International Journal of Advances in Scientific Research and Engineering (ijasre)

E-ISSN : 2454-8006

DOI: 10.31695/IJASRE.2024.11.3

Volume 10, Issue 11
November - 2024

Class Prediction of High-Dimensional Data with Class Imbalance: Breast Cancer Gene Expression Data

Gideon Nyatuga Nyakundi, John Ndiritu, Joseph Mwaniki Ivivi , Timothy Kamanu

Department of Mathematics,

University of Nairobi,

Nairobi, Kenya

ABSTRACT

Breast cancer remains a leading cause of mortality among women worldwide, with early and accurate diagnosis being critical for effective treatment. Gene expression profiling has emerged as a powerful tool for understanding the molecular mechanisms of cancer and for developing predictive models. However, the high dimensionality and class imbalance inherent in gene expression data pose significant challenges for developing robust predictive models.

This study aims to develop and evaluate a predictive model for classifying breast cancer subtypes using high-dimensional gene expression data, addressing the challenges of class imbalance. The objective is to improve the accuracy and reliability of breast cancer subtype prediction to facilitate better diagnostic and treatment strategies.

A comprehensive dataset of breast cancer gene expression profiles was utilized, comprising numerous gene expression levels across multiple samples. To address class imbalance, resampling techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and Random Under-sampling, were employed. The machine learning algorithms employed included Support Vector Machines (SVM), Random Forests, and Neural Networks. The algorithms were trained and evaluated using cross-validation to identify the most effective model. The performance of these models was assessed based on metrics such as accuracy, precision, recall, and F1-score. The results indicate that the use of SMOTE in combination with SVM provided the most balanced and accurate predictions, with an F1-score significantly higher than models without resampling. The Random Forest algorithm also showed promising results, particularly in handling the high-dimensionality aspect of the data.

The study demonstrates that addressing class imbalance through advanced resampling techniques can significantly enhance the predictive accuracy of models trained on high-dimensional gene expression data. The findings underscore the potential of machine learning models, particularly SVMs and Random Forests, in improving breast cancer subtype classification.

Key Words: Breast Cancer, Gene Expression Profiling, High-Dimensional Data, Class Imbalance, Machine Learning, Cancer Prediction Models.

---

1. INTRODUCTION

Breast cancer is one of the most common cancers affecting women globally. It accounts for 12.5% of all global cancer cases. In Kenya, Breast Cancer accounts for about 23% of all cancer cases among women [1]. Previously, diagnosis, prognosis and treatment of such cancers heavily relied on clinical-pathological analysis of breast cancer tissues and other auxiliary lymph nodes. Traditionally, breast cancer is categorized into four sub-types based on a combination of tumour characteristics and expression of specific hormone receptors such as steroid hormone receptor and human epidermal receptor-2 (HER2) [2]. The disease prognosis and treatment recommendations heavily has often heavily relied on the features that would not perfectly predict the clinical outcomes. These treatment approaches have led to clinicians overprescribing immunotherapy to many patients even when the benefits of the treatment type were marginal [3].

Gene expression profiling presents an alternative method for clinical oncology researchers. This technique enabled researchers to detect differences in the gene expression among thousands of genes, leading to the creation of gene profiles for various types of cancer. Gene expression data provides information about the activity levels of genes in a particular biological sample [4]. This data is crucial for understanding how genes function and how their activity correlates with various biological processes, including health and disease. It involves the translation of information that is encoded in a gene into gene products such as proteins, rRNA, tRNA, snRNA, or mRNA. As technology continues to evolve, the importance of gene expression data in the healthcare field continues to increase.

Existing research has shown that gene expression data present promising biomarkers that can be used for cancer prediction or prognosis [5]. According to Petinrin *et al*., the data can also be used to predict response to treatment, identify risk factors for cancer recurrence, determine the overall survival for patients, and intuitively develop treatments appropriate to the various classification of cancer [5]. With the increased generation of gene expression data in the biomedical field, two challenges have emerged. These include the high-dimensional nature and class imbalance of the data. Gene expression data are high-dimensional in nature, yielding samples with a high number of features but few observations. Additionally, data from gene expression with regards to cancer are class-imbalanced in that some classes are over-represented (majority classes), while the rest are highly under-represented (minority classes). This presents a challenge of analysis and redundancy of some of the features.

Historically, the approaches applied to the analysis of high-dimensional and class imbalanced gene expression data have been closely tied to advancements in both genomic and computational methods [6]. As genomic data keeps growing in volume and complexity, researchers have started to rely more on computational and machine learning methods for analysis. Class imbalance has also become more recognized alongside its complex compounding effect on high dimensionality of gene expression data.

While the use of data-driven analyses of oncology data presents promising rapid prediction of cancer sub-type, statistical analysis of imbalanced high-dimensional data poses several challenges. The first challenge is the classification problems in high-dimensional data with class imbalance. Standard classification methods, when applied to class-imbalanced data classification, produce classification rules that perform poorly in classifying minority class elements. While some studies have developed statistical methods addressing the class-imbalanced datasets and high-dimensional datasets, the problems are often considered separately, meaning the combined effect is not fully understood. Therefore, a gap exists in research studies investigating the extent to which existing methods dealing with high-dimensional data and class imbalance can be integrated. Therefore, this study aims to develop cancer prediction models for imbalanced high dimensional data.

1.1 Breast Cancer Gene expression data

Gene expression profiling of cancer has improved diagnosis and prediction of various types of cancers. According to Ram *et al*., the main approach in cancer gene profiling involves studying biomarkers in various conditions or tissues to allow identification of molecular patterns in gene expression [7]. The initial challenge is how to identify biomarkers for specific cancer subtypes based on the small p and large n curse of dimensionality problem [7]. Further, cancer is often heterogenous meaning even within a specific cancer type, such as breast cancer, there exists subtypes with unique clinical and biological features [8]. The heterogeneity of cancer types and subtypes make it difficult to delineate cases that would benefit from intense treatment approaches [8]. This complexity has necessitated coupling of clinical analyses with advanced data analysis approaches such as multigene assay.

Breast cancer consist of at least five sub-types, the category of which is based on the expression of specific receptors and underlying genes. Gene profiling approaches based on the expression of hormone receptor and HER2 status have enabled the classification of breast cancer into luminal A, luminal B, basal-like, normal-like, and HER2-rich subtypes [9]. The classification of breast cancer into these subtypes based on molecular approaches reveals patterns in single-nucleotide variations [9]. If a diagnosis of breast cancer is noted to have estrogen, progesterone, and HER-2 receptors absent, the conclusion is often a triple-negative breast cancer (TNBC). TNBC is a more aggressive type of breast cancer with high number of metastatic cases and mortality rates among the cancer subtypes [10]. Indeed, prognosis and survival are dependent on the breast cancer subtypes in addition to the stage and other individual characteristics

DOI: 10.31695/IJASRE.2024.11.3

[11]. The status of receptors is crucial for cancer treatment with TNBC being the most difficult to treat [12]. The problem of predicting TNBC which is a minority of the breast cancer subtypes, yet it is the most lethal with poor prognosis.

1.2 Cancer prediction models

Models for cancer prediction are mainly based on deep learning and machine learning techniques. Deep learning techniques employ strategies that work with any data type even complex data, requiring huge information, and computing power to give classifiers with improved accuracy [13]. The commonly used deep learning techniques such as convoluted neural networks and deep neural networks offer promise improved accuracy and precision in predicting classifiers [13]. Deep learning techniques based on multiple ensemble models have been used to predict cancer classifications based on gene expression profiling. In a study by Xiao *et al.*, the authors found that deep learning approaches that combine machine learning models in ensemble demonstrated improved accuracy and reduced bias associated with a single model when creating classifiers of lung, stomach, and breast cancers [14].

As with all deep learning techniques, several challenges abound the commonly used deep learning techniques when predicting cancer prognosis or creating classifiers based on gene expression data. Islam *et al*. noted that various deep learning techniques used in cancer subtype classification suffered from limited data sizes, high dimensionality-class imbalanced nature of gene expression data, complexity leading to costly computational time, requirement for pre-cleaning, and limited generalization [15]. The amount of patient data that is available is still inadequate compared to deep learning model requirement for accurate prediction [16]. Indeed, Kumar *et al.* noted a persistent high incidence of cancer and associated mortality despite advanced prediction techniques. Simpler and faster models that can address the challenges of high dimensional data with class imbalances are needed to aid rapid diagnosis and prediction [13].

Several machine learning algorithms have been used to predict cancer prognosis or classify the categories for diagnosis. Within the various machine learning algorithms for predictive models, differences in performance of the models have been documented. In a breast cancer prediction and classification study, Khourdifi and Bahaj compared the performance of random forest, naïve Bayes, support vector machines (SVM), and K-nearest Neighbours (K-NN) and noted that SVM gave better predictive accuracy of 97.8% [17]. Even within SVM techniques, differences abound. In a breast cancer predictive study comparing single SVM and SVM ensemble, Huang *et al*. found that SVM ensemble performed better with gradient boosting of RBF kernel and bagging when handling complex data [18]. Another study comparing the predictive accuracy of logistic regression and decision tree algorithms based on breast cancer data from malignant and benign cases showed decision tree classifiers had greater accuracy than logistic regression of 100% and 99.06% respectively [19].

Selecting machine learning algorithms with demonstrated superior predictive performance is insufficient to give models with high precision and accuracy, given the size and nature of cancer data. One approach to augment the strengths of algorithm selection is to pair classifiers in cancer prediction research [20]. For instance, Tirumala and Narayanan conducted a comparative study on the effectiveness of artificial neural networks (ANN) paired with sequential forward feature selection (SFFS), Naïve Bayes paired with SVM, AdaBoost paired with SVM, and J48 paired with SFFS [20]. Among the paired approaches, ANN paired with SFFS demonstrated the highest predictive accuracy (100%). Such combinations have better predictive powers where machine learning algorithms such as decision support system are coupled with random optimization [21].

1.3 High dimensional data with class imbalance

Current machine learning approaches struggle to accurately handle class-imbalanced data [22]. The ratio of imbalanced status affects the classifier's accuracy through skewed performance. Approaches to handle class-imbalanced data rely on kernel space oversampling of minority classes using techniques such as SMOTE [23]. Weighted kernel-SMOTE is an approach used with SVM classifier that has demonstrated improved performance of models against imbalanced data [23]. SMOTE is commonly used to address issues arising from class imbalances at the data level by random oversampling minority classes and under-sampling majority classes [24]. However, feature selection types based on group filters, while essential to predictive models, hold no significant improvement in

DOI: 10.31695/IJASRE.2024.11.3

addressing class imbalance problems [24]. Algorithm selection such as ensemble which combines several classifiers and hybrid methods have also been used to improve performance of predictive models handling class imbalanced data [25]. While algorithm level models for addressing class imbalance exist, inconsistencies in their performance calls for further research.

Biological data such as healthcare data on gene expression generated using high throughput techniques is often high-dimensional, in addition to being class imbalanced. Moon *et al*. observed that biological data generated on mass cytometry, single-cell RNA sequencing, Hi-C, and gut microbiome data require analysis and visualization approaches that can detect patterns in high-dimensional data [6]. Machine learning techniques available for data analysis were originality developed without the changes of high dimensional data in mind [26]. Current techniques for addressing the high dimensional problem in data rely on deep learning algorithms including convoluted neural networks [27]. The combined effect of high dimensionality and imbalance in data adds to the complexity of accurately making predictions using existing models. New models that address the bias in predictive analysis models are needed to improve cancer prediction and diagnosis.

2. RESEARCH METHODOLOGY

2.1 Data Description

The dataset selected for this study consisted of gene expression profiles related to breast cancer. Gene expression profiling is a powerful method for understanding the molecular basis of cancer, identifying different cancer subtypes, and discovering potential therapeutic targets [28]. The current study entailed data for 54,676 genes as variables, meaning the data is high-dimensional. In terms of sample size, the data consisted of 151 samples included in the dataset. Each sample represented a distinct instance of breast cancer, providing a broad basis for analysis and model training. For classification, the data samples were categorized into 6 classes. The classes represent different types or subtypes of breast cancer, each with distinct genetic expression patterns. The six cases included basal, luminal A, luminal B, cell-line, HER, and normal as shown in Figure 1. The classes were further subcategorized as either cell-line or non-cell-line. The cell-line class consisted of 14 counts and the non-cell-line had 137 counts, making the data frame class-imbalanced.

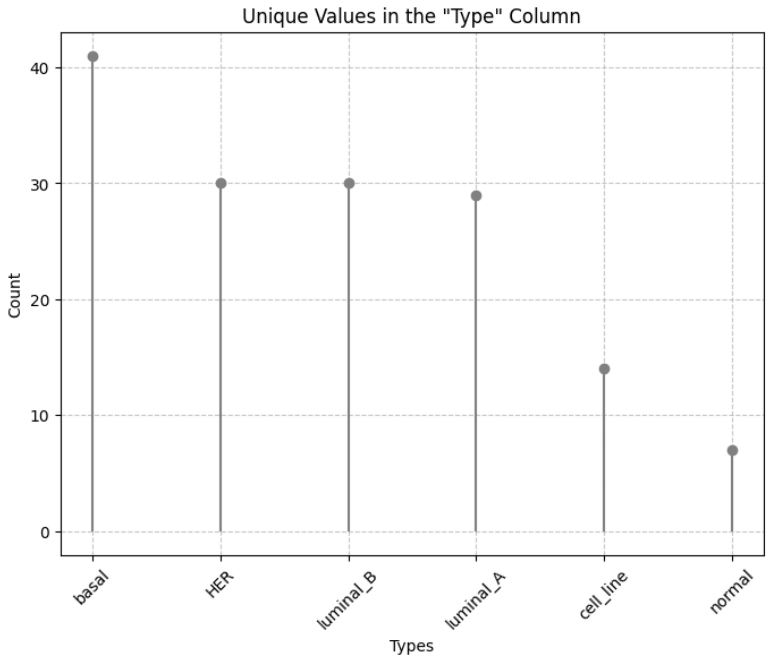

Fig. 1. Classes of data with respective counts

2.2 Methods

DOI: 10.31695/IJASRE.2024.11.3

Classification of minor class using imbalanced data

Feature selection

The features represent the gene expression values, which are measurements of the activity of genes in a biological sample. Also, the 'X' data contains the features of interest. The features were extracted by dropping the column named 'type' along the column's axis. This operation created a new data 'X' containing all the columns from "data" except for the 'type' column. Further, each row in "X" represented a sample, and each column represents a feature (such as the expression level of a specific gene). The target variable was the variable that was required to be predicted or classified based on the features. In this case, the target variable was the 'type' column, which typically represented the type or class of a sample (Cell-line and non-cell-line). The 'y' Series contained the target variable. It was extracted from the 'type' column of the data using indexing. Each element in 'y corresponded to the class label or category of the corresponding sample in "X". For the sample, *y*[1] was 1, meant that the corresponding sample represented by X [i] belongs to one class, and if `*y*[i]" is 0, it belongs to another class.

Sampling Approach

The data was divided into two parts with same variables, targets and features into 80% training and 20% testing data.

Classifiers

K-nearest Neighbour

K-nearest neighbour (KNN) was the first classifier used in this study. KNN was used for binary classification for the data by considering the majority class among the k-nearest neighbours of a given data point. For binary classification using KNN, the formula can be summarized as by determining the majority voting and Euclidean distance. For a given data point x, the class label was determined by a majority vote among its k nearest neighbours. In this regard, each neighbour's class contributes equally to the vote. For Euclidean distance, the distance between two data points X_1 and X_2 in the feature space can be calculated using the formula:

$$Euclidean\ distance = \sqrt{\{\sum_{i=1}^{n}(X_{1i} - X_{2i})^2\}}$$

Where X_{1i} and X_{2i} are the *i*-th features of data points X_1 and X_2, respectively.

Linear regression

In the binary data consisting of Cell-line and Non-cell-line classes, linear regression (LR) used as a binary classifier. The linear regression model predicts the probability that a sample belongs to the "cell-line" class. A threshold of 0.5 was then used to convert the probabilities into binary predictions. For binary classification, the formula for Linear Regression was expressed as:

$$P(y = 1|X) = \frac{1}{1 + e^{-w.x}}$$

Where; $P(y = 1|x)$ is the predicted probability that the sample belongs to the "cell-line" class given the features x; *x* is the feature vector, *w* is the weight vector (coefficients); *e* is the base of the natural logarithm, and · denotes the dot product between w and x.

The predicted probability was then converted into binary predictions using a threshold:

$$\hat{y} = \begin{cases} 1 & if\ P(y = 1|X) \geq 0.5 \\ 0 & otherwise \end{cases}$$

DOI: 10.31695/IJASRE.2024.11.3

Gene expression levels (features) were used to predict whether a sample belonged to the "cell-line" or "non_cell-line" class (target variable) with LR model learning the coefficients (weights) that best separate the classes based on the feature values.

Dummy Classifier

The Dummy classifier with the "Most Frequent" strategy predicts the class label that appears most frequently in the training data [29]. For the current study, the classes were "cell-line" and "non_cell-line". The classification formula for this strategy can be expressed as:

$$\hat{y} = Most\ frequent\ class$$

Here, y represents the predicted class for all data points, which is simply the most frequent class observed in the training data.

Assuming in the current data the class "cell-line" appears more frequently than "non_cell-line". The dummy classifier with the "most frequent" strategy would predict "cell-line" for all data points, regardless of their features. This strategy does not take into account any specific characteristics or patterns in the data; it solely relies on the class distribution observed in the training set.

Decision tree classifier

A decision tree was used to recursively partition the feature space into disjoint regions, with each region corresponding to a specific class label. To classify a new data point 2, it traverses the tree from the root node down to a leaf node based on the feature values of 2. The class label associated with the leaf node reached by *z* becomes the predicted class for 2. Here, the decision tree is denoted as T. The classification formula for a decision tree can be expressed as follows:

$$\hat{y} = \mathrm{T}(\mathrm{x})$$

Where, ŷ is the predicted class for the input data point 2 and $T(2)$ represents the prediction of the decision tree T' for data point 2. The decision tree classifier determines the predicted class for a data point by following the decision rules learned during training, which are encoded in the structure of the tree as shown in Figure 2.

DOI: 10.31695/IJASRE.2024.11.3

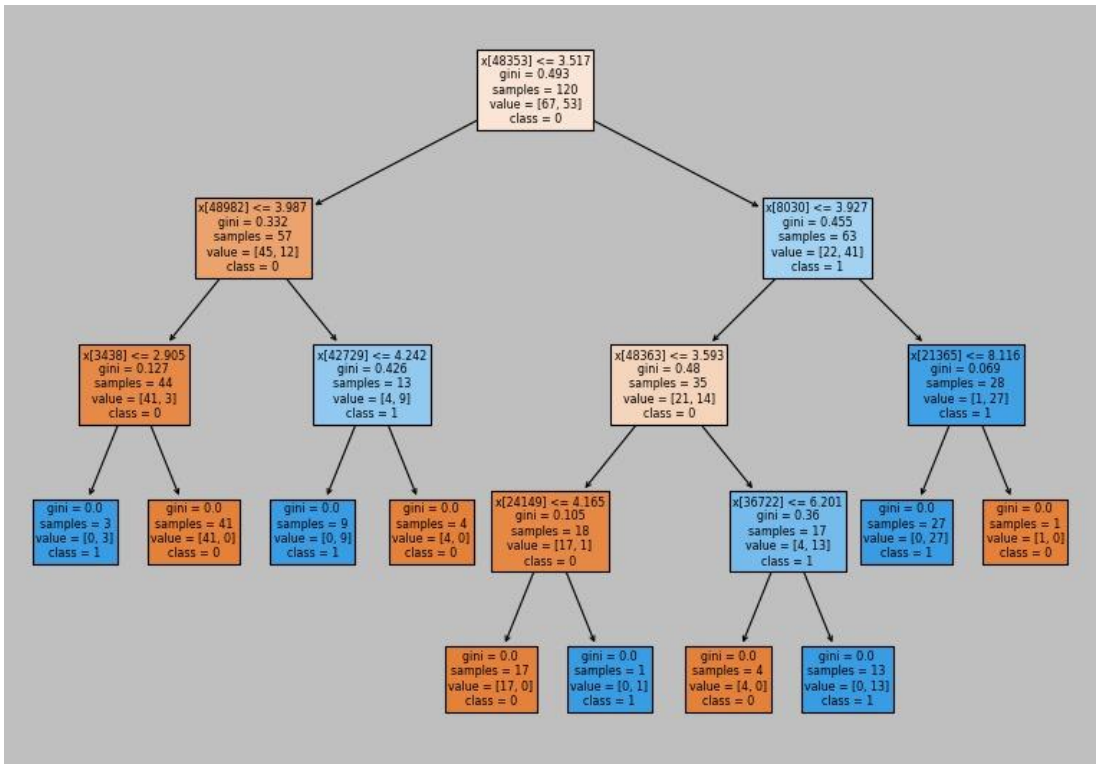

Fig. 2. Features class decision tree visualization

Random Forests

Random forests were also used as ensemble learning approach that aggregates predictions from multiple decision trees to improve the overall performance. Each decision tree in the forest independently predicts the class of a given data point. To classify a new data point 2, the algorithm collects predictions from all trees in the forest and then selects the mode (most common) class among these predictions as the final prediction. In the current study, the set of decision trees in the random forest was denoted as {*T1, T2, ..., T,*}, where *n* is the number of trees. The classification formula for random forests can be expressed as follows:

$$\hat{y} = (T_1(x), T_2(x), \ldots, T_n(x))$$

Where, $\hat{y}$ is the predicted class for the input data point 2, $T_1(x)$ represents the prediction of the *i*-th decision tree for data point 2, and *mode* (·) calculates the mode (most common) class among the predictions of all trees. By aggregating predictions from multiple trees, random forests can reduce overfitting and improve the overall robustness and accuracy of the classification model.

Support Vector Machine (SVM)

SVM works by finding the hyperplane in the feature space that maximizes the margin between classes. Given a set of training data with features; and corresponding binary labels y_i where y_i; ∈ {-1, +1}, SVM aims to find the optimal hyperplane w·x+b = 0 that separates the data points into two classes with the maximum margin. In the current study, the decision function for classifying a new data point z was given as:

$$f(x) = sign(w \cdot x + b)$$

Where, *2* is the input data point, *w* is the weight vector (normal to the hyperplane), *b* is the bias term, and sign (·) is the sign function which returns +1 if the argument is positive, -1 if it is negative, and 0 if it is zero. The hyperplane separates the feature space into two regions: one for each class.

To classify a new data point, the value of $w \cdot x + b$ was computed. When the result was positive, the data point was classified as the positive class (label +1); if negative, it was classified as the negative class (label -1). The margin is

DOI: 10.31695/IJASRE.2024.11.3

the distance between the hyperplane and the closest data point from each class. SVM was aimed at maximizing this margin, as it represents the separation between classes and improves generalization to unseen data.

Naive Bayes

Naive Bayes classifier calculates the posterior probability of each class given the input features using Bayes' theorem:

$$P(y|x_1, x_2, \dots x_n) = \frac{P(x_1, x_2, \dots x_{n|y}) \cdot P(y)}{P(x_1, x_2, \dots x_n)}$$

Where, $P(y|x_1, x_2, \dots x_n)$ is the posterior probability of class y given the input features $x_1, x_2, \dots x_n$; $P(x_1, x_2, \dots x_{n|y})$ is the likelihood of observing the input features given class *y*; $P(y)$ is the prior probability of class *y*, and $P(x_1, x_2, \dots x_n)$ is the evidence or marginal probability of observing the input features.

The predicted class (y) is the class with the highest posterior probability. Mathematically, it can be represented as:

$$\hat{y} = \underset{y}{\arg\max}\, P(y|x_1, x_2, \dots x_n)$$

Meaning, the Naive Bayes classifier assigns the input instance to the class with the highest posterior probability.

Gradient Boosting Machines (GBM)

The GBM builds a strong predictive model by sequentially adding decision trees to minimize a loss function. GBM consists of an algorithm that learns from the mistakes of previous models and iteratively improves the model's predictions. GBM initializes the model with a simple predictor, often the mean of the target variable for regression tasks or a constant value for classification tasks. During the model training, GBM sequentially adds decision trees to the ensemble, with each tree learning from the errors (residuals) of the previous trees. At each iteration, a decision tree is trained to predict the negative gradient of the loss function with respect to the current ensemble's predictions. For classification tasks, the negative gradient can be interpreted as the "pseudo-residuals," which represent the difference between the true class labels and the current ensemble's predictions.

For a binary classification problem, the prediction *y* made by the GBM model is typically obtained by summing the predictions from all the decision trees in the ensemble. This can be denoted using the formula:

$$\hat{y} = \sigma(f_1(x) + f_2(x) + \cdots + f_M(x))$$

Where, $\hat{y}$ is the predicted probability of the positive class (class 1), σ is the softmax function or logistic function used to convert raw scores into probabilities, $(f_1(x) + f_2(x) + \cdots + f_M(x)$ are the predictions from individual decision trees in the ensemble, and *M* is the total number of decision trees in the ensemble.

The objective of training GBM was to minimize a loss function $L(y, \hat{y})$, which measures the discrepancy between the true class labels y and the predicted probabilities y. The loss functions for the current classification tasks were:

Log Loss (Cross-Entropy)

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i log\hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)]$$

Hinge Loss (for SVM-based boosting):

$$L(y, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}\max(0, 1 - y_i \cdot \hat{y}_i)$$

DOI: 10.31695/IJASRE.2024.11.3

Exponential Loss:

$$L(y, \hat{y}) = \sum_{i=1}^{N} e^{-y_i \cdot \hat{y}_i}$$

Where, N is the total number of samples in the training data, y_i, is the true class label of the ith sample (0 or 1 for binary classification), $\hat{y}_i$ is the predicted probability of the positive class for the ith sample.

During each iteration of training, the negative gradient of the loss function with respect to the current ensemble's predictions was computed. For binary classification, the gradient was calculated as follows:

$$Gradient = -\frac{\partial L(y, \hat{y})}{\partial \hat{y}}$$

AdaBoost

AdaBoost, an ensemble learning technique, was used to combine the predictions from multiple weak learners to create a strong classifier. The algorithm assigns weights to each data point based on its classification accuracy, and it iteratively trains weak learners to focus more on misclassified data points in subsequent iterations. For the current study, AdaBoost was computed sequentially. Initially, each data point was assigned an equal weight $w_i \frac{1}{n}$, where n is the total number of data points in the training set. AdaBoost then sequentially trained a series of weak learners on the training data. A weak learner is a classifier that performs slightly better than random guessing, such as a decision stump. At each iteration t, a weak learner $h_t(x)$ was trained on the training data with weights ω_t. The weak learner minimizes the weighted error rate ϵ_t defined as:

$$\epsilon_t = \sum_{i=1}^{n} \omega_i^t 1[h_t(X_i \neq y_i]$$

Where: X_i; is the i-th data point, y_i; is the true label of the i-th data point, $1[\cdot]$; is the indicator function that returns 1 if the condition is true and 0 otherwise, ω_t; and are the weights assigned to the data points at iteration t. After training, the weak learner $h_t(x)$ produces predictions $\hat{y}_i^t$ for each data point X_i.

The weighted error of the weak learner $h_t(x)$ was calculated by updating the weighted error rate equation. AdaBoost updates the weights of the data points to give more emphasis to the misclassified points. The updated weights are calculated as follows:

$$w_i^{t+1} = w_i^t \cdot \exp(-\alpha_t y_i \hat{y}_i^t)$$

Where: α_t; is the weight associated with the weak learner $h_t(x)$, calculated as $\frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$, y_i is the true label of the i-th data point, and $\hat{y}_i^t$; is the prediction of the weak learner $h_t(x)$, for the i-th data point. The final prediction of AdaBoost is obtained by combining the predictions of all weak learners, weighted by their respective importance α_t.

Principle Component Analysis (PCA)

The PCA is a technique that helps to reduce dimensionality in data while preserving the most important information. The high-dimensional breast cancer data was analysed using the following steps. First, the data was centred by subtracting the mean of each feature from the data points. This ensured that the mean of each feature in the transformed data is zero. Next the covariance matrix was calculated to provide information about how each feature varies with every other feature in the dataset. Given a dataset with n observations and p features, the covariance matrix C is an p x p symmetric matrix calculated as:

DOI: 10.31695/IJASRE.2024.11.3

$$C = \frac{1}{n-1}\sum_{i=n}^{n}(X_i - \bar{x})(X_i - \bar{x})^T$$

Where: X_i; is the *i*-th observation (a vector of length *p*), $\bar{x}$; is the mean vector of all observations, $(X_i - \bar{x})$; represents the centred version of X_i, $(X_i - \bar{x})^T$; denotes the transpose of $(X_i - \bar{x})$.

PCA then conducts an eigenvalue decomposition which identifies the principal components (eigenvectors) of the covariance matrix and their corresponding variances (eigenvalues). The eigenvalue decomposition of the covariance matrix C yields:

$$C = VDV^T$$

Where: *V* is a matrix whose columns are the eigenvectors of *C*, *D* is a diagonal matrix containing the eigenvalues of C, and V^T denotes the transpose of V. The eigenvectors (principal components) corresponding to the largest eigenvalues capture the most variance in the data. Therefore, the principal components were ranked based on their corresponding eigenvalues, and the top *k* eigenvectors are selected to form the transformation matrix. The original data was projected onto the selected principal components to obtain the reduced-dimensional representation. This projection involved multiplying the centred data matrix by the transformation matrix consisting of the top k eigenvectors.

Performance Metrics

The models used for data analysis were assessed based on the performance metrics that included precision, recall, accuracy, area under the ROC curve (AUCROC), area under the precision-recall curve (AUC-PR), mean square error (MSE), root mean square error (RMSE), mean absolute percentage error (MAPE), and the coefficient of determination (R^2). Precision was used to assess the proportion of true positive predictions among all positive predictions. A high precision indicates that the model is making fewer false positive predictions, which is important when the cost of false positives is high. Precision was computed using the formula:

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

Where: *TP* represents the number of instances correctly classified as cell-line, and *FP* represents the number of instances incorrectly classified as cell-line. Sensitivity (recall) was used to assess the proportion of true positive predictions among all actual positives. A high recall indicates that the model is capturing most of the positive instances in the dataset, which is crucial when the cost of false negatives is high. It was computed using the formula:

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

Where; *FN* represents the number of instances of cell-line incorrectly classified as non-cell-line. The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is useful for comparing models with different precision-recall trade-offs. It was computed using the approach:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The model's accuracy measures the proportion of correctly classified instances among all instances. It was calculated as:

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

Where *TN* represents the number of instances correctly classified as non-cell-line. Support represented the number of instances in each class in the test set. It was used to provide context for the other metrics by showing the distribution of classes. The AUCROC measured the trade-off between true positive rate (sensitivity) and false positive rate (1 -

DOI: 10.31695/IJASRE.2024.11.3

specificity). A higher AUCROC value indicated better performance in distinguishing between classes. Also, the AUC-PR was used to determine the trade-off between precision and recall across different threshold values. AUC-PR a preferrable performance indicator for imbalanced datasets where precision and recall were more informative than accurate. The MAE was used to measure the average absolute difference between predicted and actual values. It was computed as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_1 - \hat{y}_i|$$

Where: y_1 represents the actual classification of the *i*-th instance in the dataset, and $\hat{y}_i$ represents the predicted classification. The MSE measures the average of the squared differences between predicted and actual values. It penalizes larger errors more heavily than MAE to avoid large errors and was computed as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} (y_1 - \hat{y}_i)^2$$

The RMSE is the square root of MSE, which brings the error metric back to the same scale as the original target variable. RMSE was used to interpret the average magnitude of errors in the same units as the target variable. Another metric, MAPE, was used to measure measures the average of the absolute percentage differences between predicted and actual values. MAPE was used to provide insights into the relative magnitude of prediction errors calculated as:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} |\frac{y_1 - \hat{y}_i}{y_1}| \times 100$$

Where: y_1 represents the actual classification of the *i*-th instance in the dataset, and $\hat{y}_i$ represents the predicted classification. The R^2 was used to indicate the proportion of the variance in the dependent variable that is predictable from the independent variables. R^2 ranges from 0 to 1, with higher values indicating better model fit computed as:

$$R^2 = \frac{SS_{res}}{SS_{tot}}$$

Where: SS_{res} represents the sum of squared residuals, and SS_{tot} represents the total sum of squares.

Class Imbalance Handling

The imbalance nature of gene expression data was addressed using the Synthetic Minority Over-sampling Technique (SMOTE) and the Random Under-sampling Technique (RUL). The dataset was transformed using the classifiers after min-max normalization. The dataset appears to be imbalanced as the cell_line is less than the non_cell_line.

DOI: 10.31695/IJASRE.2024.11.3

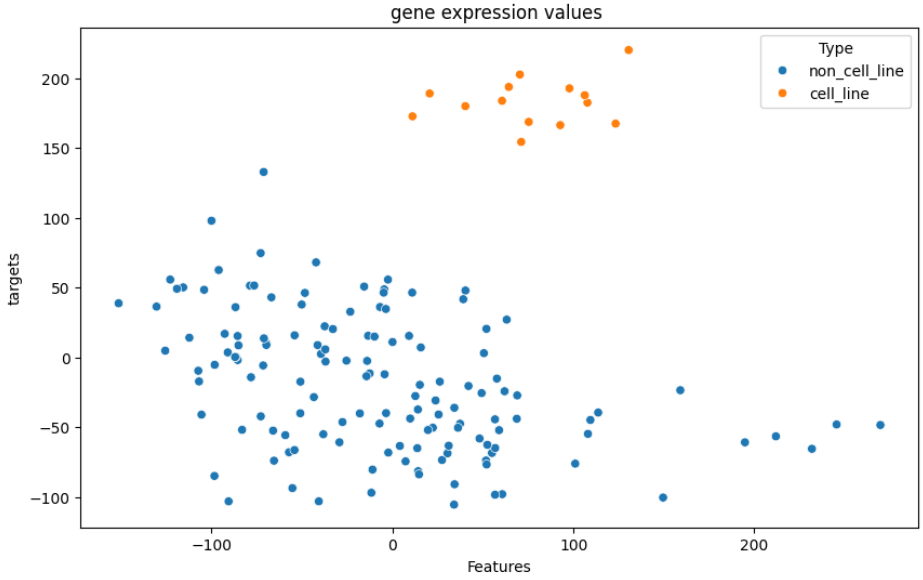

Fig. 3: Gene expression imbalance dataset

To handle the problem, the progressive SMOTE is added to the transformers as *X* being the input Data with *n* columns, and *T* as the set of transformers applied to different subsets of columns.

$$X_{transformed} = \begin{vmatrix} [X_1] \\ [X_2] \\ [.] \\ [.] \\ [X_n] \\ . \end{vmatrix}$$

X_i is the transformed subset of columns *i* using the corresponding transformer from set *T*. $X_{transformed}$ is the resulting transformed data. *T* consists of different pipelines, each applying a specific sequence of transformations to a subset of columns. Each pipeline included resampling, standard scaling, PCA for dimensionality reduction, and a classifier/regressor. To oversample the minority class by generating synthetic samples rather than replicating existing ones, SMOTE interpolated new samples along line segments connecting similar minority class instances.

$$X_{smote} = \text{SMOTE}\ (X_{minority})$$

Where: $X_{minority}$ represents the subset of data belonging to the minority class, cell_line, and *X*SMOTE represents the oversampled data after applying SMOTE. RUL, specifically NearMiss in this case, was used to undersample the majority class by selecting only the instances that are closest to the instances of the minority class.

$$X_{RUL} = \text{NearMiss}\ (X_{minority})$$

Where: $X_{majority}$ represented the subset of data belonging to the majority class, and *X*RUL represents the undersampled data after applying NearMiss. These resampling techniques are applied to different subsets of columns in the dataset, each followed by standard scaling, PCA for dimensionality reduction, and subsequent modelling steps as shown in Figure 4.

DOI: 10.31695/IJASRE.2024.11.3

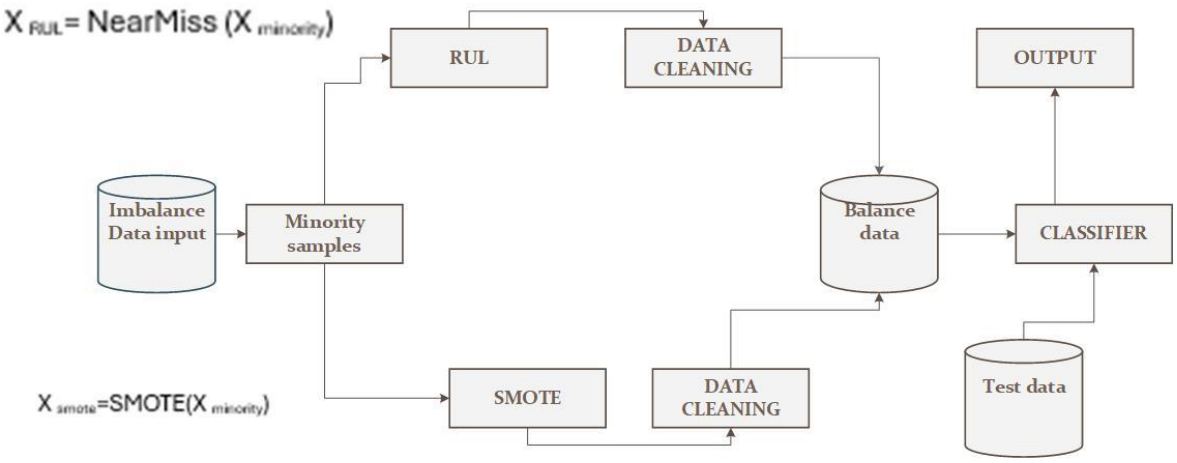

Fig. 4: Class Imbalance: SMOTE and RUL application to the classifier

Once the data was cleaned and preprocessed, SMOTE and RUL was applied to address class imbalance. The **ColumnTransformer** performed the following sequence of transformations by first applying SMOTE resampling to a subset of columns and then standard scaling followed by PCA and a kNN classifier. SMOTE resampling was then applied to another subset of columns and then standard scaling followed by PCA and a Dummy Classifier. Following this, NearMiss resampling was applied to another subset of columns and then standard scaling followed by PCA and a Decision Tree classifier. Further, SMOTE resampling as applied to another subset of columns and then standard scaling followed by PCA and a Random Forest classifier. SMOTE resampling was also applied to another subset of columns and then standard scaling followed by PCA and a SVM classifier, and this procedure repeated for Naïve Bayes and gradient-boosting classifier. A NearMiss resampling was then applied to another subset of columns and then standard scaling followed by PCA and an AdaBoost classifier, and finally, a standard scaling followed by PCA and a Linear Regression model were applied.

3. Result and Discussion

The study addressed the complex challenges associated with the statistical analysis of imbalanced high-dimensional data in breast cancer gene profiling.

Classification Problems in High-Dimensional Data with Class Imbalance

In the initial phase of the study, the researcher encountered the common challenge of classifying minority class elements accurately in high-dimensional data. Standard classification methods often struggled to perform satisfactorily due to the inherent class imbalance. As shown in Table 1, the performance of all classifiers was below 60% based on the precision, recall, F1-score, and accuracy metrics.

Table 1. Classifier performance without SMOTE

Model	Precision	Recall	F1-Score	Accuracy	Support	MAE	MSE	RMSE	MAPE (E+14)	R^2
kNN	0.3923	0.4402	0.3434	0.3871	31.0000	0.6129	0.6129	0.7829	4.3583	-1.5171
Linear Regression	0.4500	0.4679	0.3882	0.4194	31.0000	0.5806	0.5806	0.7620	4.3583	-1.3846
Dummy Classifier	0.2097	0.5000	0.2955	0.4194	31.0000	0.5806	0.5806	0.7620	0.5806	-1.3846
Decision Tree	0.5022	0.5021	0.4833	0.4839	31.0000	0.5161	0.5161	0.7184	7.2639	-1.1197
Random Forest	0.4940	0.4957	0.4303	0.4516	31.0000	0.5484	0.5484	0.7405	4.3583	-1.2521

DOI: 10.31695/IJASRE.2024.11.3

SVM	0.5299	0.5235	0.4701	0.4839	31.0000	0.5161	0.5161	0.7184	4.3583	-1.1197
Naive Bayes	0.4823	0.4850	0.4423	0.4516	31.0000	0.5484	0.5484	0.7405	5.8111	-1.2521
Gradient Boosting	0.4940	0.4957	0.4303	0.4516	31.0000	0.5484	0.5484	0.7405	4.3583	-1.2521
AdaBoost	0.5188	0.5192	0.5141	0.5161	31.0000	0.4839	0.4839	0.6956	8.7166	-0.9872
PCA	0.3923	0.4402	0.3434	0.3871	31.0000	0.6129	0.6129	0.7829	4.3583	-1.5171

As illustrated in Figure 5, synthetic data generated by SMOTE to handle the imbalanced data is more uniform and balanced than the initial data shown in Figure 3. SMOTE has formed the core for improvement of class imbalanced data to render classifiers more effective even in a previous study by Wang *et al*. [30]. However, by employing resampling techniques such as SMOTE and RUL (NearMiss), significant improvements in classification performance were observed. These techniques helped generate synthetic samples of the minority class or under-sampling of the majority class, thereby balancing the class distribution and enhancing the model's ability to generalize well to minority class instances.

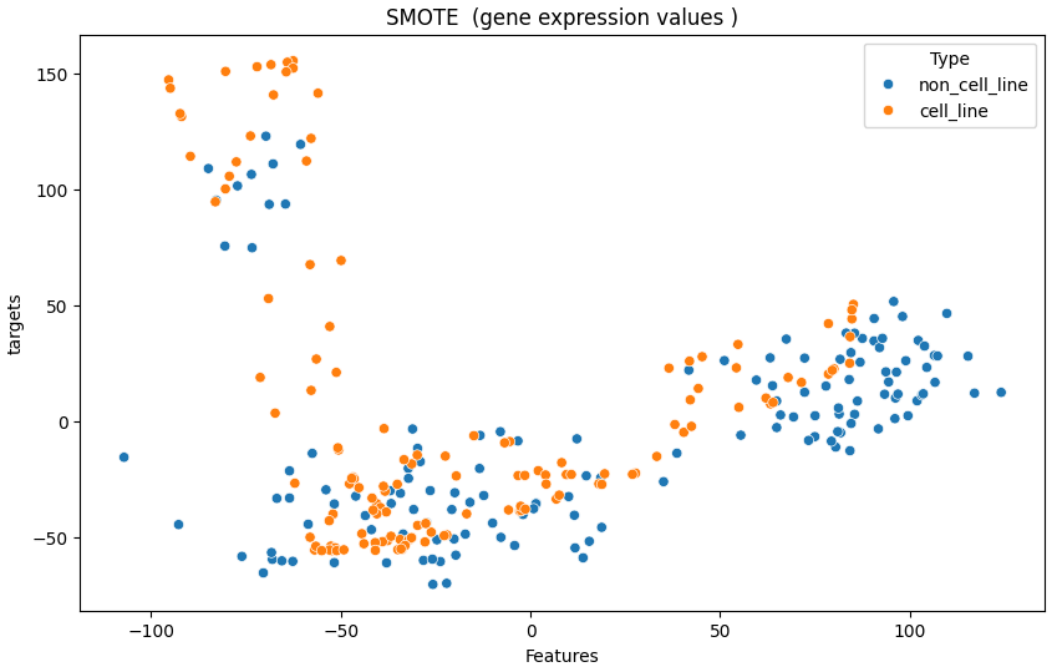

Fig. 5: SMOTE-RUL gene expression features being balanced

Table 2 illustrates the performance of classifiers after applying SMOTE on the class-imbalanced data. The tables depicting the metrics for balanced data with SMOTE and RUL clearly illustrate the substantial improvements in precision, recall, F1-score, and overall accuracy across various classification models. For instance, models trained on SMOTE and RUL balanced data consistently achieved high precision, recall, and F1-score of 1, indicating robust performance in classifying minority and majority class instances.

Classification models trained with SMOTE and RUL achieved perfect precision, recall, F1-score, and accuracy, indicating flawless classification across minority and majority classes. SVM, Decision Tree, and AdaBoost classifiers performed consistently well across both resampling methods. The results obtained in this study support previous work by Pes which favours hybrid models for handling class imbalanced, high-dimensional data [32]. On the other hand, the regression model, represented by Linear Regression, exhibited high accuracy (R^2 = 0.9706) but relatively high mean squared error (MSE = 0.0055) and root mean squared error (RMSE = 0.0742), suggesting a slight deviation from the true values. Overall, these findings underscore the effectiveness of resampling techniques in improving classification

DOI: 10.31695/IJASRE.2024.11.3

performance, while Linear Regression demonstrates strong predictive capability despite minor discrepancies in error metrics.

Table 2. The performance of classifiers with SMOTE

Model	Precision	Recall	F1-Score	Accuracy	Support	AUCROC	AUC-PR	MAE	MSE	RMSE	MAPE	R^2
SMOTE - kNN	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
SMOTE - Dummy Classifier	0.125	0.5	0.2	0.25	16	0.5	0.875	-1	-1	-1	-1	-1
SMOTE - Decision Tree	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
SMOTE - Random Forest	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
SMOTE - SVM	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
SMOTE - Naive Bayes	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
SMOTE - Gradient Boosting	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
SMOTE - AdaBoost	1	1	1	1	16	1	1	-1	-1	-1	-1	-1

Additionally, the Table 2 and Table 3 presenting metrics for imbalanced data without SMOTE and RUL offer valuable insights into the performance of classification models under challenging conditions. The tables show the performance metrics of various classification and regression models applied to the dataset without resampling techniques. Decision Tree and SVM exhibited the highest precision, recall, and F1 scores among the classifiers, indicating their effectiveness in accurately classifying instances. Notably, the Dummy Classifier, which predicts the most frequent class, shows low precision and F1-score, reflecting its inability to discriminate between classes.

DOI: 10.31695/IJASRE.2024.11.3

Table 3. The performance of classifiers with RUL

Model	Precision	Recall	F1-Score	Accuracy	Support	AUCROC	AUC-PR	MAE	MSE	RMSE	MAPE	R^2
RUL - kNN	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
RUL - Dummy Classifier	0.125	0.5	0.2	0.25	16	0.5	0.875	-1	-1	-1	-1	-1
RUL - Decision Tree	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
RUL - Random Forest	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
RUL - SVM	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
RUL - Naive Bayes	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
RUL - Gradient Boosting	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
RUL - AdaBoost	1	1	1	1	16	1	1	-1	-1	-1	-1	-1
Linear Regression	-1	-1	-1	-1	-1	-1	-1	0.062	0.006	0.074	1.1E+14	0.971

On the other hand, the regression model, represented by Linear Regression, demonstrates moderate predictive performance, with an accuracy of 41.94% and an R^2 value of -1.3846. These results suggest that while some classifiers perform reasonably well without resampling, the predictive capability of the regression model could be further improved. The findings are in line with a study by Bao *et al*. who demonstrated that classifiers such as structured k-NN as well as SMOTE sampling model show improved performance when used together to boost each other [31].

DOI: 10.31695/IJASRE.2024.11.3

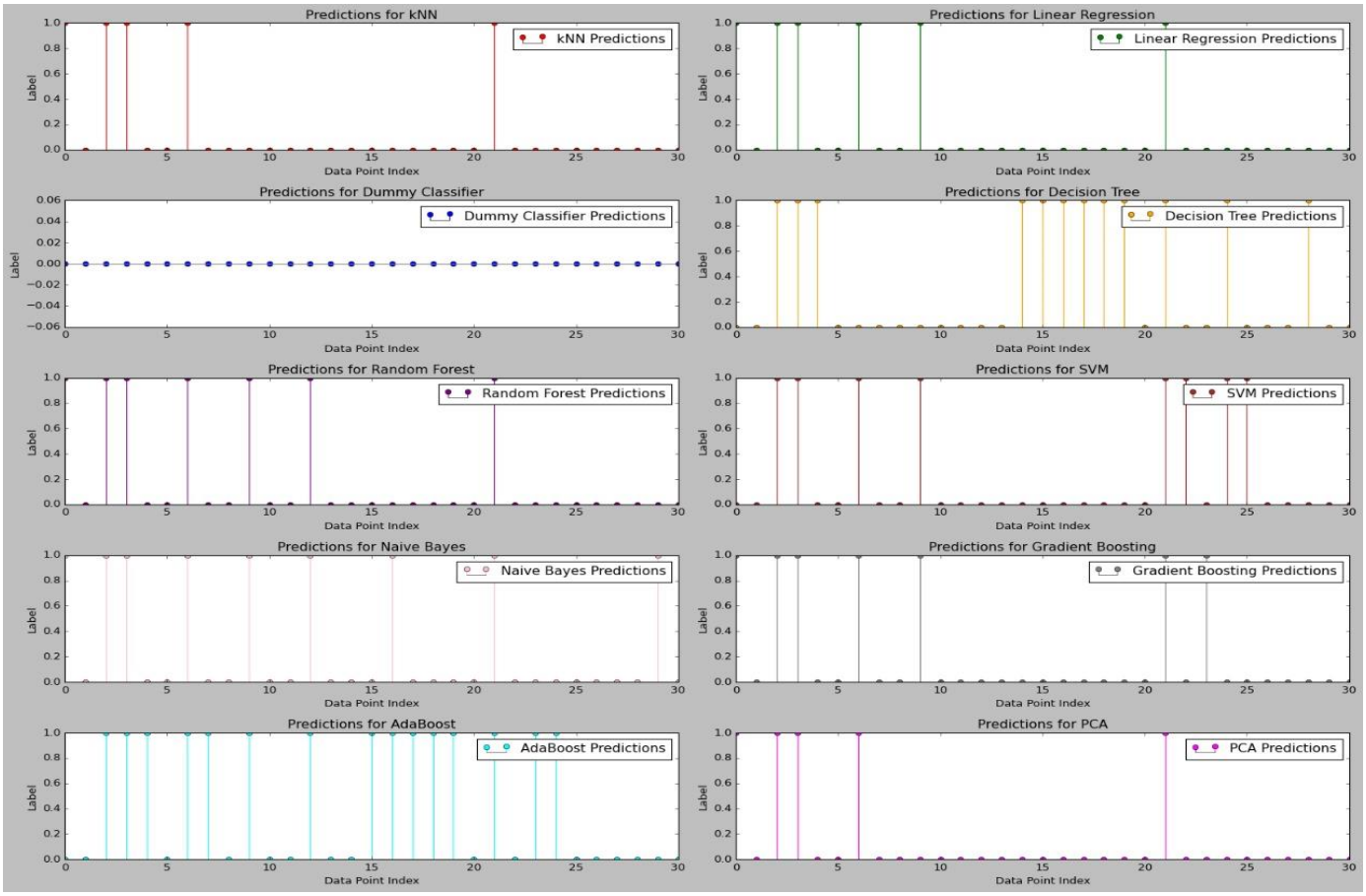

Fig. 6. Predictions for individual model under imbalance class

4.CONCLUSION

By integrating resampling techniques, advanced classifiers, and regression models, the researchers proposed a robust framework that addresses the complexities of imbalanced high-dimensional data prevalent in oncology research. The framework's design aimed to optimize the classification and prediction accuracy while ensuring the reliability and generalizability of the results. While the current approach to address imbalance and high-dimension in the gene expression data showed significant improvement, the researcher recommends the following for future studies. For starters, considering the superior performance of Decision Trees and SVM classifiers regarding precision, recall, and F1-scores, the researcher recommends a deeper exploration and optimization of these models. Also, since the Dummy Classifier showed poor discrimination between classes, its usage should be avoided in favour of more complex and thorough classifiers. It should only be used to confirm the performance of classifiers and algorithms of interest. Further, a comprehensive analysis of feature selection methods and model tuning should be sought to enhance the predictive performance across classification and regression tasks.

REFERENCES

[1] Antabe, R., Kansanga, M., Sano, Y., Kyeremeh, E., & Galaa, Y. “Utilization of breast cancer screening in Kenya: what are the determinants?” BMC health services research, Vol. *20* no.1, 2020, pp.1-9.

[2] Bao, T., & Davidson, N. E. (2008). Gene expression profiling of breast cancer. *Advances in surgery*, *42*, 249-260.

[3] Tokumaru, Y., Joyce, D., & Takabe, K. (2020). Current status and limitations of immunotherapy for breast cancer. *Surgery*, *167*(3), 628–630.

[4] Cejalvo, J. M., Martínez de Dueñas, E., Galván, P., García-Recio, S., Burgués Gasión, O., Paré, L., Antolín, S., Martinello, R., Blancas, I., Adamo, B., & Prat, A. (2017). Intrinsic subtypes and gene expression profiles in primary and metastatic breast cancer. *Cancer research*, *77*(9), 2213-2221.

DOI: 10.31695/IJASRE.2024.11.3

[5] Petinrin, O. O., Saeed, F., Salim, N., Toseef, M., Liu, Z., & Muyide, I. O. (2023). Dimension Reduction and Classifier-Based Feature Selection for Oversampled Gene Expression Data and Cancer Classification. *Processes*, *11*(7), 1940.

[6] Moon, K. R., van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D. B., Chen, W. S., ... & Krishnaswamy, S. (2019). Visualizing structure and transitions in high-dimensional biological data. *Nature biotechnology*, *37*(12), 1482-1492.

[7] Ram, M., Najafi, A., & Shakeri, M. T. (2017). Classification and biomarker genes selection for cancer gene expression data using random forest. *Iranian journal of pathology*, *12*(4), 339.

[8] Kwa, M., Makris, A., & Esteva, F. J. (2017). Clinical utility of gene-expression signatures in early-stage breast cancer. *Nature reviews Clinical oncology*, *14*(10), 595-610.

[9] Russnes, H. G., Lingjærde, O. C., Børresen-Dale, A. L., & Caldas, C. (2017). Breast cancer molecular stratification: from intrinsic subtypes to integrative clusters. *The American journal of pathology*, *187*(10), 2152-2162.

[10] Hubalek, M., Czech, T., & Müller, H. (2017). Biological subtypes of triple-negative breast cancer. *Breast Care*, *12*(1), 8-14.

[11] Hwang, K. T., Kim, J., Jung, J., Chang, J. H., Chai, Y. J., Oh, S. W., Kim, Y. A., Park, S. B., & Hwang, K. R. (2019). Impact of breast cancer subtypes on prognosis of women with operable invasive breast cancer: a population-based study using SEER database. *Clinical Cancer Research*, *25*(6), 1970-1979.

[12] Howlader, N., Cronin, K. A., Kurian, A. W., & Andridge, R. (2018). Differences in breast cancer survival by molecular subtypes in the United States. *Cancer Epidemiology, Biomarkers & Prevention*, *27*(6), 619-626.

[13] Kumar, Y., Gupta, S., Singla, R., & Hu, Y. C. (2022). A systematic review of artificial intelligence techniques in cancer prediction and diagnosis. *Archives of Computational Methods in Engineering*, *29*(4), 2043-2070.

[14] Xiao, Y., Wu, J., Lin, Z., & Zhao, X. (2018). A deep learning-based multi-model ensemble method for cancer prediction. *Computer methods and programs in biomedicine*, *153*, 1-9.

[15] Islam, M. M., Haque, M. R., Iqbal, H., Hasan, M. M., Hasan, M., & Kabir, M. N. (2020). Breast cancer prediction: a comparative study using machine learning techniques. *SN Computer Science*, *1*, 1-14.

[16] Zhu, W., Xie, L., Han, J., & Guo, X. (2020). The application of deep learning in cancer prognosis prediction. *Cancers*, *12*(3), 603.

[17] Khourdifi, Y., & Bahaj, M. (2018). Applying best machine learning algorithms for breast cancer prediction and classification. In *2018 International conference on electronics, control, optimization and computer science (ICECOCS)* (pp. 1-5). IEEE.

[18] Huang, M. W., Chen, C. W., Lin, W. C., Ke, S. W., & Tsai, C. F. (2017). SVM and SVM ensembles in breast cancer prediction. *PloS one*, *12*(1), e0161501.

[19] Sengar, P. P., Gaikwad, M. J., & Nagdive, A. S. (2020, August). Comparative study of machine learning algorithms for breast cancer prediction. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 796-801). IEEE.

[20] Tirumala, S. S., & Narayanan, A. (2019). Classification and diagnostic prediction of prostate cancer using gene expression and artificial neural networks. *Neural Computing and Applications*, *31*, 7539-7548.

[21] Ferroni, P., Zanzotto, F. M., Riondino, S., Scarpato, N., Guadagni, F., & Roselli, M. (2019). Breast cancer prognosis using a machine learning approach. *Cancers*, *11*(3), 328.

DOI: 10.31695/IJASRE.2024.11.3

[22] Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. *Information Sciences*, *513*, 429-441.

[23] Mathew, J., Pang, C. K., Luo, M., & Leong, W. H. (2017). Classification of imbalanced data by oversampling in kernel space of support vector machines. *IEEE transactions on neural networks and learning systems*, *29*(9), 4065-4076.

[24] Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., & Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics & Data Analysis*, *143*, 106839.

[25] 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), 1-30.

[26] Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, 350-361.

[27] L'heureux, A., Grolinger, K., Elyamany, H. F., & Capretz, M. A. (2017). Machine learning with big data: Challenges and approaches. *Ieee Access*, *5*, 7776-7797.

[28] Munkácsy, G., Santarpia, L., & Győrffy, B. (2022). Gene expression profiling in early breast cancer—patient stratification based on molecular and tumor microenvironment features. *Biomedicines*, *10*(2), 248.

[29] Shahid, M. F., Aafaq, N., Mahmud, S. K., & Kazmi, S. M. K. A. (2023). Influence of various ML-based binary classifiers on the performance on handwritten digit recognition. In *2023 10th International Conference on Future Internet of Things and Cloud (FiCloud)* (pp. 283-291). IEEE.

[30] Wang, L., Han, M., Li, X., Zhang, N., & Cheng, H. (2021). Review of classification methods on unbalanced data sets. *IEEE Access*, *9*, 64606-64628.

[31] Bao, F., Wu, Y., Li, Z., Li, Y., Liu, L., & Chen, G. (2020). "Effect improved for high-dimensional and unbalanced data anomaly detection model based on KNN-SMOTE-LSTM." Complexity, *2020*, 1-17.

[32] Pes, B. (2021). Learning from high-dimensional and class-imbalanced datasets using random forests. *Information*, *12*(8), 286.

DOI: 10.31695/IJASRE.2024.11.3