Overcoming the Curse of Dimensionality in Microarray Data Via Generative Data Augmentation and L1-Regularized Selection

Fawzia A. E. Mansur [1]
Department of Electrical and Computer Engineering
Libyan Academy for Postgraduate Studies
Tripoli-Libya
fawzia.almsmary@academy.edu.ly
Issmail M. Ellabib [2]
Computer Engineering Department
University of Tripoli
Tripoli-Libya
i.ellabib@uot.edu.ly

Abstract- Feature selection research is crucial to overcome the dimensionality curse. Despite numerous attempts to select features for high-dimensional data, many existing techniques suffer from limited computational efficiency and fail to capture complex and deep correlations among features, particularly in highly complex and low-sample-size. This article presents an innovative framework that uses conditional competitive generative networks (CGANs) to mitigate the limitations of differential evolution (DE) and the low-dimensional, small-scale dataset problem (HDLSS) by increasing dataset size and improving the stability of the selection process, thereby enhancing the model's generalizability. Experimental results show that the model achieved a classification accuracy of 89.59%, surpassing the best reference method of 86.10%. This highlights the framework's effectiveness in balancing accuracy and computational efficiency, as well as its applicability to diverse high-dimensional data scenarios.

Keywords---dimensionality, Feature Selection (FS), Data Augmentation, Deep Learning

I. INTRODUCTION

In many real-world applications, particularly in genomics and biomedical informatics, datasets are characterized by an extremely large number of features relative to the quantity of available samples — a setting frequently referred to as High-Dimensional Low Sample Size (HDLSS). The selection of features significantly affects the overall effectiveness of classification models, as irrelevant and redundant variables inflate the search space, increase computational cost, and degrade predictive accuracy [1]. Dimension reduction through feature selection has therefore emerged as an essential preprocessing step, enabling the elimination of uninformative features, the acceleration of the learning process, and the improvement of model interpretability. However, the complexity of feature selection grows dramatically with dimensionality, rendering exhaustive search strategies impractical in high-dimensional settings. To overcome these difficulties, a number of feature selection strategies have been put forth. Filter techniques use statistical criteria to assess features without the need for a classifier., offering computational efficiency but often failing to capture complex feature interactions [2]-[3]. Wrapper methods, while capable of identifying highly predictive subsets, incur prohibitive computational costs in high-dimensional spaces. Integrated approaches, such as LASSO, offer a promising compromise by incorporating feature selection into the model training process. Ensemble-based approaches, which combine multiple selection criteria to improve stability, have also been explored [4]-[5] More recently, evolutionary optimization strategies, such as differential evolution (DE), have been applied to feature selection; however, they suffer from statistical instability and a high computational load when applied to HDLSS datasets, limiting their practical utility in biomedical fields. Deep learning-based approaches have shown promising potential in addressing class imbalances and cluttered data [6], while comparative approaches, such as Contrast FS [7], have demonstrated competitive accuracy using filter-based discriminant classification. Despite these advancements, most current approaches address data scarcity or feature redundancy separately, without optimizing both

objectives together within a unified framework. The introduction of deep generative models, particularly competitive generative networks (GANs), has opened new avenues for addressing data scarcity in high-dimensional environments. Competitive generative networks are characterized by their ability to generate high-quality synthetic samples that maintain a conditional classification structure and understand the underlying distribution of real data. (CGANs) prolong this capability by conditioning the generative process on class labels, enabling the controlled synthesis of class-specific samples — a property especially valuable in multi-class biomedical classification tasks where minority classes are severely underrepresented. Despite the growing interest in generative augmentation and deep learning-based feature selection, few studies have systematically integrated these two complementary strategies into a cohesive, end-to-end framework specifically designed for HDLSS microarray data. To close this gap, this paper proposes a novel framework that combines CGAN-based data augmentation with L1-regularized embedded feature selection for robust classification of HDLSS microarray datasets. Specifically, the framework employs CGAN-based augmentation to mitigate sample sparsity and stabilize the feature selection process, followed by LASSO-based selection to identify a compact and discriminative feature subset achieving dimensionality reduction exceeding 99%. A Support Vector Classifier (SVC) is then trained using the chosen features. (SVC), evaluated across thirteen benchmark microarray datasets. The proposed approach is compared with the latest methods, demonstrating superior performance in terms of classification accuracy, AUC, and computational efficiency on datasets characterized by varying degrees of class and dimensional imbalance. Section 2 examines relevant research on feature selection in high-dimensional settings. Section 3 describes the proposed methodology, encompassing data preprocessing, CGAN-based augmentation, and L1-regularized feature selection. The experimental setup and findings are presented in Section 4, and Section 5 offers a comparative analysis. The paper's conclusion and future study directions are outlined in Section 6.

II. LITERATURE REVIEW

Nalić et al. [8] employed multiple feature selection algorithms and integrated their outcomes using different voting strategies combined with classification models based on logistic regression. Their evaluation relied on ROC and accuracy metrics, achieving high performance; however, the approach required substantial memory consumption, which limited its scalability.

Wu. [9] proposed HIBoost, an ensemble learning framework designed to address high-dimensional and imbalanced data problems by incorporating hubness-aware and imbalance-sensitive learning strategies within a boosting structure (AdaBoost). The method was evaluated on sixteen datasets from OpenML and UCI repositories. Despite the effectiveness of this proposed approach, its generalizability to other dataset techniques, such as gradient enhancement, random forests, and clustering, has not yet been explored.

Chen et al. [10] presented a unified feature selection framework that integrates filtering, encapsulation, and embedding techniques for mixed medical datasets, including numeric and categorical data. The approach significantly reduced feature dimensionality while preserving high classification accuracy.

Kshatri. [11] created a feature selection technique based on an ensemble employing bagging and boosting with bootstrap sampling. Feature rankings were aggregated using multiple methods for rank aggregation to identify the most crucial characteristics. However, further improvement is still needed to optimize feature subset selection and enhance classification accuracy.

Hamim. [12] proposed a dimensionality reduction framework Ant Colony Optimization for microarray data and Pearson correlation. The reduced feature set was evaluated using C5.0, Artificial Neural Networks, Logistic Regression, and Support Vector Machines for cancer classification. Although the framework demonstrated strong adaptability across datasets, further exploration of advanced data mining techniques is still required.

Kim et al. [13] introduced a filter-centric hybrid feature selection framework (ELFS) for high-dimensional unbalanced data based on ensemble learning. Experimental results showed improved performance; however, the method still suffers from high computational time when applied to extremely high-dimensional datasets.

Jaw and Wang. [14] suggested using a hybrid feature selection method combined utilizing an ensemble classifier (KODE) to eliminate irrelevant and redundant features in intrusion detection tasks. Although the method improved classification

performance, it still presents limitations that require further refinement.

Mera-Gaona et al. [15] developed a conceptual ensemble feature selection framework implemented in Python to mitigate noise and irrelevant features. The framework was evaluated using decision tree and logistic regression classifiers across multiple datasets and demonstrated stable performance. However, careful consideration is required to avoid the removal of informative features during selection.

Bashir et al. [16] applied multiple methods for feature selection such as, including filters, wrappers, and embedded methods, on medical datasets such as the Cleveland Heart Disease dataset and microarray data. SVM was used to assess performance using criteria like accuracy, F-measure, sensitivity, and specificity. Future improvements may include the use of evolutionary algorithms such as Particle Swarm Optimization.

Kumar et al. [17] suggested MISFS, a hybrid feature selection technique that uses cross-validation to combine mutual information and sequential forward selection. This method demonstrated superior performance compared to existing approaches; however, striking a balance between accuracy and ease of interpretation remains a challenge.

Zhong et al. [2] presented a feature selection method based on two-layer cross-validation, which was tested on real-world datasets. The method successfully identified informative feature subsets; however, its computational complexity remains high, particularly when combining multiple filter-based techniques.

Liu.[18] applied a hybrid approach combining the Extremely Randomized Trees (ERT), Variance Filter, and Whale Optimization Algorithm on gene expression datasets. While the method improved classification performance, gene selection remains challenging due to high dimensionality and small sample sizes.

Rufino et al. [19] addressed the problem of dataset imbalance using the repeated feature exclusion (RFE) technique with tree-based classifiers such as Random Forest, Light GBM, and XGBoost. The models were evaluated using various criteria, including sensitivity, specificity, area under the curve (AUC), and F1 score. However, further optimization for dataset-specific characteristics is recommended.

Mandal et al. [4] proposed a two-stage feature selection approach. integrating several filter techniques with Differential Evolution optimization. Although effective, the approach is computationally expensive and may have limited applicability across different datasets and algorithms. In addition, HDLSS datasets have many features but few samples, complicating data analysis and increasing computational costs, which can lead to overfitting. Traditional feature selection methods often fail to balance feature reduction with classification accuracy. The proposed two steps approach integrates five filter methods for ensemble feature selection and uses a Differential Evolution (DE) algorithm for optimization. However, it is based on specific feature selection and classification algorithms. So, the specific problem is that Differential Evolution (DE) suffers from both computational inefficiency and statistical instability and The Curse of dimensionality in high dimensions-low sample size.

III. METHODOLOGY

This section will outline the suggested approach's methodology. Figure 1 illustrates the process that was employed for data augmentation utilizing GANs for stable and selection of scalable features in order to lessen the drawbacks of Differential Evolution (DE) and the curse of dimensionality in HDLSS datasets. In this research, thirteen well-known and extensively used microarray datasets are sourced from two main repositories: The first set is from the website: CSSE. The remaining datasets are obtained from Biolab. Therefore, these datasets allow us to compare the proposed methods with other works. The specific details on the thirteen datasets are listed in Table 1 It is evident that these datasets include a lot of features but few samples, which makes classification tasks difficult and more complicated. High-Dimensional Low Sample Size (HDLSS) is the term used to describe datasets that have an exceptionally high number of characteristics compared to the number of accessible samples in many real-world applications, especially in genomics and biomedical informatics. This variability presents significant challenges, such as model overfitting, increased variability, and reduced generalizability. Feature selection has become a crucial preprocessing step to mitigate these challenges by retaining only the most useful features, thereby reducing dimensionality, accelerating learning, and improving interpretability.

TABLE 1: SYNOPISIS OF THE DATASETS

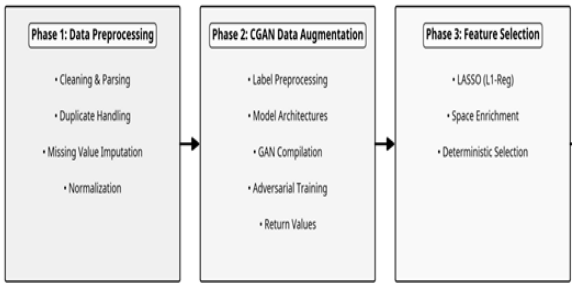

Fig.1. Deep Learning-based Feature Selection on GAN-augmented

A. Data Preprocessing

Any machine learning process begins with efficient data preparation, especially when dealing with high-dimensional microarray datasets, where the raw data is often diverse, noisy, and incomplete. The preprocessing phase in the proposed framework consists of five sequential steps designed to ensure data quality, consistency, and compatibility with subsequent deep learning components.

1) Cleaning and parsing

The datasets used in this study are stored in ARFF (Attribute Relationship File Format), a standard format commonly used in bioinformatics repositories Each file is parsed systematically to extract feature attributes and class labels. During parsing, categorical attributes encoded in binary are decoded into readable UTF-8 strings to ensure consistency across datasets. This step is essential to avoid any encoding variations that could introduce artificial noise into the feature space and negatively impact the integrity of subsequent analysis.

2) Duplicate Handling

A common problem in large-scale microarray datasets is the presence of duplicate or ambiguous feature names, which may arise from probe duplication or the merging of data from multiple experimental sources. To address this issue, a dedicated deduplication function is implemented, which systematically detects duplicate feature names and resolves conflicts by adding unique numeric suffixes. This ensures that each feature is uniquely identifiable throughout the pipeline, preventing indexing errors during feature selection and model training.

3) Missing Value Imputation

Dataset	#Genes	#Instances	#Classes
Colon Tumor	2000	60	2
Central Nervous System	7129	60	2
ALL-AML	7129	72	2
Breast Cancer	24481	97	2
Lung Cancer	12533	181	2
Ovarian Cancer	15154	253	2
ALL-AML-3	7129	72	3
ALL-AML-4	7129	72	4
Lymphoma	4026	62	3
MLL	12582	72	3
SRBCT	2308	83	4

Since most machine learning models aren't equipped to handle incomplete information, addressing missing values is a crucial prerequisite for training. These gaps in data typically stem from practical hurdles like equipment malfunctions, manual entry errors, or inconsistencies during data integration. Incomplete data is addressed using a Simple Imputer with a median strategy to maintain robustness against genomic outliers. This method helps us fill in gaps in the data intelligently instead of deleting the data entirely.

4) Normalization

Features are linearly scaled to a $[0, 1] $ interval using Min-Max scaling to ensure training stability in neural networks. When features in a dataset operate on different scales, machine learning models often struggle to treat them objectively. A feature with a high numerical range—like annual income—can "drown out" a feature with a small range—like a decimal-based health metric—simply because its numbers are larger, not because it is more important. Data standardization makes all data values comparable and balanced, allowing the model to treat all features fairly during the learning process.

5) Stratified Sampling

Resampling is performed to preserve original class proportions within the processed dataset. When a dataset is skewed—meaning one category (the majority class) significantly outnumbers another (the minority class)—a simple random split can be risky.

If the minority class is rare enough, a random shuffle might result in a training set that lacks enough examples to learn from, or a test set that doesn't contain the minority class at all, making it impossible to evaluate performance accurately. Stratified sampling means we split the data while keeping the same class distribution in every part, so the model learns and is tested on a balanced and realistic representation of the data.

6) Statistical Feature Scoring

To assess the importance of each feature relative to the target classes, three complementary statistical filter methods were utilized: the Chi-Square (χ^2) test, Mutual Information (MI), and the ANOVA F-test. The χ^2 test measures the degree of association between individual features and class labels, whereas MI evaluates the amount of information shared between a feature and the target variable. The ANOVA F-test, on the other hand, estimates a feature's discriminative capability by examining variations in feature means across different classes.

Since the three measures produce scores on different scales, all scores were normalized to the interval [0, 1] prior to aggregation. A single feature relevance score was then derived by taking the median of the normalized χ^2, MI, and F-test values for each feature:[Score_i = \operatorname{Median}(\chi^2_i, MI_i, F_i)].The median was chosen as the aggregation operator because of its resistance to outliers and extreme values, yielding a more robust and reliable estimate of feature importance. The resulting combined scores were subsequently employed to rank features and support the subsequent feature selection process.

B. CGAN-Based Data Augmentation

Enables the generation of high-quality synthetic data across various domains. This capability is particularly valuable for addressing challenges associated with small sample sizes and high-dimensional gene expression data. The primary objective of data augmentation using CGANs is to expand the available dataset without the need for additional real-world data collection, balance imbalanced class distributions, and enhance the robustness and generalizability of predictive models. Consequently, this leads to improved performance of both machine learning and deep learning algorithms. CGANs consist of two neural networks—a generator and a discriminator—that are trained simultaneously through a competitive process. The training procedure is formulated as a minimax game between the Generator (G) and the Discriminator (D), conditioned on class labels (y). As shown in Equation (1), the objective function reflects this adversarial interaction, where the generator aims to produce realistic samples conditioned on (y), while the discriminator seeks to distinguish between real and generated data.

$$\min_G \max_D V(D, G) = \mathrm{E}_{x,y\sim p_{data}(x,y)}[\log D(x,y)] + \mathrm{E}_{z\sim p_z(z),y\sim p_{data}(y)}[\log(1 - D(G(z,y),y))] \quad (1)$$

The goal function (minimax game) of a Conditional Generative Adversarial Network (CGAN), in which the Generator (G) and Discriminator (D) are trained in opposition, is represented by equation (1).

Where:

- P_{data} (x, y) is the distribution of actual labels and data.
- P_z (z) is the input noise's distribution (e.g., Gaussian).
- D (x, y) is the discriminator's calculated likelihood that x is an actual sample from class y.
- G (z, y) is the output of the generator for class y given z noise.

a). Discriminator Architecture (D)

The Discriminator in the Conditional Generative Adversarial Network (CGAN) functions as a conditional binary classifier designed to distinguish real gene expression samples from generated ones while simultaneously verifying their consistency with the associated class labels. It receives as input the concatenation of the gene expression vector x and the corresponding condition label y, enabling the model to determine not only whether a sample is real or synthetic, but also whether it is biologically consistent with its assigned class, such as a specific cancer type or subtype. The combined input is processed through multiple fully connected dense layers with nonlinear activation functions, particularly LeakyReLU, which allows small gradients for negative inputs and prevents neuron inactivation. The first hidden layer, consisting of 256 neurons, captures complex gene–gene interaction patterns and reduces noise within the high-dimensional feature space, while the second hidden layer with 128 neurons refines these representations into more compact and discriminative features. To

improve generalization and reduce overfitting in the high-dimensional, low-sample-size setting, regularization techniques such as dropout are incorporated during training. The output layer consists of a single neuron with a sigmoid activation function, producing a probability value D(x , y)∈[0,1] that represents the likelihood that the input pair originates from real biological data rather than being synthetically generated. Through adversarial training, the discriminator learns to maximize its ability to correctly classify real samples with target value 1 and generated samples with target value 0 by minimizing the binary cross-entropy loss function. As illustrated in Equation (2), the discriminator minimizes the negative log-likelihood associated with the correct classification of both real and synthetic samples, thereby ensuring that the generated data preserve meaningful biological characteristics and realistic statistical dependencies among genes.

$$L_D = -\frac{1}{m}\sum_{i=1}^{m}[Log\ D(x^{(i)}, y^{(i)}) + Log(1 - D(G(z^{(i)}, y^{(i)}), y^{(i)}))] \quad (2)$$

Where m is the batch size.

b). Generator Architecture (G)

The Generator (G) is implemented as a Multi-Layer Perceptron (MLP) that maps latent noise into the original feature space. Within the framework of Generative Adversarial Networks (GANs) and Conditional GANs (CGANs), the generator serves as a transformation model that converts random noise into realistic synthetic data samples. Specifically, it takes as input a latent noise vector $z \sim \mathcal{N}(0, I)$along with a condition label y, and learns to generate samples $G(z, y)$that approximate the distribution of real data. The generator follows a feedforward neural network architecture consisting of an input layer, multiple hidden layers, and an output layer. The input is formed by concatenating a 32-dimensional noise vector with a one-hot encoded class label. This design enables the model to generate class-specific samples conditioned on y, which is particularly important in biomedical applications such as gene expression data modeling. The hidden layers consist of two fully connected layers with 128 and 256 neurons, respectively. These layers employ LeakyReLU activation functions along with Batch Normalization to stabilize training and accelerate convergence. Functionally, these hidden layers perform the core transformation by learning complex gene–gene interactions, capturing correlation structures inherent in biological systems, and modeling high-dimensional gene expression patterns. Through successive nonlinear transformations, the generator progressively converts random noise into biologically meaningful representations. The conditional information $y$is incorporated into the network through concatenation, allowing the generator to produce synthetic samples that are consistent with specific biological classes (e.g., leukemia, breast cancer, or healthy samples). The output layer produces a synthetic gene expression vector with the same dimensionality as the real data. Depending on the data characteristics, activation functions such as Tanh or Sigmoid may be used to ensure that the generated values fall within a valid range. The generator is trained to minimize the probability that the discriminator correctly identifies its outputs as fake. This is typically achieved using the non-saturating loss function, which encourages the generator to maximize the discriminator's confidence in generated samples being real. As shown in Equation (3), the generator loss is defined as:

$$L_G = -\frac{1}{m}\sum_{i=1}^{m} Log\ D(G(z^{(i)}, y^{(i)}), y^{(i)}) \quad (3)$$

c). Adversarial Optimization Objective

Networks (GANs) and Conditional GANs (CGANs), where the generator and discriminator are jointly optimized within a minimax framework. In this setting, both networks are trained simultaneously using the Adam optimizer with a learning rate of 0.0002. The generator aims to reduce the discriminator's ability to distinguish between real and generated samples by producing increasingly realistic synthetic data. In contrast, the discriminator seeks to maximize its classification accuracy by correctly identifying real samples as real and generated samples as fake. This interaction establishes a two-player adversarial game in which both models iteratively improve: the discriminator becomes more effective at classification, while the generator becomes more capable of generating data that closely resembles the true distribution. At convergence, the system reaches an equilibrium where the generated samples become nearly indistinguishable from real data, indicating that the generator has successfully captured the underlying data distribution. As shown in Equation (4), this process is mathematically formulated as a minimax optimization problem,

where the discriminator maximizes the objective function while the generator minimizes it:

$$\min_{G}\max_{D} V(D, G) = Ex\sim p_{data}[\log D(x|y)] + E_z \sim p_z[\log(1 - D(G(z|y)|y))]. \quad (4)$$

Binary Cross-Entropy (BCE) loss functions are commonly used to train both the generator and discriminator in Conditional Generative Adversarial Networks (CGANs). These loss functions quantify how effectively the discriminator distinguishes between real and generated samples, thereby guiding the adversarial training process. For the discriminator, the objective is to maximize the probability of correctly classifying real data pairs (x, y)as real and generated pairs $(G(z, y), y)$as fake. This is achieved by minimizing the negative log-likelihood of correct predictions. Equation (2) above illustrates the discriminator's loss. For the generator, the objective is to minimize its loss by producing synthetic samples that the discriminator classifies as real. In other words, the generator attempts to maximize the discriminator's output for generated data. Equation (3) above illustrates the generator's loss.

1) Label Preprocessing

A Label Encoder is first initialized to transform categorical class labels, if present, from their original string representation into numerical form. The encoder is fitted to the label vector y, producing an integer-encoded label vector y encoded. The integer labels are subsequently converted into one-hot encoded vectors. This representation is required for conditional GAN training, as it enables both the Generator and Discriminator to explicitly incorporate class information as conditioning variables. The number of distinct classes, denoted as num_classes, is inferred from the second dimension of the one-hot encoded label matrix. In addition, the input dimensionality input_dim, corresponding to the number of features in the dataset, is obtained from the shape of the feature matrix X.

2) Model Architectures (Inner Functions)

The generator and discriminator architectures were implemented using the Keras Functional API. The build generator (latent_dim, num_classes, output_dim) function defines the Generator network, which receives two inputs: a random noise vector from the latent space and a one-hot encoded class label. These inputs are concatenated to condition the data generation process on the target class label. The combined input is then passed through two fully connected hidden layers containing 128 and 256 neurons, respectively, each employing the ReLU activation function to introduce non-linearity and enable the learning of complex feature representations. The output layer consists of output_dim neurons with a linear activation function, allowing the generator to produce synthetic feature vectors with flexible continuous values consistent with the scaled real dataset. The resulting model maps the pair [noise, label] to a generated sample. Similarly, the build discriminator (input_dim, num_classes) function defines the Discriminator network, which takes as input either a real or generated feature vector along with its corresponding one-hot encoded class label. After concatenating the data and label inputs, the merged representation is processed through two hidden Dense layers containing 256 and 128 neurons, respectively, with ReLU activation functions. The final output layer contains a single neuron with a sigmoid activation function that estimates the probability of the input sample being real rather than synthetic. Consequently, the discriminator model maps the pair [data, label] to a real/fake probability score, enabling adversarial training between the two networks.

3) GAN Setup and Compilation

In this stage, the Conditional Generative Adversarial Network (CGAN) is formally constructed and prepared for adversarial training. First, the two core components of the architecture—the Generator (G) and the Discriminator (D)—are instantiated according to the predefined conditional design, where both networks incorporate class label information to guide data generation and discrimination. The Discriminator is then compiled using the Adam optimizer with a learning rate of 2×10^{-4}, a configuration widely adopted in GAN-based models due to its stability in adversarial optimization. The loss function used is binary cross-entropy, which is appropriate for the Discriminator's task of performing binary classification between real and synthetically generated samples conditioned on class labels. Before constructing the full cGAN model, the Discriminator's weights are frozen by setting D. trainable = False. This is a critical step that ensures that during the training of the combined model, only the Generator's parameters are updated, while the

Discriminator remains fixed and serves as a static adversarial evaluator. This prevents unintended updates to the Discriminator during Generator optimization. The full conditional GAN (combined model) is then defined by connecting the Generator and Discriminator sequentially. The model receives two inputs: a latent noise vector $z$and a class label y. The Generator transforms these inputs into synthetic samples $\hat{x} = G(z, y)$, which are then passed to the Discriminator along with the corresponding label to compute $D(\hat{x}, y)$, representing the probability that the generated sample is real under the given condition. Finally, the combined model is compiled using the same Adam optimizer configuration and binary cross-entropy loss function. In this setup, the Generator is trained indirectly through the Discriminator's feedback, with the objective of maximizing the probability that generated samples are classified as real. This adversarial objective drives the Generator to produce increasingly realistic and class-consistent synthetic data, thereby improving the overall fidelity of the conditional data generation process.

4) Adversarial Training Loop

The adversarial training process is executed iteratively over a predefined number of epochs, with progress monitored using trange for real-time visualization. This loop implements the fundamental minimax optimization framework of the Conditional Generative Adversarial Network (CGAN), where the Generator (G) and Discriminator (D) are trained in opposition under class-conditioning constraints.

5) Return Values

Returns the trained G model, the fitted le (Label Encoder), and the history dictionary containing the recorded losses, which can be used for performance monitoring and diagnostics.

C. Feature Extraction

The final feature subset is defined as: $S_{final} = \{j \mid \neq 0\}$, where only features with non-zero coefficients are retained after L1-regularized optimization. This process effectively removes irrelevant and redundant variables, resulting in a highly compact representation of the data. As a consequence, the method achieves feature reduction ratios exceeding 99%, significantly decreasing dimensionality while preserving the most informative features for downstream analysis and classification tasks. The most useful features were retained by performing the following procedure:

1) L1-Regularized Logistic Regression (LASSO)

This approach integrates logistic regression with an L1 penalty term, which enforces sparsity in the model coefficients. As a result, many coefficients are driven exactly to zero, allowing the model to retain only the most informative features while eliminating irrelevant or redundant ones. This sparsity-inducing property significantly enhances model interpretability, reduces the risk of overfitting, and improves generalization performance, particularly in high-dimensional settings such as gene expression data. By focusing on a subset of relevant features, LASSO facilitates more robust and biologically meaningful analysis. The optimization objective combines the negative log-likelihood of logistic regression with the L1 regularization term $\| \beta \|_1$, which controls the degree of sparsity through the regularization parameter λ. As shown in Equation (5), the objective function to be minimized is defined as:

$$\min_{\beta_0 \beta} \left[-\frac{1}{N} \sum_{i=1}^{N} (y_i \log(p_i) + (1 - y_i) \log(1 - p_i)) + \lambda \sum_{j=1}^{p} |\beta_j| \right] \quad (5)$$

Where:

- N is the number of samples in the augmented dataset.
- p is the number of features.
- y_i is the true class label (0 or 1) for sample i.
- $p_i = \sigma(\beta_0 + x_i^T \beta)$ is the predicted probability, with $\sigma(t)$=1/ (1+e^{-1}) being the sigmoid function.
- β_0 is the intercept, are the feature coefficients.
- λ is the regularization strength controlling sparsity. Features i where $\beta_i \neq 0$ are selected.

Sparsity in the model is governed by the regularization parameter λ, where larger values impose stronger penalties on the magnitude of the coefficients, thereby driving a greater proportion of them to zero. In contrast, smaller values of λ permit the retention of more features. In practice, the optimal value of λ is typically determined via cross-validation to achieve an appropriate balance between predictive accuracy and model parsimony. In the

domain of gene expression analysis, the number of features (genes) often substantially exceeds the number of available samples, leading to a high-dimensional, low-sample-size (HDLSS) setting. Under such conditions, LASSO demonstrates significant utility by effectively reducing the dimensionality of the feature space to a compact subset of biologically relevant genes. This reduction not only improves computational efficiency but also enhances model interpretability by facilitating the identification of potential biomarkers associated with specific phenotypic conditions. Furthermore, when applied to datasets augmented using Conditional Generative Adversarial Networks (CGANs), LASSO plays a critical role in refining the expanded data representation. Although CGAN-based augmentation alleviates issues related to data scarcity and class imbalance through the generation of synthetic samples, it may also introduce redundant or non-informative features. The application of L1-regularized logistic regression subsequent to data augmentation enables the selective retention of the most discriminative features across both real and synthetic samples. This process contributes to improved robustness and reliability in downstream classification tasks. The final predictive model is constructed based on the subset of features corresponding to non-zero coefficients (β_j), which represent the most significant variables for class discrimination. Consequently, the integration of CGAN-based data augmentation with LASSO-based feature selection constitutes a robust and effective framework for handling high-dimensional biological data, yielding improvements in both predictive performance and model interpretability.

2) Sample Space Enrichment

Refers to the process of expanding the original dataset by incorporating high-quality synthetic samples generated by the trained generator. After model convergence, the generator produces a synthetic cohort S_{syn}, which is then combined with the original real dataset $S_{real}$to form an enriched dataset S_{aug}, defined as $S_{aug} = S_{real} \cup S_{syn}$. This augmentation increases the diversity and size of the sample space, mitigates data scarcity and class imbalance, and ultimately improves the robustness and generalization capability of downstream machine learning models. This augmentation stabilizes feature evaluation by reducing sensitivity to individual sample variations. [20]

3) Deterministic Feature Selection via L1-Regularization

Deterministic feature selection is performed using L1-regularized logistic regression (LASSO), which is an embedded feature selection method particularly well-suited for high-dimensional datasets such as gene expression data. This approach imposes an L1 penalty on the model coefficients, encouraging sparsity by forcing less informative features to have zero weights, while retaining only the most discriminative ones. To efficiently optimize this objective in large-scale and sparse settings, the model is trained using the SAGA solver. It is an advanced stochastic gradient-based optimization algorithm that combines the benefits of fast convergence and scalability. It is particularly effective for convex problems with non-smooth regularization terms, such as L1 regularization, making it highly suitable for high-dimensional biological data. The optimization process aims to minimize a regularized loss function that balances classification accuracy and sparsity. Specifically, it combines the empirical loss over the augmented dataset with an L1 penalty term applied to the model coefficients. This formulation ensures that the model not only achieves accurate classification performance but also performs automatic feature selection by shrinking less informative coefficients toward zero. As shown in Equation (6), the optimization objective is defined as:

$$\min_{w} \left(\sum_{i=1}^{|Saug|} \mathcal{L}(yi, w^T xj) + \lambda \sum_{j=1}^{D} \left| w_j \right| \right) \quad (6)$$

This equation helps the model improve prediction accuracy while automatically selecting the most important features and reducing noise in high-dimensional data Minimizes classification error, where the loss term $\sum L(y_i, x_j)$ensures that the model learns to accurately predict the target labels y_i. And it enables automatic feature selection through the L1 regularization term $\lambda \sum | x_j |$, which drives less important feature coefficients toward zero, effectively removing irrelevant variables from the model. Hance it is particularly effective for high-dimensional data, such as gene expression datasets, where thousands of features must be handled efficiently. Finally, it helps

to reduce overfitting by penalizing unnecessary complexity, resulting in a simpler and more generalizable model with improved predictive performance. The use of the SAGA optimizer enables efficient convergence even in high-dimensional feature spaces, while maintaining the sparsity-inducing property of L1 regularization. As a result, the model can effectively identify a compact subset of relevant genes from the augmented dataset, improving both computational efficiency and interpretability. Solved using the SAGA algorithm with 5000 iterations. [21].

IV. EXPERIMENTAL RESULT AND EVALUATION

The proposed framework follows a systematic and reproducible experimental pipeline for each dataset. Initially, a stratified 80:20 split is performed to divide the dataset into training and testing subsets, ensuring that class distributions are preserved. The testing set remains completely independent throughout the process and is reserved exclusively for final evaluation to prevent information leakage. For each dataset, the experimental workflow begins with the initialization of the training process, where a status message is generated to indicate the start of dataset processing, thereby improving transparency and traceability during batch experiments. Subsequently, the Conditional Generative Adversarial Network (CGAN) is trained using only the training data. The CGAN training is conducted for 1000 epochs while retaining the default latent-space dimension and batch-size parameters. Upon completion, the training procedure returns the optimized generator model, the corresponding class encoder, and the complete training history containing both generator and discriminator loss values. To analyze training behavior and convergence stability, loss curves for the generator and discriminator are visualized using a dedicated plotting function. These plots are saved in PNG format and serve as diagnostic tools for monitoring convergence trends, identifying potential mode collapse, and assessing the adversarial equilibrium achieved during training. Once the CGAN training phase is completed, the trained generator is employed to produce synthetic samples. The number of generated instances is set equal to the size of the original training dataset, thereby maintaining balance between real and synthetic samples and ensuring consistency during subsequent analysis. The real and synthetic samples are then combined to form an augmented training dataset. Feature selection is subsequently performed using L1-regularized logistic regression, which promotes sparsity by eliminating less informative features while retaining those with the strongest predictive relevance. The selected features, reported in Table 2, are then used to train a Support Vector Machine (SVM) classifier. The trained classifier is evaluated exclusively on the independent real testing set to assess its generalization capability. Performance is quantified using classification accuracy and the Area Under the Receiver Operating Characteristic Curve (AUC), with the obtained results summarized in Table 3. For each dataset, the selected feature indices, classification metrics, and the complete CGAN training history are stored in a structured results dictionary to facilitate systematic comparison and ensure experimental reproducibility. After all datasets have been processed, the collected results are aggregated into a structured tabular format using the Pandas framework. Key evaluation metrics, including accuracy, AUC, and the number of selected features, are compiled to enable comparative and statistical analyses across datasets. Finally, the consolidated results are exported to a CSV file (e.g., gan_feature_selection_results.csv), providing a permanent and shareable record of the experiments for reporting, visualization, and further analysis.

Table 2. PROPOSED REDUCE DIMENSIONALTY

Dataset	Original features	Class	Selected feature
Breast	24.841	2	87
CNS	7.129	2	93
Colon	2.000	2	43
Leukemia	7.129	2	50
Leukemia-3	7.129	3	127
Leukemia-4	7.129	4	134
Lung	12.600	2	521
Lymphoma	4.026	3	111
MLL	12.502	3	66
Ovarian	15.154	2	79
SRBCT	2,308	4	105

A. Experimental details

All experimental evaluations were performed using an identical computational setup to guarantee a fair and reproducible comparison between the investigated approaches. The experiments were executed on a system equipped

with an NVIDIA GeForce RTX 3060 GPU featuring 16 GB of dedicated memory and 32 GB of RAM. GPU-based acceleration was employed during the training phases of the CGAN model and the proposed CGAN+L1 framework to improve computational efficiency. The implementation was developed using Python 3.10.20 with TensorFlow-directml-plugin: 0.4.0. dev230202 and scikit-learn 1.7.2

V. RESULTS AND DISCUSSION

This research proposes a framework based on Conditional Generative Adversarial Networks (CGANs) to enhance feature selection and classification outcomes in high-dimensional, low-sample-size (HDLSS) datasets. The framework addresses the limitations caused by insufficient training samples by generating synthetic instances through a CGAN trained solely on the training portion of the data. In the current study, our primary focus was to develop and evaluate an internal filtering mechanism for CGAN-generated synthetic samples and to assess its impact on feature selection and downstream classification performance. These synthetic samples are subsequently merged with the original training data, producing an enriched dataset that increases sample diversity while preserving the underlying class distributions. To guide the feature selection process, the candidate features were initially ranked using the combined statistical score described in Section III, which integrates Chi-Square (χ^2), Mutual Information (MI), and ANOVA F-test measures. This preliminary ranking step reduces the search space by identifying the most informative features before applying the subsequent selection mechanism. To extract the most relevant features, an L1-regularized logistic regression model is employed as the feature selection mechanism. This approach encourages sparsity by removing non-informative and redundant variables, thereby retaining only features with significant predictive importance. The resulting feature subset is then used to train a Support Vector Machine (SVM) classifier. Model evaluation is conducted using a separate testing set that remains unseen during training, ensuring a reliable assessment of the model's generalization performance. The results reported in Tables 3 and 4 indicate that the proposed framework successfully reduces the dimensionality of the datasets without compromising predictive effectiveness. The achieved accuracy and AUC scores demonstrate that the selected features retain the critical information necessary for accurate classification. In addition, the incorporation of CGAN-generated synthetic data helps mitigate the challenges associated with limited sample sizes, leading to a more stable and reliable feature selection process. Overall, the combination of CGAN-based data augmentation, L1-regularized feature selection, and SVM classification constitutes a robust and reproducible methodology for handling HDLSS datasets and enhancing classification performance

TABLE 3. ACCURACY OF THE PROPOSED METHOD

Dataset	Original features	Selected features	Accuracy	AUC	CGAN+L1 (min)
Breast	24,481	87	80	76.77	4.08
CNS	7,129	93	75	68.75	3.50
Colon	2,000	43	69.23	72.5	3.38
Leukemia	7,129	50	100	100	3.57
Leukemia-3c	7,129	127	100	100	4.72
Leukemia-4c	7,129	134	80	94.16	8.16
Lung	12,600	521	95.12	98.68	11.29
Lymphoma	4,026	111	92.86	100	4.60
MLL	12,582	66	93.33	98.67	4.08
Ovarian	15,154	79	100	100	5.22
SRBCT	2,308	105	100	100	6.64
Overall Average			**89.59**		**59.25**

TABLE 4. F-SCORE OF THE PROPOSED METHOD

Dataset	Original features	Selected features	AUC	F1-Score
Breast	24,481	87	76.77	79.17
CNS	7,129	93	68.75	69.75
Colon	2,000	43	72.5	63.89
Leukemia	7,129	50	100	100
Leukemia-3c	7,129	127	100	100
Leukemia-4c	7,129	134	94.16	58.54
Lung	12,600	521	98.68	87.89
Lymphoma	4,026	111	100	91.58
MLL	12,582	66	98.67	92.59
Ovarian	15,154	79	100	100
SRBCT	2,308	105	100	100
Overall Average				**85.76%**

The framework demonstrated resilience against typical data challenges, including outliers, missing values, and class imbalance, by leveraging the generative capacity of GANs and the end-to-end

learning abilities of neural networks. Empirical results confirmed the effectiveness of the approach: the proposed model achieved a classification accuracy of 89.59%, outperforming the best benchmark method, which reached 86.10%. These results demonstrate the proposed framework's superiority in balancing accuracy and computational efficiency and affirm its potential for broader application in biomedical and high-dimensional data scenarios. Table 5 shows the comparison between the proposed model with the state-of-the-art model. It is clear that our proposed model gained the best performance for both feature reduction and metrics. Compared to the baseline ensemble–DE approach [4], the proposed GAN-augmented deep learning framework achieves consistent improvements in both classification accuracy and AUC, demonstrating superior discriminative capability across HDLSS datasets. Hence, class reparability with higher predictive power in our proposed approach.

TABLE 5. PERFORMANCE AND FEATURES SUMMARY: DE vs CGAN+L1

Metric	DE	CGAN+L1	Delta(%)
Accuracy	86.10%	89.59%	+4.05%
F1-score	84.73%	85.76%	+1.22%
AUC	90.31%	91.78%	+1.63%
Avg Selected Features	**149.18**	**120.09**	**-19.50%**

The results demonstrate consistent improvement in both classification accuracy and AUC. In particular, the proposed method achieves an average accuracy of approximately 89.59% and an AUC close to 91.8, outperforming the baseline method, which attains around 86.10% accuracy and 90.3 AUC, the proposed framework consistently outperforms the DE benchmark across all evaluation metrics, yielding improvements of 3.49% in Accuracy, 1.04% in F1-score, and 1.46% in AUC. These gains highlight the effectiveness of GAN-based data augmentation and deep learning–based feature selection in enhancing classification performance on HDLSS datasets.

VI. CONCLUSION

In summary, the proposed framework demonstrates the potential of combining deep learning and generative modeling to overcome the key challenges of feature selection in HDLSS environments. The proposed framework was developed in response to the key limitations of traditional methods, particularly their struggles with optimizing the trade-off between accuracy and computational efficiency. The existing methods often suffer from computational inefficiency and statistical instability, especially in high-dimensional settings. The proposed approach employs CGAN-based data augmentation to mitigate sample sparsity and stabilize the feature selection process, thereby enabling improved model generalization. In addition to generative augmentation, the approach integrates deep learning–based embedded feature selection techniques, including Sparse Autoencoders, L1-Regularized Neural Networks, and Concrete Feature Selectors, to further reduce dimensionality while maintaining high predictive performance. Enhancing both model interpretability and robustness. The significant performance improvement over benchmark methods underscores its practical value and lays a strong foundation for future research in hybrid deep learning-based feature selection systems. Future research will concentrate on strengthening feature selection methods, incorporating explainable AI techniques, increasing CGAN stability, and validating the framework on bigger and more varied biomedical datasets.

REFERENCE

[1] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," IEEE Trans. Evol. Comput., vol. 20, no. 4, pp. 606–626, 2016.
[2] Y. Zhong, P. Chalise, and J. He, "Nested cross-validation with ensemble feature selection and classification model for high-dimensional biological data," Commun. Stat. Simul. Comput., vol. 52, no. 1, pp. 110–125, 2023.
[3] J. Zhang, Y. Xiong, and S. Min, "A new hybrid filter/wrapper algorithm for feature selection in classification," Anal. Chim. Acta, vol. 1080, pp. 43–54, 2019.
[4] A. K. Mandal, M. Nadim, H. Saha, T. Sultana, M. D. Hossain, and E.-N. Huh, "Feature subset selection for high-dimensional, low sampling size data classification using ensemble feature selection with a wrapper-based search," IEEE Access, 2024.
[5] S. Gupta and S. K. Mehta, "Feature selection for dimension reduction of financial data for detection of financial statement frauds in context to Indian companies," Global Bus. Rev., vol. 25, no. 2, pp. 323–348, 2024.
[6] M. A. Gutiérrez-Mondragón, A. Vellido, and C. König, "A study on the robustness and stability of explainable

deep learning in an imbalanced setting: The exploration of the conformational space of G protein-coupled receptors," Int. J. Mol. Sci., vol. 25, no. 12, Art. no. 6572, 2024.
[7] C. Cao, Q. Zhang, and Y. Deng, "A contrast-based feature selection algorithm for high-dimensional datasets in machine learning," Inf. Sci., Art. no. 122308, 2025.
[8] J. Nalić, G. Martinović, and D. Žagar, "New hybrid data mining model for credit scoring based on feature selection algorithm and ensemble classifiers," Adv. Eng. Informat., vol. 45, Art. no. 101130, 2020.
[9] Q. Wu, Y. Lin, T. Zhu, and Y. Zhang, "HIBoost: A hubness-aware ensemble learning algorithm for high-dimensional imbalanced data classification," J. Intell. Fuzzy Syst., vol. 39, no. 1, pp. 133–144, 2020.
[10] C. W. Chen, Y. H. Tsai, F. R. Chang, and W. C. Lin, "Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results," Expert Syst., vol. 37, no. 5, Art. no. e12553, 2020.
[11] S. S. Kshatri, D. Singh, and D. S. Sisodia, "Multiple aggregation model using ensembles classifier for feature selection: A homogeneous approach," in Proc. IEEE Int. Conf. Technol., Res., Innov. Betterment Soc. (TRIBES), 2021, pp. 1–6.
[12] M. Hamim, I. El Mouden, M. Ouzir, H. Moutachaouik, and M. Hain, "A novel dimensionality reduction approach to improve microarray data classification," IIUM Eng. J., vol. 22, no. 1, pp. 1–22, 2021.
[13] J. Kim, J. Kang, and M. Sohn, "Ensemble learning-based filter-centric hybrid feature selection framework for high-dimensional imbalanced data," Knowl.-Based Syst., vol. 220, Art. no. 106901, 2021.
[14] E. Jaw and X. Wang, "Feature selection and ensemble-based intrusion detection system: An efficient and comprehensive approach," Symmetry, vol. 13, no. 10, Art. no. 1764, 2021.
[15] M. Mera-Gaona, D. M. López, R. Vargas-Canas, and U. Neumann, "Framework for the ensemble of feature selection methods," Appl. Sci., vol. 11, no. 17, Art. no. 8122, 2021.
[16] S. Bashir, I. U. Khattak, A. Khan, F. H. Khan, A. Gani, and M. Shiraz, "A novel feature selection method for classification of medical data using filters, wrappers, and embedded approaches," Complexity, vol. 2022, Art. no. 8190814, 2022.
[17] A. Kumar, A. Kaur, P. Singh, M. Driss, and W. Boulila, "Efficient multiclass classification using feature selection in high-dimensional datasets," Electronics, vol. 12, no. 10, Art. no. 2290, 2023.
[18] J. Liu et al., "A new hybrid algorithm for three-stage gene selection based on whale optimization," Sci. Rep., vol. 13, no. 1, Art. no. 3783, 2023.
[19] J. Rufino et al., "Performance and explainability of feature selection-boosted tree-based classifiers for COVID-19 detection," Heliyon, vol. 10, no. 1, Art. no. e23219, 2024.
[20] I. J. Goodfellow et al., "Generative adversarial nets," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 27, 2014.
[21] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 27, 2014.
[22] Z. Zhang, Q. An, Y. Wang, C. Wu, B. Dong, and C. Zhou, "A high-dimensional feature selection algorithm based on multiobjective differential evolution," arXiv preprint arXiv:2505.05727, 2025.
[23] N. Taheri and H. Nezamabadi-pour, "A hybrid feature selection method for high-dimensional data," in Proc. 4th Int. Conf. Comput. Knowl. Eng. (ICCKE), 2014.
[24] J. Wang, J. Xu, C. Zhao, Y. Peng, and H. Wang, "An ensemble feature selection method for high-dimensional data based on sort aggregation," Syst. Sci. Control Eng., vol. 7, no. 2, pp. 32–39, 2019.