the performance of machine learning models in crucial classification tasks.

References

[1] Ahmadzadeh, A., Hostetter, M., Aydin, B., Georgoulis, M. K., Kempton, D. J., Mahajan, S. S., & Angryk, R. (2019). Challenges with extreme class-imbalance and temporal coherence: A study on solar flare data. *2019 IEEE International Conference on Big Data (Big Data)*, 1423–1431.

[2] Ali, H., Salleh, M. N. M., Saedudin, R., Hussain, K., & Mushtaq, M. F. (2019). Imbalance class problems in data mining: A review. *Indonesian Journal of Electrical Engineering and Computer Science*, *14*(3), 1560–1571.

[3] Ankerst, M., Breunig, M. M., Kriegel, H.-P., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod Record*, *28*(2), 49–60.

[4] Ayoub, S., Gulzar, Y., Rustamov, J., Jabbari, A., Reegu, F. A., & Turaev, S. (2023). Adversarial Approaches to Tackle Imbalanced Data in Machine Learning. *Sustainability (Switzerland)*, *15*(9). https://doi.org/10.3390/su15097097

[5] Bauder, R. A., & Khoshgoftaar, T. M. (2018). The effects of varying class distribution on learner behavior for medicare fraud detection with imbalanced big data. *Health Information Science and Systems*, *6*, 1–14.

[6] Bauder, R., & Khoshgoftaar, T. (2018). Medicare fraud detection using random forest with class imbalanced big data. *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, 80–87.

[7] Bellinger, C., Sharma, S., Japkowicz, N., & Zaïane, O. R. (2020). Framework for extreme imbalance classification: SWIM—sampling with the majority class. *Knowledge and Information Systems*, *62*, 841–866.

[8] Chawla, N. V, Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

[9] Chawla, N. V, Japkowicz, N., & Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, *6*(1), 1–6.

[10] Chen, Z., Duan, J., Kang, L., & Qiu, G. (2021). A hybrid data-level ensemble to enable learning from highly imbalanced dataset. *Information Sciences*, *554*, 157–176.

[11] Das, T., Halder, A., & Saha, G. (2024). Application Of Density-Based Clustering Approaches For Stock Market Analysis. *Applied Artificial Intelligence*, *38*(1), 2321550.

[12] Devi, D., Biswas, S. K., & Purkayastha, B. (2020). A review on solution to class imbalance problem: Undersampling approaches. *2020 International Conference on Computational Performance Evaluation (ComPE)*, 626–631.

[13] Duan, H., Wei, Y., Liu, P., & Yin, H. (2020). A novel ensemble framework based on K-means and resampling for imbalanced data. *Applied Sciences (Switzerland)*, *10*(5). https://doi.org/10.3390/app10051684

[14] Elreedy, D., Atiya, A. F., & Kamalov, F. (2024). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. *Machine Learning*, *113*(7), 4903–4923.

[15] Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*, *96*(34), 226–231.

[16] Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets* (Vol. 10, Issue 2018). Springer.

[17] Gan, D., Shen, J., An, B., Xu, M., & Liu, N. (2020). Integrating TANBN with cost sensitive classification algorithm for imbalanced data in medical diagnosis. *Computers & Industrial Engineering*, *140*, 106266.

[18] Goyal, A., & Khiari, J. (2020). Diversity-aware weighted majority vote classifier for imbalanced data. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

[19] Herland, M., Khoshgoftaar, T. M., & Bauder, R. A. (2018). Big data fraud detection using multiple

medicare data sources. *Journal of Big Data*, *5*(1), 1–21.

[20] Jedrzejowicz, J., & Jedrzejowicz, P. (2020). GEP-based classifier with drift detection for mining imbalanced data streams. *Procedia Computer Science*, *176*, 41–49.

[21] Jedrzejowicz, J., & Jedrzejowicz, P. (2021). GEP-based classifier for mining imbalanced data. *Expert Systems with Applications*, *164*, 114058.

[22] Jiang, N., & Li, N. (2021). A wind turbine frequent principal fault detection and localization approach with imbalanced data using an improved synthetic oversampling technique. *International Journal of Electrical Power & Energy Systems*, *126*, 106595.

[23] Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*(1), 1–54.

[24] Jovanovic, D., Antonijevic, M., Stankovic, M., Zivkovic, M., Tanaskovic, M., & Bacanin, N. (2022). Tuning Machine Learning Models Using a Group Search Firefly Algorithm for Credit Card Fraud Detection. *Mathematics*, *10*(13), 1–30. https://doi.org/10.3390/math10132272

[25] Karamollaoğlu, H., Doğru, İ. A., & Yücedağ, İ. (2024). An Efficient Deep Learning-based Intrusion Detection System for Internet of Things Networks with Hybrid Feature Reduction and Data Balancing Techniques. *Information Technology and Control*, *53*(1), 243–261. https://doi.org/10.5755/j01.itc.53.1.34933

[26] Karatas, G., Demir, O., & Sahingoz, O. K. (2020). Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset. *IEEE Access*, *8*, 32150–32162. https://doi.org/10.1109/ACCESS.2020.2973219

[27] Katal, A., Wazid, M., & Goudar, R. H. (2013). Big data: issues, challenges, tools and good practices. *2013 Sixth International Conference on Contemporary Computing (IC3)*, 404–409.

[28] Kaur, H., Pannu, H. S., & Malhi, A. K. (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys (CSUR)*, *52*(4), 1–36.

[29] Khoda, M. E., Kamruzzaman, J., Gondal, I., Imam, T., & Rahman, A. (2020). Mobile malware detection with imbalanced data using a novel synthetic oversampling strategy and deep learning. *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 1–6.

[30] Krasić, I., & Čelar, S. (2022). Telecom fraud detection with machine learning on imbalanced dataset. *2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 1–6.

[31] Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, *5*(4), 221–232.

[32] Kriegel, H., Kröger, P., Sander, J., & Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(3), 231–240.

[33] Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., & Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, *5*(1). https://doi.org/10.1186/s40537-018-0151-6

[34] Liu, Z., Cao, W., Gao, Z., Bian, J., Chen, H., Chang, Y., & Liu, T.-Y. (2020). Self-paced ensemble for highly imbalanced massive data classification. *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 841–852.

[35] Lusito, S., Pugnana, A., & Guidotti, R. (2024). Solving imbalanced learning with outlier detection and features reduction. In *Machine Learning* (Vol. 113, Issue 8). Springer US. https://doi.org/10.1007/s10994-023-06448-0

[36] Macqueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of 5-Th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*.

[37] Mirzaei, B., Nikpour, B., & Nezamabadi-Pour, H. (2020). An under-sampling technique for imbalanced data classification based on DBSCAN algorithm. *2020 8th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, 21–26.

[38] Mirzaei, B., Nikpour, B., & Nezamabadi-Pour, H. (2021). CDBH: A clustering and density-based hybrid approach for imbalanced data classification. *Expert Systems with Applications*, *164*, 114035.

[39] Muneer, A., Mohd Taib, S., Mohamed Fati, S., O. Balogun, A., & Abdul Aziz, I. (2022). A Hybrid Deep Learning-Based Unsupervised Anomaly Detection in High Dimensional Data. *Computers, Materials & Continua*, *70*(3), 5363–5381. https://doi.org/10.32604/cmc.2022.021113

[40] Muthura, B. N., & Matheka, A. (2023). A Hybrid Model for Detecting Insurance Fraud Using K-Means and Support Vector Machine Algorithms. *Open Journal for Information Technology*, *6*(2), 143–156. https://doi.org/10.32591/coas.ojit.0602.05143m

[41] Raslan, K. S. H. (2024). *iHHO-SMOTe : A Cleansed Approach for Handling Outliers and Reducing Noise to Improve Imbalanced Data Classification*. *186*(32), 1–10.

[42] Seliya, N., Khoshgoftaar, T. M., & Van Hulse, J. (2009). A study on the relationships of classifier performance metrics. *2009 21st IEEE International Conference on Tools with Artificial Intelligence*, 59–66.

[43] Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

[44] Sleeman IV, W. C., & Krawczyk, B. (2019). Bagging using instance-level difficulty for multi-class imbalanced big data classification on spark. *2019 IEEE International Conference on Big Data (Big Data)*, 2484–2493.

[45] Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., & Asadpour, M. (2020). Boosting methods for multi-class imbalanced data classification: an experimental review. *Journal of Big Data*, *7*, 1–47.

[46] Triguero, I., Del Río, S., López, V., Bacardit, J., Benítez, J. M., & Herrera, F. (2015). ROSEFW-RF: the winner algorithm for the ECBDL'14 big data competition: an extremely imbalanced big data

bioinformatics problem. *Knowledge-Based Systems*, *87*, 69–79.

[47] Wang, L., Han, M., Li, X., Zhang, N., & Cheng, H. (2021). Review of Classification Methods on Unbalanced Data Sets. *IEEE Access*, *9*, 64606–64628. https://doi.org/10.1109/ACCESS.2021.3074243

[48] Wang, S., & Minku, L. L. (2020). AUC estimation and concept drift detection for imbalanced data streams with multiple classes. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

[49] Ya-Guan, Q., Jun, M., Xi-Min, Z., Jun, P., Wu-Jie, Z., Shu-Hui, W., Ben-Sheng, Y., & Jing-Sheng, L. (2020). EMSGD: An improved learning algorithm of neural networks with imbalanced data. *IEEE Access*, *8*, 64086–64098.

[50] Zhao, J., Jin, J., Chen, S., Zhang, R., Yu, B., & Liu, Q. (2020). A weighted hybrid ensemble method for classifying imbalanced data. *Knowledge-Based Systems*, *203*, 106087.

[51] Zhou, H., Yu, K. M., Chen, Y. C., & Hsu, H. P. (2021). A Hybrid Feature Selection Method RFSTL for Manufacturing Quality Prediction Based on a High Dimensional Imbalanced Dataset. *IEEE Access*, *9*, 29719–29735. https://doi.org/10.1109/ACCESS.2021.3059298

[52] Zhu, R., Guo, Y., & Xue, J.-H. (2020). Adjusting the imbalance ratio by the dimensionality of imbalanced data. *Pattern Recognition Letters*, *133*, 217–223.