scientific reports

Check for updates

OPEN

Optimizing imbalanced learning with genetic algorithm

Muhammad Usman Safder[1], Syed Sarib Naveed[1], Khawar Khurshid[1]✉, Ahmad Salman[2,4] & Imran Fareed Nizami[3]

Training AI models on imbalanced datasets with skewed class distributions poses a significant challenge, as it leads to model bias towards the majority class while neglecting the minority class. Various methods, such as Synthetic Minority Over Sampling Technique (SMOTE), Adaptive Synthetic Sampling (ADASYN), Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), have been employed to generate synthetic data to address this issue. However, these methods are often unable to enhance model performance, especially in case of extreme class imbalance. To overcome this challenge, a novel approach to generate synthetic data is proposed which uses Genetic Algorithms (GAs) and does not require large sample size. It aims to outperform state-of-the-art methods, like SMOTE, ADASYN, GAN and VAE in terms of model performance. Although GAs are traditionally used for optimization tasks, they can also produce synthetic datasets optimized through fitness function and population initialization. Our synthetic data generation approach analyzes the Simple as well as the Elitist Genetic Algorithms, along with Logistic Regression and Support Vector Machines to evaluate the population initialization and fitness function. Experimental results across three datasets (Credit Card Fraud Detection, PIMA Indian Diabetes, and PHONEME) demonstrate that the proposed method significantly outperforms the previous techniques based on the commonly used performance metrics, including accuracy, precision, recall, F1-score, ROC-AUC, and AP (Accuracy-Precision) curve. This highlights the potential of GAs in the development of accurate and reliable AI models for imbalanced datasets.

In recent years, Neural Networks are considered one of the most significant breakthroughs in the field of Machine Learning. Despite being widely used, Neural networks heavily rely on the quality and distribution of the training data[1]. A significant challenge they face is the imbalanced nature of many datasets, where the number of instances across different classes is unevenly distributed. This imbalance leads to biased model predictions, favoring the majority class while neglecting the minority class. Such bias can severely impact the model's overall performance and accuracy, especially in critical areas like medical diagnosis or anomaly detection, where minority instances are vital. Recent applications in healthcare shows the critical importance of addressing class imbalance, such as in predicting mechanical ventilation outcomes and mortality rates[2], orthopedic disease classification[3], cardiovascular disease detection[4], and lung cancer classification[5]. To address this problem, various methods have been developed over time, which are broadly classified into three categories, data-level methods that modify data samples, algorithm-level methods that adjust the learning algorithms, and hybrid methods that combine both approaches[6].

The data-level methods allow the use of standard machine learning architectures and pipelines which has made them widely popular. Standard data-level methods include random over-sampling, which increases the number of minority class instances through random duplication, and random under-sampling, which decreases the number of majority class instances by randomly discarding samples of this class. The effectiveness of over-sampling versus under-sampling has been widely researched. Some studies have found under-sampling to be more advantageous in certain situations, while others suggest that a combination of both techniques effectively addresses imbalanced datasets. For example, a boosting-based approach that incorporates both over-sampling and under-sampling to handle imbalanced data has been used in a study[7].

A comparative study[8] of several over-sampling and under-sampling methods shows that the performance varies based on the dataset and the classifier used. Synthetic Minority Over-sampling Technique (SMOTE)[9] generates synthetic samples for the minority class by interpolating between existing minority class instances. Borderline SMOTE is a variant of SMOTE[10], which performs synthetic instance generation near the decision

[1]Department of Computer Science, Namal University, Mianwali, Punjab 42250, Pakistan. [2]School of Computing, Skyline University College, 1797 Sharjah, United Arab Emirates. [3]Department of Electrical Engineering, Bahria University, Islamabad, ICT 44000, Pakistan. [4]School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan. ✉email: khawar.khurshid@namal.edu.pk

boundary. Other data-level methods, such as Adaptive Synthetic Sampling (ADASYN)[11], Edited Nearest Neighbor (ENN)[12], and Cluster-Based Over-Sampling (CBO)[13], have also been proposed.

Algorithm-level methods modify the learning process to handle imbalanced data, allowing the possibility of more customized and specific solutions. Several methods fall in this category, such as Cost-Sensitive Learning[14], which modifies the learning algorithm to assign higher costs to misclassifications of the minority class. Class weightings, ensemble methods[15], Boosting[16], Kernel-Based Methods[17] have also been applied. The Active Learning approach[18] has been employed as well, where the learning algorithm actively selects the most informative samples to label, focusing on minority class instances.

Some hybrid methods that combine both data-level and algorithm-level approaches have also been utilized. Another approach to data generation is Variational Autoencoders (VAEs)[19], which are a type of generative model that combines the principles of deep learning and Bayesian inference to generate new data samples that resemble the training data. Although different from the traditional data-level and algorithm-level methods mentioned above, their behavior and a comparative analysis with our approach is also presented. While all the above-mentioned data generation methods find applications in a wide range of tasks, they still have some limitations, due to which optimal model performance is not achieved. One major issue with methods like SMOTE is the higher probability of overfitting, as synthetic instances are created by interpolating between the minority instances of the dataset. This leads to the creation of models that generalize well over the training data but perform poorly when presented with unseen test data[20]. A by product of overfitting is noise amplification, especially if noise is present within the minority class, further impacting the accuracy of the trained model. While some methods like CBO aim to reduce overfitting by encapsulating the synthetic data generation process within clusters, the probability of overfitting remains high if the data is of a higher dimension. Moreover, the effectiveness of CBO is heavily dependent on the clustering step. Inaccurate selection of the clustering algorithm or its parameters can result in clusters that do not appropriately represent the structure of the underlying data[21]. Algorithm-level methods often introduce additional complexity to the training process. For example, Cost-Sensitive Learning requires the incorporation of a cost matrix when modifying the learning algorithm, which makes training comparatively more complex[22].

While hybrid methods such as Tomek Link and SMOTE[23] address some of these drawbacks, they are often computationally quite extensive during the training process, especially when dealing with large datasets or real-time applications. In this research, a novel approach is proposed which uses Genetic Algorithms (GAs)[24] for producing synthetic data for the ANNs in data-constrained environments. GAs, modeled after the phenomenon of natural selection, have found wide usage in various domains, most commonly in optimization and search problems. Their ability to explore a large search space and evolve solutions makes them a potential candidate for solving many real-world problems, including noise removal[25], transportation and logistics[26], image segmentation[27], routing problems[28], and manufacturing services[29]. GAs have also shown promise in biomedical signal processing, particularly in EEG-based classification tasks where optimization of feature selection and classification accuracy is crucial[30].

Over the years, GAs have also been used for synthetic data generation by other researchers. For instance, synthetic datasets have been generated using GAs for testing and validating software systems, ensuring diverse and comprehensive test cases[31,32]. Additionally, realistic synthetic data has been applied to privacy-preserving data publishing, balancing the trade-off between data utility and privacy[33]. In the field of intrusion detection, synthetic attack data has been utilized for the development and testing of robust detection systems[34].

Another study presents a method combining WCGAN-GP for synthetic attack data generation and Genetic Algorithms (GA) for feature selection to enhance Intrusion Detection Systems (IDS)[35]. While these applications are fairly useful, Genetic Algorithms (GAs) have not been specifically used to generate synthetic training data for Artificial Neural Networks (ANNs) in order to improve their performance and to mitigate the effect of imbalanced training data[36]. In this study, GAs are used to produce synthetic data while simultaneously addressing the limitations of existing data generation methods and improving ANN performance. Initially, a fitness function that accurately captures the underlying characteristics of the data is developed. Since a precise mathematical description of the data is difficult to achieve analytically, therefore, the process of creating the fitness function is automated. This approach utilizes Support Vector Machines (SVM)[37] and logistic regression[38] to fit a model to the data generating the equations for the underlying data distribution, and creating the fitness functions to maximize the minority class representation. The synthetically generated data is then used to train the neural network using three benchmark datasets all of which contain binary imbalanced classes.

To validate the proposed method, a comprehensive comparative analysis of various techniques is conducted, including different variants of Genetic Algorithms, such as Simple GA, Elitist GA, and SVM-based GA, alongside previously established methods, such as SMOTE and ADASYN. To measure the effectiveness of these techniques, several evaluation metrics, including accuracy, precision, recall, F1-score, ROC-AUC, and average precision (AP) are used. These metrics provide a comprehensive view of the model's performance, particularly in terms of its ability to correctly classify minority class instances without compromising the overall accuracy.

Related work

Handling imbalanced data

Imbalanced data pose a significant challenge in machine learning, impacting the performance and accuracy of predictive models[39]. Researchers have explored various strategies to mitigate this challenge due to its significant implications[6].

A detailed comparison of techniques for managing unbalanced data in machine learning, particularly in the context of electricity theft detection, is provided by[40]. Another study evaluates several machine learning methods aimed at overcoming the obstacles presented by extremely unbalanced datasets in industrial quality control[41]. Additionally, research has applied Random Forest algorithms to imbalanced datasets, enhancing

detection accuracy and reliability, as seen in network monitoring analysis[42,43]. The Synthetic Minority Oversampling Technique (SMOTE)[9] is a widely used method that generates synthetic samples for the minority class, thus balancing the dataset and improving model learning effectiveness across both classes.

Synthetic data generation

Synthetic data generation techniques serve as viable solutions for addressing class imbalance in datasets. Methods such as Adaptive Synthetic (ADASYN)[11] and Borderline-SMOTE[10] have emerged as effective oversampling techniques.

ADASYN In[11], ADASYN is employed to tackle imbalanced datasets by generating synthetic samples for minority class instances. The core concept of ADASYN is to utilize a weighted distribution for different minority class samples, generating more synthetic data for instances that are harder to learn. This approach reduces bias from class imbalance and adapts the classification decision boundary toward difficult examples, ultimately enhancing classifier performance. ADASYN is applied in combination with Random Forest to effectively identify fraud in telecommunication, showcasing the technique's versatility in handling imbalanced datasets[44].

*Borderline-SMOTE*In[10], Borderline-SMOTE is used to address imbalanced datasets by generating synthetic samples along the decision boundary between classes. This method, an extension of SMOTE, focuses on oversampling only the minority class instances near the borderlines, where class imbalance is most pronounced. By targeting these critical instances, Borderline-SMOTE enhances the classifier's ability to differentiate between classes, resulting in improved true positive rates and F-values compared to traditional SMOTE and random oversampling methods.

Cost-Sensitive Learning The foundations of cost-sensitive learning are explored in[14], with emphasis on the need to incorporate misclassification costs into the learning process. It is illustrated how traditional classification algorithms can be adapted to minimize total costs rather than merely focusing on error rates, providing a more practical approach for scenarios with varying misclassification costs.

Genetic Algorithms for Data Generation The use of Genetic Algorithms (GAs) for data generation is investigated in[31]. Recent work includes[5] applying Greylag goose optimization with multilayer perceptron for lung cancer classification, and[45] using Game Shapley local search embedded binary social ski-driver optimization for cancer classification from RNA sequencing data. A hybrid approach is described in[46] that combines GAs with reinforcement learning to automate software test data generation. Since significant development costs are associated with software testing, substantial savings can be achieved through automation, especially in complex domains. Despite the success of GAs in generating simple test data, their application to more complex data types such as images, videos, sounds, and 3D models is rarely explored.

Machine Learning and Data Augmentation In[47], a Generative Adversarial Network (GAN) is utilized for data augmentation, highlighting its ability to generate synthetic data without predefined classes. This method is proven effective for infrared small target detection, outperforming real data. However, it is acknowledged that GANs present challenges, including high training complexity which requires substantial computational power and time to converge. Additionally,[48] proposes a three-phase hybrid soft computing approach for cancer classification of gene expression micro-array data. The article[49] provides a systematic review of machine learning techniques in cancer classification for imbalanced medical datasets.

In another study[50], GAN-based data augmentation is applied to preserve image objects and maintain translation consistency. Although effective, GAN models can be complex to implement and interpret, and evaluating generated data quality poses ongoing challenges. Furthermore, GANs are susceptible to overfitting, particularly with limited augmentation scope, and their success largely depends on the quality and diversity of the original training data.

Active Learning in Imbalanced Data Classification In[18], Active Learning is introduced as a novel approach to address imbalanced data. It is demonstrated that active learning can effectively resolve class imbalance by providing the learner with more balanced classes. This method selects informative instances from a smaller sample pool, eliminating the need for exhaustive searches through the entire dataset, resulting in an efficient querying system applicable to large datasets.

Oversampling for Imbalanced Data Classification In[51], an oversampling method is implemented using an Adversarial Network to tackle class imbalance. A synthetic minority dataset is generated via a black-box oversampler, followed by refinement with a network trained using adversarial loss. Striking a balance between generating realistic synthetic data and maintaining data quality is crucial, as is improving classification performance by ensuring synthetic data closely resembles minority class instances.

SMOTE-Boost-based Sparse Bayesian Model In[52], the SMOTE algorithm is employed in combination with a boosting procedure to address imbalanced datasets. The proposed model comprises three modules: SMOTE-based data enhancement, an AdaBoost training strategy[53], and sparse Bayesian model construction[54]. The SMOTE algorithm is utilized for data enhancement, generating additional minority samples to reduce imbalance, while a specific AdaBoost strategy is applied to adaptively enhance predictive ability and mitigate overfitting, ultimately improving learning from minority class instances.

Tomek Link and SMOTE Approaches In[23], SMOTE and Tomek Link techniques are combined to address challenges in imbalanced dataset classification. These methods are applied alongside various classifiers, including Naïve Bayes, support vector machines, and k-nearest neighbors, to enhance classification performance. Our study investigates their application in condition monitoring systems for electrical machines, revealing the practical challenges posed by imbalanced data. Results indicate that combining SMOTE with Tomek Link improves performance across all classifiers, particularly k-nearest neighbors, thereby enhancing classification accuracy in scenarios with limited fault data.

Diffusion-based Synthetic Data Generation In[55], the labor-intensive task of preparing training data for deep vision models is tackled by leveraging generative models to produce synthetic data. Unlike traditional models

that generate image-level category labels, a novel approach is employed using the text-to-image generative model, Stable Diffusion (SD), to create pixel-level semantic segmentation labels. By utilizing text prompts, cross-attention, and self-attention mechanisms, three innovative techniques are introduced: class-prompt appending, class-prompt cross-attention, and self-attention exponentiation. These techniques generate segmentation maps corresponding to synthetic images, serving as pseudo-labels for training semantic segmenters and minimizing the need for labor-intensive pixel-wise annotation.

Privacy-Preserving Data Publishing In[56], an information-driven distributed genetic algorithm (ID-DGA) is presented for optimal anonymization through attribute generalization and record suppression. The proposed study addresses the privacy-preserving data publishing (PPDP) problem by incorporating various components, including an information-driven crossover operator, mutation operator, improvement operator, and a two-dimensional selection operator. Additionally,[57] introduces a technique for image captioning using hierarchical clustering and deep learning.[58] presents improved GA-based clustering with new selection methods for categorical dental data.

Our contribution

Our work contributes significantly to synthetic data generation and class imbalance handling in machine learning:

- We introduce the first GA-based synthetic data generation approach that systematically integrates SVMs and logistic regression within both the initialization and fitness evaluation phases. Unlike existing GA applications in data generation[31,36,47] that rely on random initialization and simplistic fitness functions, our method uses SVM decision boundaries to intelligently initialize populations near classification boundaries and employs logistic regression-based fitness evaluation to ensure synthetic samples contribute meaningfully to model performance. This dual-model integration represents a significant departure from traditional GA approaches and addresses the critical limitation of generating synthetic samples that may not enhance classification performance.
- A critical limitation of existing oversampling techniques is addressed by developing a distribution-aware synthetic data generation method. Unlike SMOTE and ADASYN, which rely on simple interpolation between neighboring points, the GA-based approach introduces controlled variations in the feature space. This allows for better exploration of potential minority class instances while maintaining the core characteristics of the original data distribution, as shown in the KDE plot analyses.
- A comprehensive comparison is presented between the Simple Genetic Algorithm (SGA) and SVM-guided GA approaches for synthetic data generation. The experiments reveal that incorporating SVM decision boundaries enhances the effectiveness of synthetic samples, particularly near classification boundaries where discrimination is crucial. The SVM-guided approach consistently outperforms both traditional oversampling methods and basic GA implementations across the evaluation metrics.
- It is demonstrated that the GA-based framework enhances model generalization through strategic synthetic data generation. By incorporating machine learning models in the fitness evaluation process, it is ensured that synthetic samples meaningfully contribute to the learning process rather than merely balancing the dataset numerically. This strategy addresses common challenges in synthetic data generation, such as amplifying noise or producing unrealistic samples that do not aid in classification.
- The experimental results across multiple datasets highlight the value of this approach, particularly in severe class imbalance scenarios where traditional methods often struggle.

Dataset

The proposed algorithm is evaluated on three benchmark datasets, with their class distributions summarized in Table 1.

Credit card fraud detection

The Credit Card Fraud Detection dataset[59] comprises credit card transactions made by European cardholders over two days in September 2013. With a total of 284,807 samples, only 492 transactions (0.172%) are classified as fraudulent, resulting in a highly imbalanced class distribution. This dataset contains 30 features.

Pima Indian diabetes

The Pima Indian Diabetes dataset[60] provides information about diabetes cases within a population near Phoenix, Arizona. It consists of 768 samples divided into two classes: positive and negative cases. The minority class contains 268 samples (34.9% of the total), while the majority class comprises 500 samples. This dataset includes a total of 8 features.

Dataset	Majority class	Minority class	Ratio
Credit card fraud detect	284,315	492	578:1
Pima Indian diabetes	500	268	1.87:1
Phoneme	3818	1586	2.41:1

Table 1. Class distributions and imbalance ratios of benchmark datasets.

Phoneme

This dataset[61] contains 5404 samples and 5 features, with two classes distinguishing between nasal (class 0) and oral (class 1) sounds. The minority class has 1586 samples (29.35% of all samples), while the majority class has 3818 samples.

Proposed algorithm

In proposed methodology, various machine learning (ML) algorithms, including logistic regression and support vector machines (SVM) are utilized to generate optimal equations for the dataset, which are subsequently integrated into the Genetic Algorithms (GAs) for population initialization and fitness function evaluation, as shown in Fig. 1. The process begins with the *Original Data*, which consists of the raw dataset to be processed. This data is passed into the machine learning models for feature extraction and predictive modeling.

Theoretical Foundations of GAs for Synthetic Data Generation Genetic Algorithms (GAs) are particularly effective for synthetic data generation in imbalanced datasets due to their evolutionary optimization, which addresses the challenges of sparse minority class instances and complex distributions. The schema theorem[24] formalizes GAs' ability to explore high-dimensional feature spaces:

$$\xi(S,t+1) \geq \xi(S,t) \cdot \frac{f(S)}{\bar{f}} \cdot \left(1 - p_c \cdot \frac{\delta(S)}{l} - p_m \cdot o(S)\right) \tag{1}$$

where $\xi(S,t)$ is the number of schema S instances at generation t, $f(S)$ is schema fitness, $\bar{f}$ is average fitness, p_c and p_m are crossover and mutation probabilities, $\delta(S)$ is the defining length, is chromosome length, and $o(S)$ is schema order. This ensures efficient navigation of non-linear minority class distributions, unlike SMOTE and ADASYN's local interpolation, which may generate noisy samples[52,62]. Our fitness function (Eqs. 4, 9) prioritizes misclassified minority instances, guided by fitness proportionate selection:

$$P(x_i) = \frac{f(x_i)}{\sum_j f(x_j)} \tag{2}$$

This focuses synthetic data on challenging classification regions, enhancing model performance. Single-point crossover (Eq. 10) and mutation (Eq. 11) introduce controlled diversity, ensuring synthetic samples explore new feature space regions while remaining representative, unlike GANs and VAEs, which struggle with sparse data due to training instability and latent space assumptions[63]. GAs' robustness to noise, modeled as:

$$E[L] = \int L(h(x), y)\, p(x, y)\, dx\, dy \tag{3}$$

mitigates outlier effects, outperforming deep generative models in data-constrained settings. These mathematical foundations provide rigorous theoretical support, validated by superior F1-scores (Section 7), justifying GAs' effectiveness for class imbalance.

The output of the ML algorithms serves as the starting population for the GA. The next phase is *Population Initialization*, where the initial solutions from the ML models are introduced as candidates in the GA population. The GA then proceeds through an iterative process, beginning with *Selection*, where candidate solutions are evaluated based on a fitness function. This fitness function is derived from the ML models' predictions, ensuring that only the most optimal solutions are retained. To introduce diversity and explore the solution space more thoroughly, *Mutation* is applied, where random alterations are made to certain solutions. Following mutation,

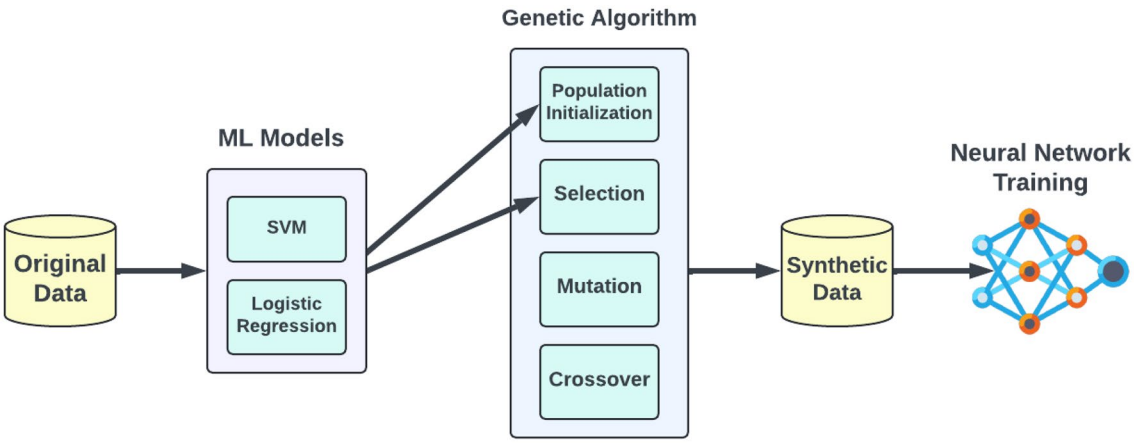

Fig. 1. Methodology framework for synthetic data generation using a genetic algorithm approach. This figure illustrates the process flow of the proposed methodology, from original data to training an NN with synthetic data via a Genetic Algorithm.

Crossover is used to combine features from selected solutions, producing new off-springs that inherit traits from multiple parent solutions. After applying the GA process, synthetic data is generated, which is used to train a neural network model, improving its performance by providing a more diverse training set.

ML algorithms

Logistic Regression Logistic regression is widely used in classification analysis that models the probability of a binary outcome based on one or more variables. It is particularly suitable for cases where the dependent variable can take on two possible outcomes. The logistic regression model is given in Eq. (4).

$$\mathrm{P(y)} = \mathrm{Sigmoid}(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p) \tag{4}$$

where $\mathrm{P(y)}$ is the probability of the class for the dependent variable y being 1, β_0 is the intercept, and $\beta_1, \beta_2, \ldots, \beta_p$ are the coefficients of the independent variables $x_1, x_2, \ldots, x_p$.

Logistic regression is selected due to its effectiveness in handling binary classification problems[64] and its straightforward implementation and interpretability which make it a suitable choice for initial population generation and fitness evaluation in GAs. The logistic regression model is trained on the dataset to produce the coefficients β and the intercept β_0, which are then used to initialize the GA population and evaluate the fitness of individuals.

Support Vector Machines (SVM) Support Vector Machines (SVM) are powerful supervised learning models used for classification[65] and regression tasks. SVMs aim to find the optimal hyperplane that best separates data points of different classes in a high-dimensional space. The optimal hyperplane is defined by maximizing the margin between the closest data points of the classes, known as support vectors. SVMs have also been used for the imbalanced learning problems. For example, in[66] support vector machine ensemble is utilized to effectively classify imbalanced data. The decision function for a logistic SVM is given in Eq. (5):

$$f(\mathrm{x}) = \mathrm{w} \cdot \mathrm{x} + b \tag{5}$$

where w is the weight vector, x is the input feature vector, and b is the bias term. The objective is to minimize the optimization problem as given in Eq. (6):

$$\min_{\mathrm{w},b} \frac{1}{2}\|\mathrm{w}\|^2 + C\sum_{i=1}^{n} \xi_i \tag{6}$$

subject to the constraints as given in Eq. (7):

$$y_i(\mathrm{w} \cdot \mathrm{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n \tag{7}$$

Here, y_i represents the class labels, ξ_i are the slack variables that allow for misclassifications, C is the regularization parameter that controls the trade-off between maximizing the margin and minimizing the classification error, and n is the number of training samples.

Although nonlinear kernels such as RBF could potentially model more complex minority class distributions, we use linear SVMs to maintain computational efficiency during population initialization. This choice ensures scalability across large datasets without incurring the heavy computational cost associated with kernel-based methods.

GA components

While GAs have traditionally been known for optimization problems, in this study, they are utilized for generating an optimized population for the minority class. The two types of GAs used are Simple Genetic Algorithm (SGA) and Elitist Genetic Algorithm (EGA)[67], and their performance is compared using a range of evaluation metrics, providing a comprehensive view of their effectiveness. The key components of GAs that impact their performance are Population Initialization, Fitness Function, Parent Selection, Crossover, and Mutation. These components are designed to enhance the effectiveness of the proposed approach.

Population Initialization Population Initialization is the first and one of the most crucial steps in GAs. The quality of the initial population significantly influences the performance of the algorithm. To enhance the initialization process, machine learning algorithms such as Logistic Regression and Support Vector Machines (SVM) are utilized to derive the equations that predict the output class on training data. The rationale behind selecting these algorithms is their ability to provide a mathematical representation of the data, which facilitates the generation of a more informed and diverse initial population. The initial population is created using these equations with slight random variations to ensure diversity, as given in Eq. (8).

$$\mathbf{P_0} = \mathrm{S} + \mathcal{N}(0, 0.05) \tag{8}$$

where $\mathbf{P_0}$ represents the initial population, S is the sample of misclassified minority class instances with central probability, and $\mathcal{N}(0, 0.5)$ is a normal distribution with mean 0 and standard deviation 0.05. The standard deviation can be adjusted based on the scope of features in the dataset to adaptively control the variation introduced.

Fitness Function The Fitness Function is a key component of GAs as it determines which samples are identified as suitable to proceed to the next generation. In our dataset, the output class format is binary (0 for the majority class and 1 for the minority class). The aim is to generate a dataset such that the model trained on it

can effectively perform classification on the original test data. For this purpose, emphasis is given to the subset of data that is misclassified by logistic regression. The minimum and maximum probabilities of misclassified samples of class 1, as well as of the minimum and maximum probabilities of the entire class (from only the training dataset), are identified and used in the fitness function. The goal is to generate a synthetic dataset for class 1 with probabilities falling mostly between these minimum and maximum probabilities of the misclassified class (extracted from logistic regression and support vector machine). By doing so, it is ensured that the synthetic samples are representative of the challenging cases, thereby enhancing the model's ability to correctly classify the minority class in the original dataset. The coefficients from the ML algorithms are used to predict the output class for each sample in the initial population. Samples with probabilities that lie within the probability range of a minority class are considered fit samples and are assigned a fitness score of 1. Among these, samples with probabilities within the range of misclassified minority class samples are considered the fittest samples and are given a fitness score of 2. Samples whose probabilities do not fall within the minority class probability range are assigned a fitness score of 0, as given in Eq. (9). By doing this, samples for the minority class are generated, with more emphasis on the misclassified samples.

$$f(\mathrm{x}) = \begin{cases} 2 & \text{if } P_{\min} \leq p \leq P_{\max} \& MP_{\min} \leq p \leq MP_{\max} \\ 1 & \text{if } P_{\min} \leq p \leq P_{\max} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $f(\mathrm{x})$ is the fitness function, $P_{\min}$ and $P_{\max}$ are the minimum and maximum probability of minority class respectively. $MP_{\min}$ and $MP_{\max}$ are the minimum and maximum probabilities of misclassified samples of minority class respectively. These values are extracted from the probabilities output by the machine learning model (Logistic Regression or SVM).

Parent Selection Parent Selection is another critical aspect of GAs, where the fitness scores are used to select parents for the next generation. In the proposed approach, a tournament selection method with a size of 5 is employed, which was found to yield the best results after experimenting with different sizes.. From these 5 individuals, the one with the highest fitness score is selected as a parent, as described in Algorithm 1.

```
procedure SELECTPARENTS(population, fitness_scores)
    parents = []
    for i = 1 to population_size do
        selected = random_sample(population, size=5)
        best_individual = argmax(fitness_scores[selected])
        parents.append(best_individual)
    end for
    return parents
end procedure
```

Algorithm 1. Tournament selection

Crossover and Mutation Crossover and Mutation are genetic operators used to maintain genetic diversity from one generation of a population to the next.

Crossover: Single-point crossover is utilized, where a crossover point is randomly selected, and the genetic material is exchanged between two parents to produce offspring, as given in Eq. (10).

$$\text{offspring} = [\mathbf{p_1}(1 : \text{crossover\_point}), \mathbf{p_2}(\text{crossover\_point} + 1 :)] \tag{10}$$

Mutation: Mutation introduces random variations in the offspring. Each gene in the offspring has a probability of being altered as given in Eq. (11).

$$\text{mutated\_gene} = \text{gene} + \mathcal{N}(0, \sigma) \tag{11}$$

where σ is the mutation rate which is 0.01 in our case. Following the approach discussed above, the resulting Simple Genetic Algorithm is given in Algorithm 2.

```
procedure GeneticAlgorithm(population_size, mutation_rate, generations)
    population = initialize_population()
    for generation = 1 to generations do
        fitness_values = evaluate_fitness(population)
        parents = SelectParents(population, fitness_values)
        offspring = []
        for i = 1 to population_size/2 do
            parent1, parent2 = select_pair(parents)
            child1, child2 = crossover(parent1, parent2)
            mutate(child1), mutate(child2)
            offspring.append(child1), offspring.append(child2)
        end for
        population = offspring
        best_fitness = max(fitness_values)
    end for
    return population
end procedure
```

Algorithm 2. Simple genetic algorithm

Stopping Criteria for GA Iterations The Genetic Algorithm (GA) iterations were stopped after a fixed number of generations (50) across all datasets. This criterion was chosen based on preliminary experiments, which showed that fitness values and synthetic data diversity stabilized before reaching 50 generations in most cases. Fixing the number of generations ensures computational efficiency while maintaining high-quality synthetic data generation.

GA Components and Hyperparameter Specification The hyperparameters for Simple Genetic Algorithm (SGA), Elitist Genetic Algorithm (EGA), and Support Vector Machine-based Genetic Algorithm (SVMGA) were configured to balance computational efficiency and solution quality. The key settings were:

- Population Size: Varied based on the target synthetic data percentage (20%–100% of majority class size)..
- Number of Generations: 50.
- Crossover Probability: 0.5.
- Mutation Probability: 0.01.
- Selection Method: Tournament selection (size: 5).
- Elitism (EGA only): Top 2 individuals retained.
- Initialization: Population initialized near ML Models decision boundaries.

These settings enabled better exploration and convergence, with crossover ensuring diversity and mutation preventing stagnation. EGA's elitism preserved high-quality solutions, while SVMGA's initialization focused the search near critical decision boundaries, enhancing performance across datasets.

Elitist genetic algorithm

Elitism is a strategy that ensures the best individuals (elite) are retained across generations. In the Elitist GA, a specified number of elite individuals (2 in our case) with the highest fitness scores are carried over to the next generation without alteration, as given in Algorithm 3. This mechanism ensures that the best solutions are not lost due to crossover or mutation, contributing to a steady improvement in the overall fitness of the population.

```
procedure ELITISTGA(population_size, mutation_rate, generations, elitism_size)
    population = initialize_population()
    for generation = 1 to generations do
        fitness_values = evaluate_fitness(population)
        elite_indices = argsort(fitness_values)[-elitism_size:]
        elite_individuals = population[elite_indices]
        parents = SelectParents(population, fitness_values, elitism_size)
        offspring = []
        for i = 1 to (population_size - elitism_size)/2 do
            parent1, parent2 = select_pair(parents)
            child1, child2 = crossover(parent1, parent2)
            mutate(child1), mutate(child2)
            offspring.append(child1), offspring.append(child2)
        end for
        population = vstack(offspring, elite_individuals)
        best_fitness = max(fitness_values)
    end for
    return population
end procedure
```

Algorithm 3. Elitist genetic algorithm

Neural network model

After the synthetic dataset is generated using the Genetic Algorithm and merged with the original dataset, a Neural Network is trained to predict the minority class. Initially, a simplified ANN model with only two hidden layers is tested. followed by a more complex model with batch normalization and dropout to stabilizing and speed up the learning process and prevent overfitting. This refined model facilitates the comparative analysis of the effectiveness of the synthetic dataset.

Time complexity analysis

In terms of computational complexity, SMOTE and ADASYN are relatively lightweight, with time complexities of approximately $\mathcal{O}(n \cdot k \cdot d)$, where n is the number of samples, k is the number of neighbors, and d is the dimensionality of the feature space. These methods rely on nearest-neighbor computations and linear interpolation, making them suitable for quick augmentation but limited in generating diverse data. Deep learning approaches like Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) offer more sophisticated data generation capabilities but at significantly higher computational costs, typically $O(E \cdot B \cdot M)$, where E is the number of training epochs, B is the batch size, and M is the model complexity, which can be orders of magnitude larger than traditional methods due to the neural network architectures involved. Additionally, these approaches require substantial training data to achieve stable performance, potentially limiting their application in extreme imbalance scenarios where minority class samples are scarce.

In contrast, the GA-based methods introduce additional overhead due to their iterative nature. For a population size P, number of generations G, and fitness evaluation cost F, the overall complexity is $\mathcal{O}(G \cdot P \cdot F)$. In our implementation, F primarily depends on model-based probability predictions using logistic regression or SVM, which are efficient for low-to-moderate dimensional data. While the GA-based methods are computationally more intensive, they operate on relatively small synthetic batches and converge within a limited number of generations (typically $G \leq 50$), making the runtime practical for offline data preparation. Furthermore, the improved performance and generalization offered by the GA-generated data justify this modest computational cost, especially in critical tasks where minority class detection is vital. These theoretical complexity analyses are further validated by empirical runtime and memory usage measurements presented in Subsection 7.6, which demonstrate the practical tradeoffs between computational efficiency across all methods discussed.

Distribution of synthetic data vs. original data

The comparative analysis of synthetic data distributions through various methodologies reveals distinct patterns that highlight fundamental differences between traditional oversampling techniques and the proposed genetic algorithm-based approaches. To understand how the GA methods (SGA and SVM-based GA) differ from SMOTE, ADASYN, GAN and VAE in replicating the underlying distribution of data, a detailed analysis is performed using using the PIMA Indian Diabetes dataset. The distributions are illustrated using KDE plots, where the y-axis represents the Probability Density Estimate (PDE) and the x-axis represents the features.

The analysis shows that traditional oversampling methods demonstrate a notably conservative approach to synthetic data generation. As shown in Fig. 2, the synthetic data generated by SMOTE is constrained to lie within the convex hull formed by the existing minority class instances and therefore closely resembles the distribution

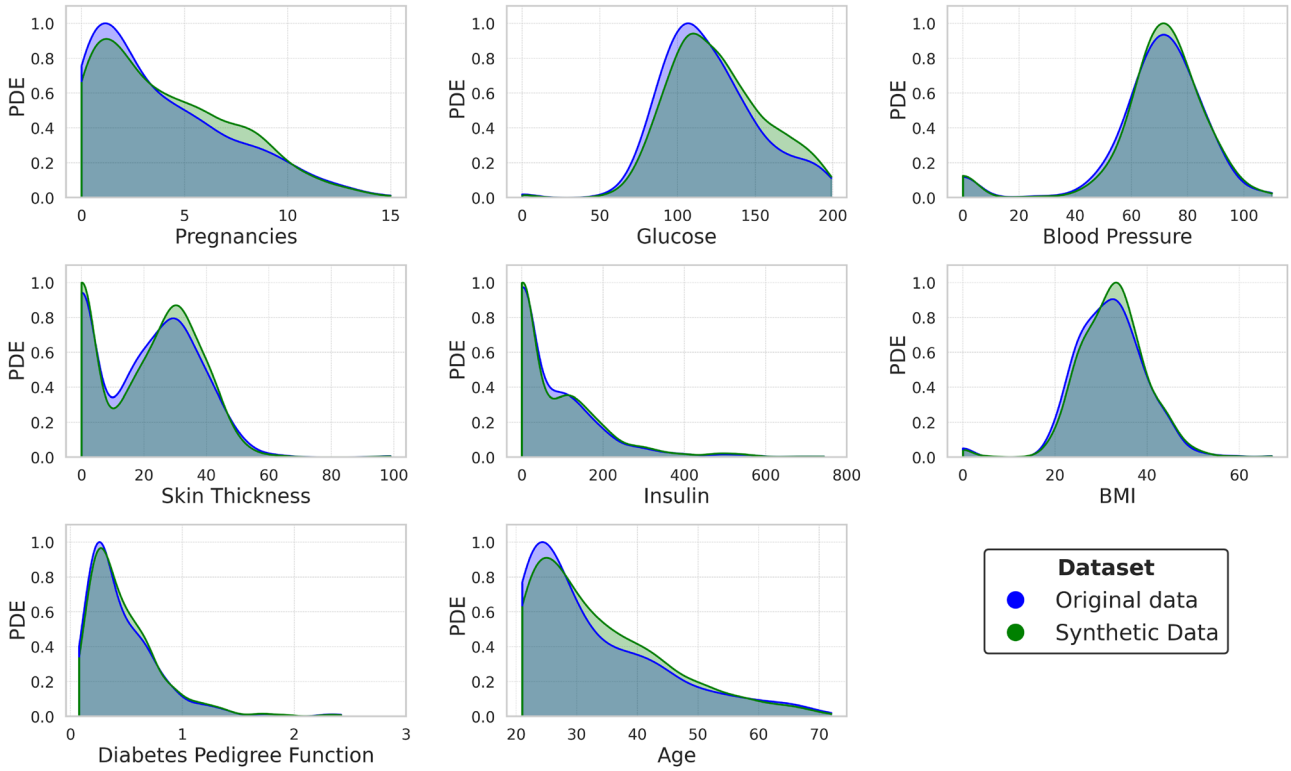

Fig. 2. Comparison of original and synthetic data generated by SMOTE.

of the original dataset, with only slight variations in some features. This high correlation between the original and synthetic data points limits the exploration of new, potentially beneficial regions in the feature space.

The performance of the ADASYN (Adaptive Synthetic Sampling) method is also evaluated on the same datasets. Like SMOTE, ADASYN aims to balance class distribution by creating synthetic samples. It adapts to the local density distribution of the data, using a density-based sampling strategy to generate more synthetic data in regions where the minority class is sparse, effectively focusing on harder-to-learn examples. However, as shown in Fig. 3, despite these adaptations, the ADASYN-generated data still largely overlaps the original data distribution, not exploring beyond the constrained space defined by existing minority class instances.

On the other hand, the SGA-based method introduces a certain variability into the synthetic data while still capturing the broader features of the original data, as observed in the differences between the distributions of the original and synthetic datasets in Fig. 4. Moreover, this method produces noticeably different distribution pattern, characterized by enhanced feature space exploration and evolutionary diversity.

In contrast to these traditional methods, Variational Autoencoders (VAEs) represent a deep learning approach to synthetic data generation. As illustrated in Fig. 5, VAEs encode the original data into a latent space and then decode it to generate new samples. While VAEs can theoretically create diverse synthetic instances by sampling from the learned latent distribution, the analysis reveals that VAE-generated data exhibits distinct characteristics. The distribution shows that VAEs tend to capture the central tendencies of the original data but may smooth out some of the finer details and extremes of the distribution. Similarly, the GAN-based approach introduces a higher degree of variability while retaining essential distribution characteristics, as shown in Fig. 6. By learning the underlying data distribution, GANs generate synthetic samples that approximate the original dataset, albeit with occasional deviations in features such as Skin Thickness and Insulin. This variability reflects GAN's capacity to capture complex data patterns while enabling exploration of diverse feature spaces.

This variability is due to the genetic diversity introduced and propagated by the GA methods during the data generation process. The population is randomly initialized around a starting point, crossover is applied to merge features from different individual samples, and samples are also mutated. Controlled mutation introduces strategic variations in the feature space. Moreover, the fitness function allows for a range of possible solutions that approximately fulfill the target criteria rather than imposing strict uniformity. The combination of these mechanisms across successive generations leads to a diverse set of data points, which differ considerably from the original data points but still retain their most significant attributes. This expanded exploration is particularly significant for discovering new, yet valid, instances of the minority class.

The SVM-based GA method exhibits a more nuanced distribution pattern, as shown in Fig. 7. Notable variations in the synthetic dataset are produced; however, the distribution aligns more closely with the original dataset compared to the SGA method. This balanced behavior can be attributed to three key factors:

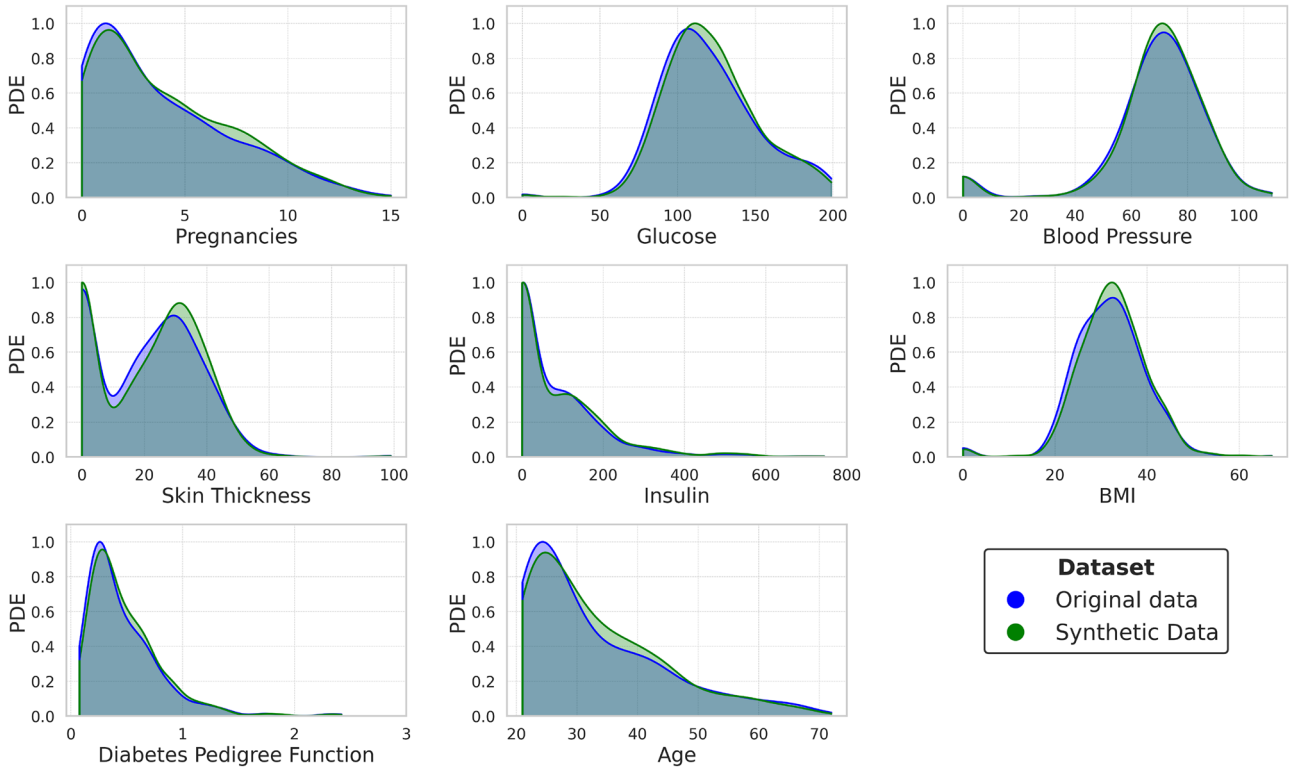

Fig. 3. Comparison of original and synthetic data generated by ADASYN.

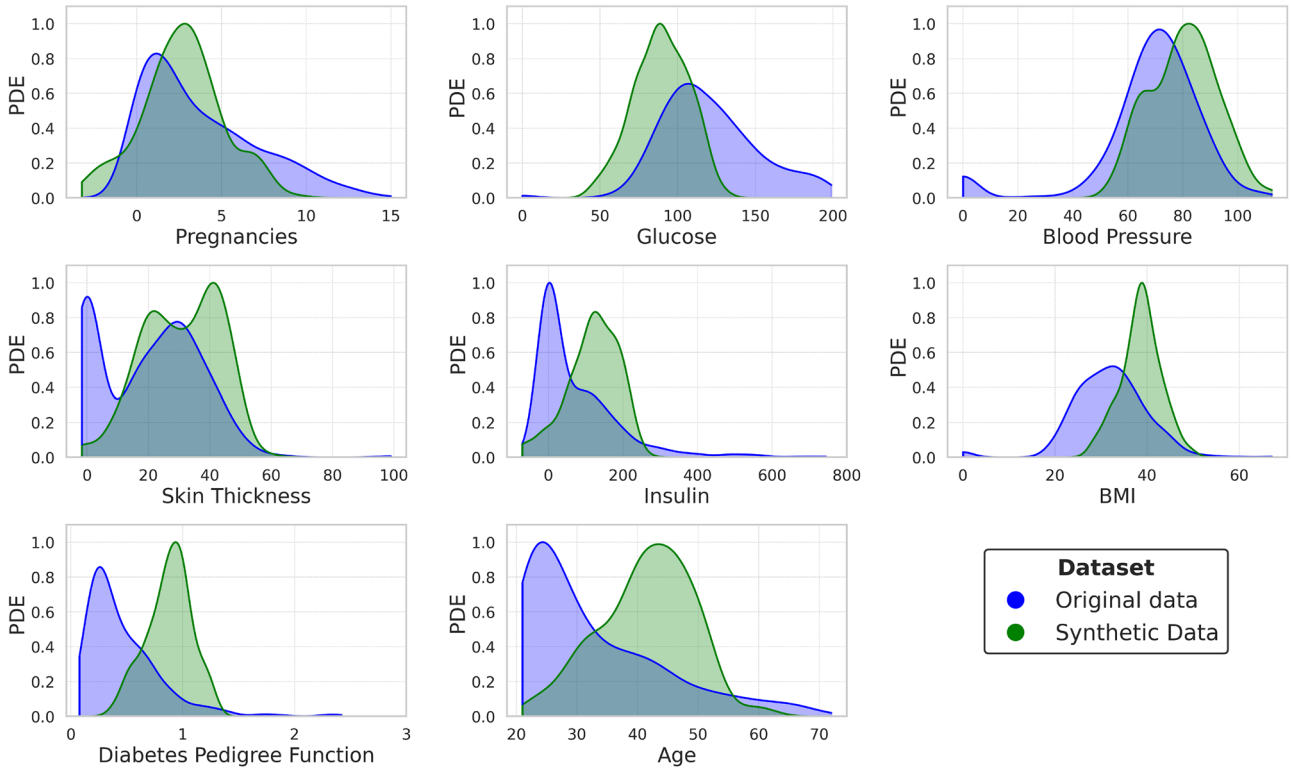

Fig. 4. Comparison of original and synthetic data generated by SGA.

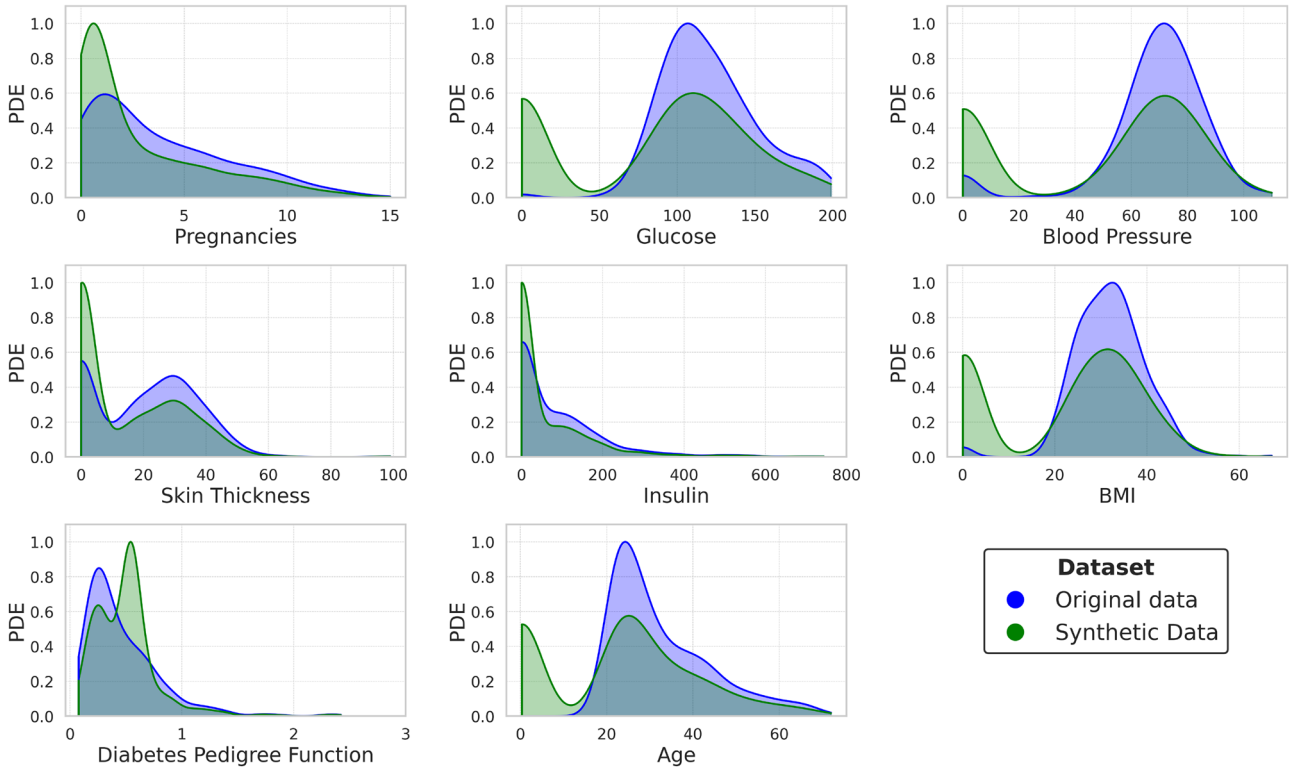

Fig. 5. Comparison of original and synthetic data generated by VAE.

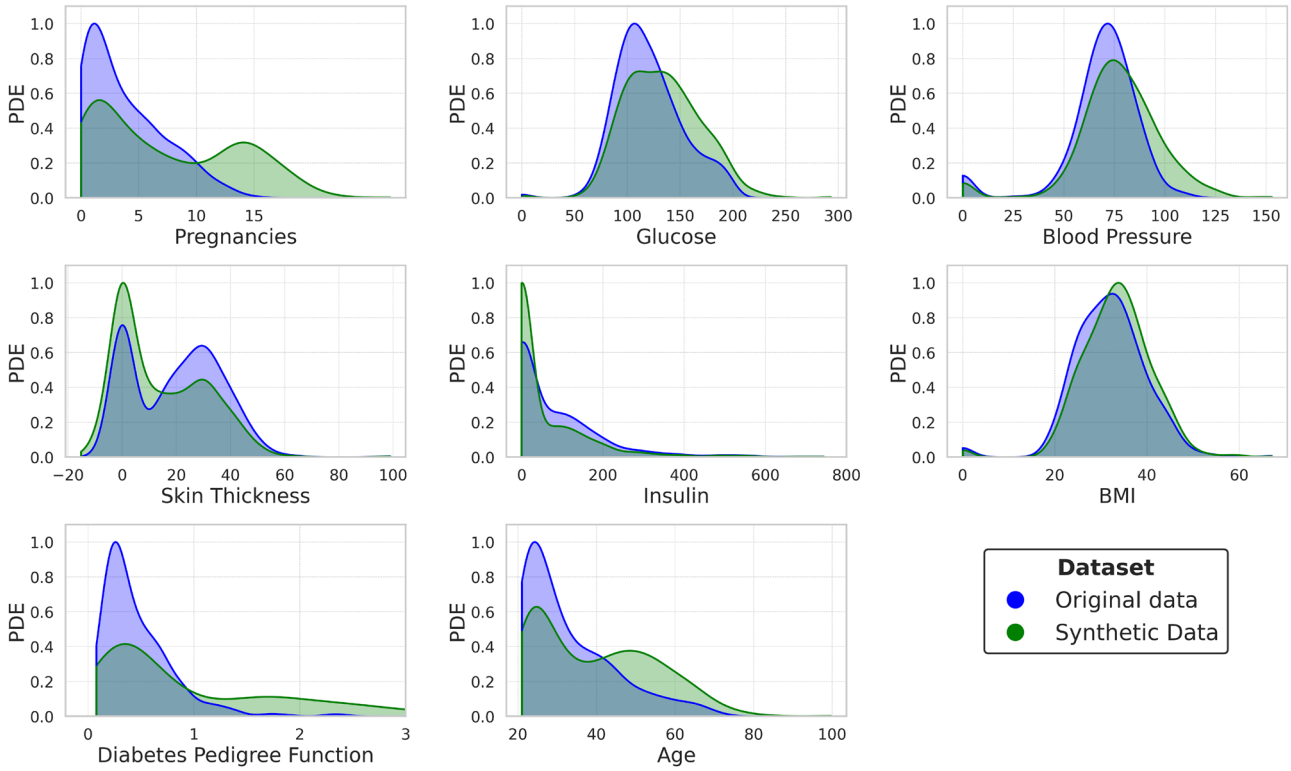

Fig. 6. Comparison of original and synthetic data generated by GAN.

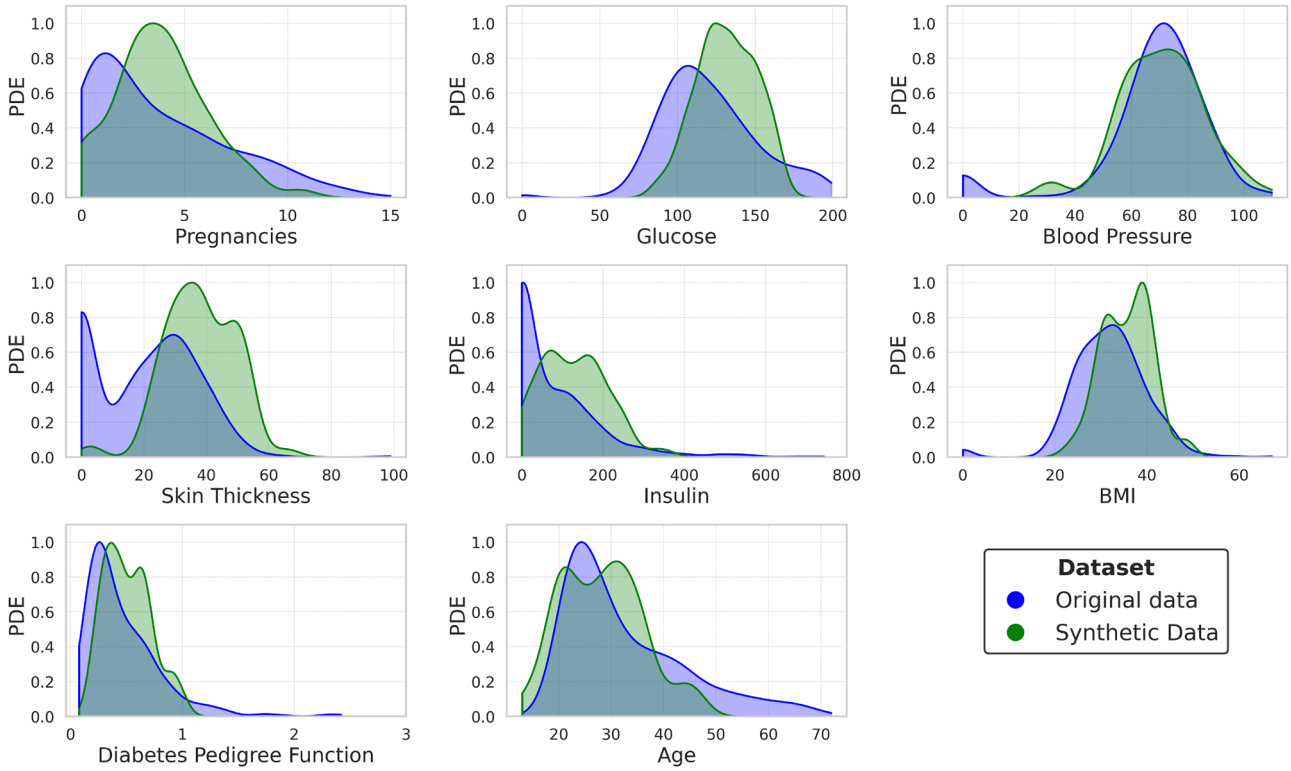

Fig. 7. Comparison of original and synthetic data generated by SVM-based GA.

- The SVM decision boundaries guide synthetic data generation, creating meaningful clusters near critical regions.
- The dual-layered fitness function ensures that synthetic samples maintain proximity to the original distribution while introducing meaningful variations.
- The synthetic data points concentrate in clusters at the decision boundary, where class separation is most crucial.

The theoretical foundations of these distribution patterns stem from the fundamental principles of genetic algorithms. The combination of initialization, crossover mechanisms, and mutation operators leads to synthetic samples that are both diverse and valid within the problem domain. Significant implications for downstream machine learning tasks are associated with this approach, particularly in enhancing model generalization through exposure to a more diverse set of training examples while preventing overfitting that might occur with more closely clustered synthetic data. A balance between maintaining the essential characteristics of the original data and introducing meaningful variations that contribute to improved model performance is thus achieved by the proposed genetic algorithm-based method. This results in the production of a comprehensive training set that helps the model generalize better.

To further examine the empirical differences in data distributions, a t-SNE projection was applied to the original training set of Phoneme together with all seven synthetic datasets as shown in Fig. 8. In this two-dimensional embedding, each point corresponds to a single sample, colored by its source. The original data forms a compact yet clearly structured cluster, reflecting the intrinsic feature correlations present in the raw observations. The t-SNE projection in Fig. 8 presents the two-dimensional embedding of the original training data alongside seven distinct synthetic datasets. Each point represents an individual sample, colored according to its source: the original dataset, GA-generated data (both basic and enhanced implementations), SMOTE and its variant, and samples produced by ADASYN, GAN, and VAE approaches. The original data exhibits a well-defined cluster structure, while the GA-based synthetic samples closely follow its overall topology but introduce additional dispersion along key feature axes. In contrast, SMOTE and ADASYN yield denser, more uniform clusters around the original distribution, and the GAN- and VAE-generated points display broader exploration of the feature space. These patterns confirm that each generation method imparts a unique distributional signature, with GA variants striking an effective balance between fidelity and diversity.

To further support explainability and practical model interpretability, we analyzed how GA-generated synthetic samples influence the decision boundaries of classifiers. Our observations indicate that these samples frequently populate regions close to class boundaries or areas with high misclassification rates. As a result, they encourage more refined and adaptive decision surfaces compared to interpolation-based methods like SMOTE and ADASYN, which often reinforce already well-represented regions. This effect is especially noticeable in SVM-guided GA, where the fitness function inherently targets the most informative zones near decision margins. By

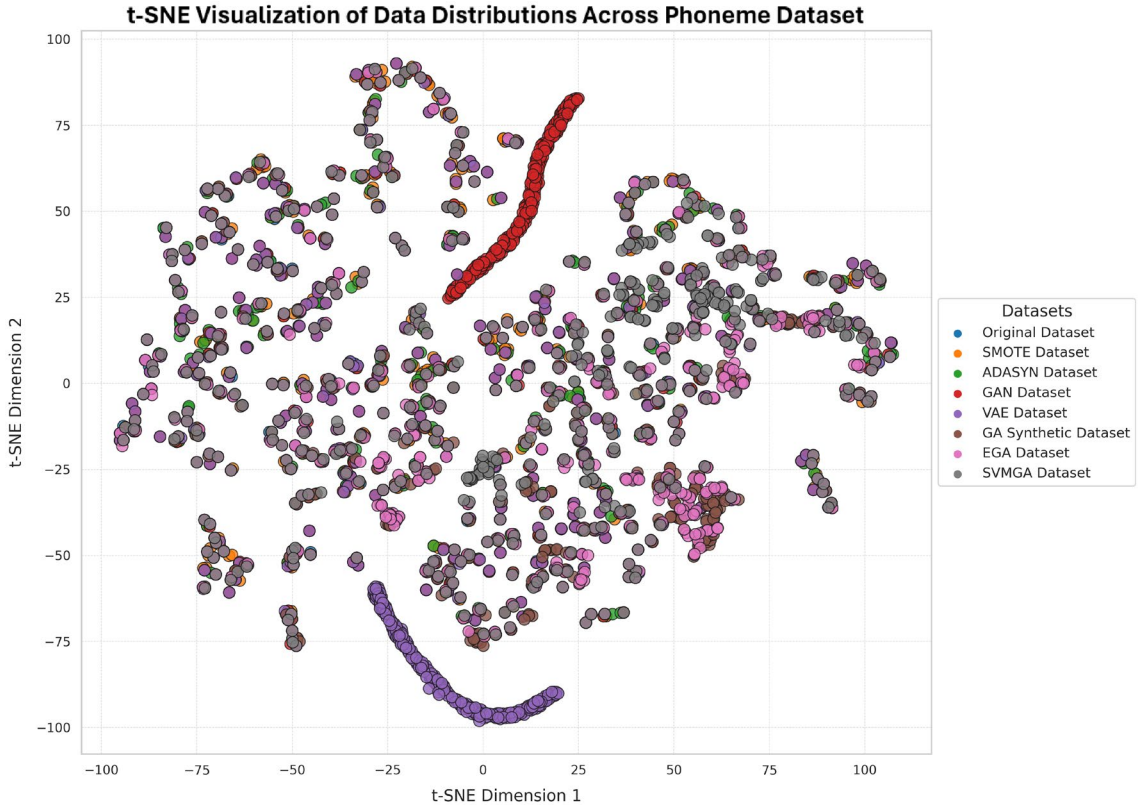

Fig. 8. t-SNE projection of the original training set of and seven synthetic datasets. GA-based samples maintain the core structure of the original cluster while extending its boundaries, SMOTE/ADASYN densely fill the minority-class envelope, and generative models (GAN, VAE) explore broader feature-space regions. This comparative embedding highlights the balance between data fidelity and diversity achieved by each synthetic data generation method.

Method	Average SNR (dB)
SMOTE	27.87
ADASYN	26.83
GAN	20.28
VAE	**29.58**
GA (SGA)	26.29
Elitist GA (EGA)	23.40
SVM-guided GA (SVMGA)	28.44

Table 2. Average signal-to-noise ratio (SNR in dB) for various methods. Significant values are in bold.

placing synthetic instances in these sensitive regions, GA-based approaches improve the classifier's capacity to learn complex separations, thus producing more interpretable and trustworthy models. Consequently, beyond performance gains, the proposed method also enhances the explainability of learned models by influencing decision boundaries in a meaningful and controlled manner.

Quantitative noise analysis

To empirically assess the claim that GA-based methods introduce less harmful noise, we computed the average Signal-to-Noise Ratio (SNR) for synthetic data generated by different methods across all datasets. SNR is a widely used metric to quantify the amount of noise relative to the signal, where higher values indicate cleaner and more consistent data generation.

Table 2 summarizes the average SNR values for each synthetic data generation method.

While SNR values are comparable across most methods–especially between SMOTE, ADASYN, and GA variants–the crucial distinction lies in the nature of the variability introduced. Traditional oversampling methods like SMOTE and ADASYN often produce synthetic samples that closely mimic existing data points, limiting the exploration of new patterns. In contrast, our GA-based methods introduce purposeful and fitness-guided

variability, especially near class boundaries or underrepresented regions in the feature space. This controlled variability does not equate to harmful noise. Instead, it enhances the diversity of the training set in a structured manner, allowing models to better generalize and learn complex decision boundaries. The slightly lower SNR of EGA, for instance, reflects this diversity rather than random noise, as evidenced by its strong classification performance across multiple datasets.

Results

In this study, the dataset is initially split into training and test sets randomly. Synthetic data is generated using the training data, and the model is trained on the combined dataset (original training data and synthetic data). The original test data, which is kept separate from the start, is used for evaluation. This approach aims to evaluate the effectiveness of synthetic data in improving classification performance when applied to real-world data. Specifically, the original dataset serves as a reference to gauge the enhancements brought about by synthetic data augmentation.

In evaluating the performance of our data generation models, a range of performance metrics is utilized, including accuracy, precision, recall, F1 score, and ROC AUC score. While accuracy is a common measure, it alone can be misleading, especially in the presence of class imbalance. Additionally, as demonstrated by another study[68], the Area Under the ROC Curve (AUC) is a valuable metric for assessing the overall discriminative ability of the model across all classification thresholds, offering a more comprehensive evaluation beyond what accuracy alone can provide. As noted by[69], accuracy might mask poor performance on minority classes, leading to an overestimation of a model's effectiveness. Therefore, other metrics are also considered to obtain a comprehensive overview of the model's performance.

In this study, the noise introduced by synthetic data generation methods such as SMOTE and ADASYN is also analyzed, as it can compromise the quality of the generated data and, consequently, the performance of machine learning models.

Moreover, as discussed by[70], the generation of synthetic training data often leads to the production of unwanted noise that can adversely affect the domain adaptation of deep learning models. The noise introduced by synthetic data generation methods such as SMOTE and ADASYN is analyzed and compared with our proposed GA-based methods, which focus on producing noiseless synthetic data.

Credit card fraud detection

Initially, logistic regression is employed to generate an equation for both population initialization and the fitness function. Subsequently, support vector machine (SVM) is applied for the purposes of population initialization and the fitness function. These methods ensure a feasible starting point for the genetic algorithms. Following this, models are trained on both synthetic and original datasets, with both being tested on a consistent subset of the original data. Synthetic data is generated using state-of-the-art methods, SMOTE and ADASYN. Our experiments involve the implementation of Logistic Regression-based Genetic Algorithms (SGA & EGA) and SVM-based Genetic Algorithm. The performance metrics of models trained on synthetic data by our proposed method are compared with those of models trained on the synthetic dataset generated by SMOTE and ADASYN, and the proposed methods show notable improvements, as given in Table 3.

SMOTE Results SMOTE shows moderate initial performance with low precision and F1 scores at low data sampling, but precision and F1 score drop more as more data is sampled, particularly precision falling to 57.2% and F1 score decreasing to 68.2% when 80% of the data is sampled. This indicates that as synthetic data increases, patterns too specific to the data are learned by the model (overfitting), and hence the generalizability of the model is reduced. Despite this, the ROC AUC remains stable around 91.6%-92.3%, and the accuracy in predicting the minority class steadily increases proportional to the increase in data size. SMOTE achieves the highest Recall (85.1) and ROC AUC (91.6), but this leads to the lowest precision (57.1) and F1 score (68.3) after 80% sampling. It shows that the model trained on SMOTE-based generated synthetic data becomes more biased towards class 1, which tends to increase the number of false positives.

ADASYN Results In Table 3, the ADASYN results show a decline in precision and F1 scores as the percentage of synthetic data increases, particularly after 60% or more of the data is sampled. Initially, the method achieves high precision (80.1%) and a balanced F1 score (79.5%) at 20% sampling. However, at 80% data usage, precision drops to 57.2%, and the F1 score decreases to 68.2%, despite a stable high ROC AUC of 92.3%. Similar to SMOTE, this decline suggests that while the ADASYN method is effective in balancing class distribution, noise is introduced as the amount of synthetic data samples increases. This leads to a reduction in precision as the model starts to overfit. This trend is not observed in our proposed GA-based methods.

GAN Results The results of the GAN-based approach, as detailed in Table 3, highlight its potential in generating synthetic data for fraud detection. At lower data usage levels (20%), GAN achieves relatively high precision (83.0%) and balanced F1 scores (82.4%), indicating a good initial performance. However, as the percentage of synthetic data increases, the results suggest diminishing returns. For example, at 80% data usage, precision drops to 61.6%, and the F1 score falls to 74.2%. This decline may indicate that the GAN-generated data introduces noise or fails to generalize effectively as the volume of synthetic data increases. Despite this, GAN maintains a competitive ROC AUC (80.8%-90.1%) across all data splits, showcasing its capability to distinguish between classes. Overall, while GAN demonstrates promise, it faces challenges in maintaining precision and F1 scores as the data volume scales up, suggesting room for improvement in its generative capacity.

VAE Results The results of the VAE-based approach, as presented in Table 3, demonstrate its effectiveness in generating synthetic data for fraud detection. At 20% data usage, VAE achieves respectable precision (82.0%) and recall (80.2%), resulting in a competitive F1 score (81.1%) and strong ROC AUC (90.1%). As the percentage of synthetic data increases, VAE shows better stability compared to traditional methods like SMOTE and ADASYN. For instance, at 80% data usage, VAE maintains a precision of 84.2% and F1 score of 82.6%, notably

%	Metrics	SMOTE[9]	ADASYN[11]	GAN[71]	VAE[63]	SGA	EGA	SVMGA
20%	Accuracy	99.9	99.9	99.9	99.9	99.9	99.9	99.9
	Precision	87.8	80.1	83.0	82.0	**90.5**	90.0	86.7
	Recall	78.3	79.0	81.8	80.2	76.0	79.0	**80.4**
	F1 Score	82.7	79.5	82.4	81.1	82.7	**84.1**	83.4
	ROC AUC	89.1	89.5	90.1	90.1	88.0	89.4	**90.2**
40%	Accuracy	99.9	99.9	99.9	99.9	99.9	99.9	99.9
	Precision	69.2	77.4	86.8	83.8	85.0	**93.1**	84.4
	Recall	**83.3**	81.8	81.1	82.6	82.6	78.9	82.6
	F1 Score	75.6	79.5	83.9	83.2	83.8	**85.5**	83.5
	ROC AUC	**91.6**	90.9	90.5	91.3	91.3	89.5	91.3
60%	**Accuracy**	99.9	99.9	99.9	99.9	99.9	99.9	99.9
	Precision	71.8	72.1	85.7	84.4	85.0	**87.0**	83.8
	Recall	**84.7**	82.6	78.2	82.6	82.6	82.6	83.2
	F1 Score	77.7	77.0	81.8	83.5	83.8	**84.7**	82.9
	ROC AUC	**92.3**	91.2	89.1	91.3	91.3	91.3	**92.3**
80%	Accuracy	99.9	99.9	99.9	99.9	99.9	99.9	99.9
	Precision	57.1	57.2	**93.4**	84.2	86.3	87.6	83.3
	Recall	**85.1**	84.5	61.6	81.1	82.1	82.1	83.3
	F1 Score	68.3	68.2	74.2	82.6	84.1	**84.8**	83.3
	ROC AUC	91.6	**92.3**	80.8	90.5	91.3	91.3	91.6
100%	Accuracy	99.9	99.9	99.9	99.9	99.9	99.9	99.9
	Precision	68.5	73.1	**92.4**	86.4	78.8	78.4	82.6
	Recall	83.1	82.4	71.0	73.9	83.3	**84.0**	82.5
	F1 Score	75.1	77.5	80.3	79.6	81.0	81.1	**82.5**
	ROC AUC	91.9	**92.3**	85.5	86.9	91.6	92.0	92.0

Table 3. Comparison of credit card fraud detection results across different models. This table presents a detailed comparison of performance metrics such as accuracy, precision, recall, F1 score, and ROC AUC achieved by different models (SMOTE, ADASYN, GAN, VAE, SGA, EGA, and SVMGA) on the credit card fraud detection dataset at varying percentages of data usage. Significant values are in bold.

higher than SMOTE's 57.1% precision and 68.3% F1 score at the same threshold. However, VAE experiences a slight performance decline at 100% data usage, with recall dropping to 73.9% and F1 score decreasing to 79.6%. This suggests that while VAE effectively captures the underlying data distribution at moderate synthetic data volumes, it may introduce some generalization challenges when exclusively relying on synthetic samples. Nevertheless, VAE demonstrates more consistent performance across different data sampling percentages compared to SMOTE and ADASYN, indicating its reliability in generating quality synthetic data for imbalanced credit card fraud detection scenarios.

Logistic Regression-Based GA Results The results of the data generated by the Logistic Regression-based GA for the Credit Card Fraud Detection dataset, as demonstrated in Table 3, show a consistent improvement in performance metrics as the percentage of data usage increases. The data generated by SGA follows the trends of improved performance with increasing data. Overall, it achieves greater accuracy in minority class predictions, i.e., 83.3%. Interestingly, the highest precision of 90.5% is observed at 20% data sampling, and the lowest precision of 78.8% is recorded at 100% data sampling. The F1 score remains relatively stable throughout. The ROC AUC increases steadily.

The experimental results obtained from the data generated by EGA also demonstrate a consistent improvement in performance metrics with the increase in available data. It starts with precision and F1 scores better than SGA at 20% data usage, but as data usage increases, EGA shows a more pronounced improvement. For example, at 20% data usage, EGA achieves an F1 score of 84.1% and a ROC AUC of 89.4%, compared to SGA's 82.7% F1 score and 88.0% ROC AUC. This trend continues at higher data levels, with EGA reaching a final F1 score of 81.1% and a ROC AUC of 92.0% at 100% data usage, outperforming SGA at both metrics.

SVM-Based GA Results The SVM-based GA results demonstrate high accuracy and precision across all data splits, with ROC AUC consistently above 90%, indicating the strong discriminative power of this classifier. The accuracy and precision remain high, particularly with increasing data percentages, highlighting the method's effectiveness in correctly identifying the minority class samples. The Support Vector Machine Genetic Algorithm (SVMGA) method achieves the highest Recall (83.3) and highest ROC AUC (92.0), indicating its superior ability to capture patterns of class 1 and differentiate between classes more effectively than other methods.

Metric	SMOTE[9]	ADASYN[11]	GAN[71]	VAE[63]	SGA	EGA	SVMGA
Accuracy	**99.9**	**99.9**	**99.9**	**99.9**	**99.9**	**99.9**	**99.9**
Precision	70.88	71.98	84.9	84.1	85.12	**87.3**	83.92
Recall	**82.90**	82.06	74.7	80.1	81.32	81.32	82.40
F1 score	75.88	76.34	80.5	82.0	83.38	**84.04**	83.12
ROC AUC	91.30	91.24	87.2	90.0	90.70	90.70	**91.48**

Table 4. Average model performance on credit card fraud detection dataset. This table presents the performance metrics (accuracy, precision, recall, F1 score, and ROC AUC) for models trained with different synthetic data generation techniques (SMOTE, ADASYN, GAN, VAE, SGA, EGA, and SVMGA) on the credit card fraud detection dataset. Significant values are in bold.

Models comparison

In this section, the performances of models are compared by taking average values of all the metrics including accuracy, precision, recall, F1 score, and ROC AUC. The results show the dominance of the models using our proposed GA-based synthetic data, as given in Table 4.

From Table 4, it is observed that the accuracy remains consistent across all models. The highest precision is achieved by EGA (87.3), followed by SGA (85.12), GAN (84.9), VAE (84.1), and SVMGA (83.92). A notable difference is found between the precision of the proposed GA-based models and that of SMOTE (70.88) and ADASYN (71.98). All three of the proposed methods, as well as GAN and VAE, show higher precision compared to SMOTE and ADASYN, indicating that the false positive rate in these models is significantly lower. While the highest recall (82.9) is achieved by SMOTE, followed closely by SVMGA (82.40) and both SGA and EGA (81.32), both GAN (74.7) and VAE (80.1) lag behind in this metric. This suggests that while SMOTE is slightly better at identifying all positive instances, the GA-based models still perform more competitively with a more balanced precision-recall trade-off than GAN and VAE. Similarly, the proposed EGA-based model exhibits the highest F1 score (84.04), followed by SGA (83.38), SVMGA (83.12), VAE (82.0), and GAN (80.5). SMOTE (75.88) and ADASYN (76.34) perform poorly in terms of F1 score. Regarding ROC AUC scores, SVMGA achieves the highest value (91.48), followed by SMOTE (91.30) and ADASYN (91.24), while SGA and EGA both score 90.7, VAE achieves 90.0, and GAN shows the lowest performance with 87.2, indicating its limited ability to distinguish between classes effectively.

The Precision-Recall Curve after 100% data sampling, as illustrated in Fig. 9, shows similar trends with slightly varying values. The Average Precision of SGA (0.85) is improved, while SVMGA and ADASYN experience slight decreases, with AP scores of 0.85 and 0.79, respectively. Despite the slight decline, the highest AP score of 0.85 is still achieved by SVMGA, matching the performance of SGA and EGA, which also have an AP score of 0.85. SMOTE matches the AP score of 0.83 with the Original data.

The ROC AUC is plotted after 100% data sampling, which provides a complete picture of how well the model can separate the two classes, considering all possible thresholds, as given in Fig. 9. The highest ROC AUC score of 0.97 is achieved by the SVMGA and EGA methods, indicating that models trained on data generated by these methods can effectively distinguish between the classes. The SGA method attains a slightly lower ROC AUC score of 0.96, while both SMOTE and ADASYN match the original data's ROC AUC score of 0.95. The proposed GA-based methods are found to outperform the SMOTE and ADASYN methods by a significant margin.

Statistical Significance Testing of F1 Scores To determine whether the observed differences in model performance are statistically significant, pairwise t-tests were conducted on the average F1 scores of all evaluated models. Table 5 presents the corresponding p values and mean differences for each comparison. The results indicate that the proposed GA-based methods–particularly EGA and SGA–frequently achieve p-values below the conventional threshold of 0.05 when compared to traditional oversampling techniques such as SMOTE and ADASYN. For example, EGA demonstrates statistically significant improvements over SMOTE ($p = 0.0307$) and ADASYN ($p = 0.0298$). Similarly, SGA exhibits statistically significant differences when compared to SMOTE ($p = 0.0475$) and ADASYN ($p = 0.0470$).

In contrast, comparisons between GA-based approaches and generative models such as VAE and GAN reveal fewer statistically significant differences. Although VAE achieves competitive F1 scores, the difference between EGA and VAE is statistically significant ($p = 0.0030$), as is the comparison between SGA and VAE ($p = 0.0149$). However, GAN does not yield statistically significant differences in most pairwise comparisons, suggesting its performance is not consistently distinguishable from the other methods.

PIMA Indian diabetes

The results with the PIMA Indian Diabetes dataset show notable differences in the performance of the EGA, SGA, and SVMGA methods. Overall, it is observed that SVMGA outperforms all other methods, indicating that more reliable and discriminative synthetic data is generated by SVMGA, as given in Table 6.

SMOTE Results The model trained on the synthetic data generated by the SMOTE method shows a considerable improvement in recall and F1 Score as the percentage of sampled synthetic data increases up to 40%, with the highest value of precision being observed at this point compared to earlier stages, as given in Table 6. However, at higher percentages (80% and 100%), precision and F1 Score begin to decline, particularly at 100%, where both metrics drop significantly. This demonstrates that the ability of the model to accurately predict class labels decreases with the increase in generated synthetic data samples.

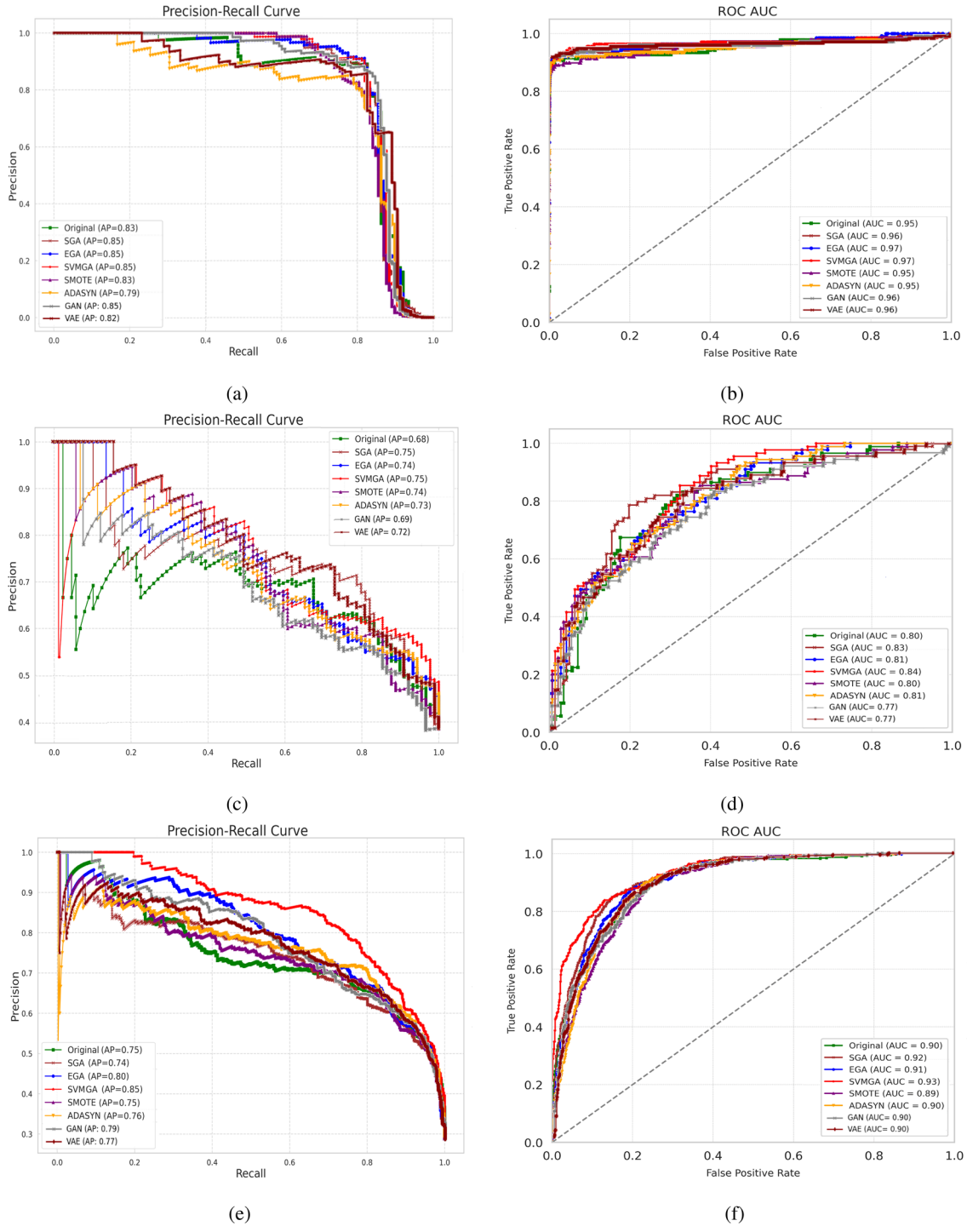

Fig. 9. Resultant Images of all three datasets with 100% data sampling where (**a**) represents the PR-Curve for the Credit Card Fraud Detection dataset, (**b**) represents the ROC-AUC for the Credit Card Fraud Detection dataset, (**c**) represents the PR-Curve for the PIMA dataset, (**d**) represents the ROC-AUC for the PIMA dataset, (**e**) represents the PR-Curve for the Phoneme dataset, and (**f**) represents the ROC-AUC for the Phoneme dataset.

ADASYN Results The results of the ADASYN method on the PIMA dataset, as given in Table 6, show that while balanced performance across different metrics is maintained, the overall effectiveness slightly decreases when using the full dataset (100%), with lower precision (55.2%) and ROC AUC (71.4%) compared to earlier stages. This suggests that noise may be introduced by ADASYN when generating synthetic data, which can affect the model's precision and generalization as more data is utilized. The method demonstrates optimal performance with the highest ROC AUC (73.4%) after 40% sampling on the dataset.

Model	SMOTE[9]	ADASYN[11]	GAN[71]	VAE[63]	SGA	EGA	SVMGA
SMOTE	75.88	− 0.46 (0.7291)	− 4.64 (0.0305)	− 6.12 (0.0754)	− 7.20 (0.0475)	− 8.16 (0.0307)	-7.24 (0.0355)
ADASYN	0.46 (0.7291)	76.34	− 4.18 (0.0023)	− 5.66 (0.0734)	− 6.74 (0.0470)	− 7.70 (0.0298)	− 6.78 (0.0326)
GAN	4.64 (0.0305)	4.18 (0.0023)	80.52	− 1.48 (0.4583)	− 2.56 (0.2425)	− 3.52 (0.1224)	− 2.60 (0.1959)
VAE	6.12 (0.0754)	5.66 (0.0734)	1.48 (0.4583)	82.00	− 1.08 (0.0149)	− 2.04 (0.0030)	− 1.12 (0.1584)
SGA	7.20 (0.0475)	6.74 (0.0470)	2.56 (0.2425)	1.08 (0.0149)	83.08	− 0.96 (0.0261)	− 0.04 (0.9352)
EGA	8.16 (0.0307)	7.70 (0.0298)	3.52 (0.1224)	2.04 (0.0030)	0.96 (0.0261)	84.04	0.92 (0.2125)
SVMGA	7.24 (0.0355)	6.78 (0.0326)	2.60 (0.1959)	1.12 (0.1584)	0.04 (0.9352)	− 0.92 (0.2125)	83.12

Table 5. Pairwise comparison of mean F1-scores across various different models on credit card fraud detection dataset. Diagonal entries show the average F1-score for each model. Off-diagonal entries represent the mean difference in F1-score between the row and column models, followed by the p-value from a paired t-test in parentheses. Significant values are in bold.

%	Metrics	SMOTE[9]	ADASYN[11]	GAN[71]	VAE[63]	SGA	EGA	SVMGA
20%	Accuracy	71.1	75.0	76.6	73.1	72.1	74.3	**78.2**
	Precision	61.0	67.9	**74.6**	58.8	64.5	65.9	73.6
	Recall	62.0	61.8	59.5	67.4	**69.6**	65.1	62.9
	F1 Score	62.2	64.7	66.2	65.9	65.6	65.5	**67.8**
	ROC AUC	69.1	71.8	73.4	72.0	71.8	72.0	**74.4**
40%	Accuracy	75.0	75.4	67.0	75.5	73.0	**75.6**	75.4
	Precision	65.6	67.4	54.9	**68.9**	65.3	67.0	67.0
	Recall	73.0	67.4	**89.9**	67.4	69.6	70.7	68.5
	F1 Score	**69.1**	67.4	65.5	68.1	67.4	68.9	67.8
	ROC AUC	**74.5**	73.4	69.7	74.2	73.2	**74.5**	73.7
60%	Accuracy	73.2	74.0	71.4	72.7	76.9	75.1	**78.9**
	Precision	64.1	65.2	59.3	65.1	73.5	67.0	**73.7**
	Recall	66.3	67.4	**82.0**	62.9	56.0	66.3	66.3
	F1 Score	65.1	66.3	68.8	64.0	64.0	66.6	**69.8**
	ROC AUC	71.5	72.4	73.4	70.9	71.8	72.9	**75.7**
80%	Accuracy	70.5	67.6	72.2	74.0	60.6	69.0	**76.1**
	Precision	**65.0**	56.7	60.0	62.8	57.0	60.0	60.8
	Recall	71.0	85.0	84.2	79.7	82.0	78.0	**88.3**
	F1 Score	67.9	68.0	70.0	70.3	67.3	67.8	**72.0**
	ROC AUC	70.0	72.0	74.5	75.1	67.0	72.0	**75.7**
100%	Accuracy	68.8	69.0	69.2	76.2	71.3	73.6	**76.4**
	Precision	59.1	55.2	58.3	67.7	**70.8**	60.6	59.0
	Recall	72.6	83.4	70.7	73.0	76.0	74.0	**91.4**
	F1 Score	65.2	66.4	63.9	70.2	73.3	66.6	**71.5**
	ROC AUC	71.0	71.4	69.5	75.6	72.0	74.0	**76.3**

Table 6. Comparison of PIMA Indian results across different models. This table shows the performance metrics (accuracy, precision, recall, F1 score, and ROC AUC) for models trained with different synthetic data generation techniques (SMOTE, ADASYN, GAN, VAE, SGA, EGA, and SVMGA) on the PIMA Indian diabetes dataset at varying percentages of data usage. Significant values are in bold.

GAN Results The GAN-based approach on the PIMA Indian Diabetes dataset, as shown in Table 6, demonstrates inconsistent performance across sampling percentages. At 20%, it achieves moderate precision (74.6%) but low recall (59.5%), missing many positive cases. As sampling increases to 40%, performance significantly deteriorates with precision dropping to 54.9% and accuracy decreasing to 67.0%–the lowest among all methods at this level. Although recall improves at 40% and 60% sampling (89.9% and 82.0%), precision suffers, indicating issues with generating balanced synthetic data. This instability suggests GAN fails to accurately capture the underlying data distribution of the PIMA dataset. By 100% data sampling, GAN's F1 Score (63.9%) is among the lowest of all methods, demonstrating its limitations in generating quality synthetic data for this classification task.

VAE Results The VAE method on the PIMA Indian Diabetes dataset, as presented in Table 6, shows several shortcomings across different sampling percentages. Starting with a suboptimal 58.8% precision at 20% sampling–among the lowest of all methods–VAE struggles to generate discriminative synthetic samples. While precision improves to 68.9% at 40% sampling, the recall remains stagnant at 67.4%, indicating a persistent

problem with identifying positive cases. As sampling increases to 60%, VAE shows a regression in performance with recall dropping to 62.9%, demonstrating an inability to maintain consistent improvement as more synthetic data is incorporated. Although VAE reaches 76.2% accuracy at 100% sampling, its overall metrics still fall short compared to GA-based methods, particularly in recall and F1 Score when compared to SVMGA. These limitations highlight VAE's deficiency in generating optimal synthetic data for imbalanced classification problems like the PIMA dataset.

Logistic Regression-based GA Results The results from the experiments on the PIMA Indian Diabetes dataset show notable differences in the performance of the EGA and the baseline (SGA) methods across various metrics, as given in Table 6. Overall, it is observed that SGA outperforms the EGA methods, particularly in terms of precision and F1 Score, indicating that more reliable and discriminative synthetic data is generated by SGA. When focusing on experimental results obtained after 100% data sampling, the advantages of SGA become more apparent. It achieves better precision (70.8) and F1 Score (73.3) compared to EGA, which reports precision values of 60.6 and F1 Scores of 66.6 after 100% data sampling. The ROC AUC for EGA at 100% data sampling is superior (74.0) compared to SGA (72.0), suggesting a stronger overall performance in distinguishing between classes.

SVM-based GA Results The results from the SVM-based GA, as given in Table 6, demonstrate consistent improvement in performance metrics as the percentage of data used increases. Notably, the highest performance is achieved with 100% sampling of the data, where strong recall (91.4%), F1 Score (71.5%), and ROC AUC (76.3%) are shown, indicating that the SVM-based GA is effective in enhancing the model's ability to identify true positives and maintain a balanced performance across various metrics as more data is leveraged. Initially, after 20% sampling, SVMGA achieves better scores across accuracy (78.2), precision (73.6), F1 Score (67.8), and ROC AUC (74.4), indicating the effectiveness of the data generated by SVMGA. All other methods perform similarly across all metrics but remain lower than SVMGA, with SGA achieving the highest recall of 69.6.

Models comparison

As given in Table 7, it is observed that the SVM-based method exhibits the best performance in terms of accuracy (77.0), precision (66.82), F1 Score (69.78), and ROC AUC (76.16) while GAN shows highest recall (77.26) followed by SVM-based method (75.48). This indicates that the SVM-based GA method generates data with considerable variability while effectively capturing the critical decision boundaries of the original data.

The Precision-Recall curve after 100% sampling on the data, illustrated in Fig. 9, shows a considerable improvement in the performance of the SGA (0.75) and EGA (0.74) methods, while the SVM-based GA (0.75) method maintains its highest position. This is attributed to the availability of the entire dataset, which introduces more diversity, inherently suitable for the GA-based methods. The performance of the ADASYN method worsens slightly, while the rest of the methods do not show a pronounced change in their outputs.

Figure 9 shows the ROC (Receiver Operating Characteristic) curve for the original and synthetic data and is used to evaluate the trade-off between the true positive rate (sensitivity) and the false positive rate (1-specificity) for different threshold values, with the Area Under the Curve (AUC) providing an overall measure of model performance. The SVM-based Genetic Algorithm (AUC = 0.84) outperforms all other methods, indicating that it is the most effective in improving model performance by maximizing the true positive rate while minimizing the false positive rate. SMOTE-based methods (AUC = 0.80) perform comparably to the original data (AUC = 0.80). EGA (AUC = 0.81) and ADASYN (AUC = 0.81) slightly outperform the original data-trained model but do not match the performance of the SVM-based GA. SGA (AUC = 0.83) shows better performance than EGA and ADASYN, though it still lags behind the SVM-based GA. Overall, the SVM-based GA and SGA-based approaches demonstrate superior effectiveness in handling class imbalance, as reflected by their higher AUC scores.

Statistical Comparison of F1 Scores on the PIMA Dataset Table 8 presents pairwise comparisons of mean F1-scores for models evaluated on the PIMA dataset. Diagonal entries show the average F1-score per model, while off-diagonal values indicate the mean difference and corresponding p-values from paired t-tests. SVMGA achieved the highest mean F1-score (69.78) and showed statistically significant improvements over SMOTE ($p = 0.0451$) and ADASYN ($p = 0.0146$). While comparisons with other models such as GAN and VAE yielded favorable differences, the p-values suggest marginal significance. Other GA-based methods (SGA and EGA) also demonstrated competitive performance, though their advantages were not statistically significant. These results indicate that GA-based approaches, particularly SVMGA, offer superior classification performance on the PIMA dataset compared to conventional resampling and generative methods.

Metric	SMOTE[9]	ADASYN[11]	GAN[71]	VAE[63]	SGA	EGA	SVMGA
Accuracy	71.72	72.2	71.3	74.3	70.8	73.52	**77.0**
Precision	62.96	62.48	61.42	64.7	65.84	64.1	**66.82**
Recall	68.96	73.0	**77.26**	70.08	70.64	70.82	75.48
F1 Score	65.9	66.56	66.9	67.7	67.53	67.08	**69.78**
ROC AUC	71.22	72.2	72.1	73.5	71.16	73.08	**75.16**

Table 7. Average model performance on PIMA dataset. This table presents the average performance metrics (accuracy, precision, recall, F1 score, and ROC AUC) for models trained with different synthetic data generation techniques (SMOTE, ADASYN, GAN, VAE, SGA, EGA, and SVMGA) on the PIMA Indian Diabetes dataset. Significant values are in bold.

Model	SMOTE[9]	ADASYN[11]	GAN[71]	VAE[63]	SGA	EGA	SVMGA
SMOTE	65.90	− 0.66 (0.4003)	− 0.98 (0.5447)	− 1.80 (0.2184)	− 1.62 (0.4308)	− 1.18 (0.1389)	− 3.88 (0.0451)
ADASYN	0.66 (0.4003)	66.56	− 0.32 (0.7748)	− 1.14 (0.3226)	− 0.96 (0.5750)	− 0.52 (0.1498)	− 3.22 (0.0146)
GAN	0.98 (0.5447)	0.32 (0.7748)	66.88	− 0.82 (0.6758)	− 0.64 (0.8073)	− 0.20 (0.8758)	− 2.90 (0.0722)
VAE	1.80 (0.2184)	1.14 (0.3226)	0.82 (0.6758)	67.70	0.18 (0.8626)	0.62 (0.6077)	− 2.08 (0.1078)
SGA	1.62 (0.4308)	0.96 (0.5750)	0.64 (0.8073)	− 0.18 (0.8626)	67.52	0.44 (0.8007)	− 2.26 (0.1784)
EGA	1.18 (0.1389)	0.52 (0.1498)	0.20 (0.8758)	− 0.62 (0.6077)	− 0.44 (0.8007)	67.08	− 2.70 (0.0615)
SVMGA	3.88 (0.0451)	3.22 (0.0146)	2.90 (0.0722)	2.08 (0.1078)	2.26 (0.1784)	2.70 (0.0615)	69.78

Table 8. Pairwise comparison of mean F1-scores across various different models on PIMA Dataset. Diagonal entries show the average F1-score for each model. Off-diagonal entries represent the mean difference in F1-score between the row and column models, followed by the p-value from a paired t-test in parentheses. Significant values are in bold.

%	Metrics	SMOTE[9]	ADASYN[11]	GAN[71]	VAE[63]	SGA	EGA	SVMGA
20%	Accuracy	**83.7**	82.7	81.0	81.6	81.5	82.8	83.2
	Precision	**68.9**	66.4	62.2	64.2	62.9	66.7	68.1
	Recall	79.4	80.7	86.9	81.8	**87.1**	80.9	79.2
	F1 Score	**73.8**	72.9	72.5	71.9	73.0	73.1	73.2
	ROC AUC	82.4	82.1	**82.8**	81.7	83.1	82.3	**82.4**
40%	Accuracy	**82.6**	81.7	79.9	80.9	81.6	82.0	**82.6**
	Precision	**65.9**	63.5	60.8	62.1	63.1	64.3	64.7
	Recall	82.8	85.2	85.2	**86.7**	**86.7**	84.1	85.8
	F1 Score	73.4	72.8	71.0	72.3	73.1	72.9	**73.8**
	ROC AUC	82.7	82.7	81.5	82.6	83.1	82.6	**83.5**
60%	Accuracy	**83.9**	78.3	80.2	82.6	80.6	81.8	82.4
	Precision	**68.0**	57.9	60.5	66.2	63.7	64.3	64.7
	Recall	83.7	**89.7**	89.5	81.3	83.0	83.0	88.0
	F1 Score	**75.0**	70.4	72.2	73.0	72.1	72.5	74.6
	ROC AUC	83.8	81.7	82.9	82.2	83.0	82.2	**84.3**
80%	Accuracy	79.9	78.6	80.2	80.2	79.7	80.8	**80.9**
	Precision	60.2	52.3	60.5	61.2	58.4	**61.6**	59.1
	Recall	89.3	**94.2**	89.9	85.8	88.8	87.5	90.7
	F1 Score	71.9	67.3	**72.3**	71.5	70.5	**72.3**	71.6
	ROC AUC	82.6	80.1	83.1	81.9	82.5	82.7	**83.8**
100%	Accuracy	69.8	77.2	79.5	78.7	77.5	79.3	**79.6**
	Precision	52.5	55.8	59.7	58.5	58.9	58.5	**60.8**
	Recall	**94.2**	92.7	88.2	89.7	87.1	87.2	90.5
	F1 Score	67.4	69.7	71.2	70.8	70.3	70.0	**72.7**
	ROC AUC	76.9	81.7	82.1	82.0	80.3	82.3	**82.7**

Table 9. Comparison of PHONEME Indian results across different models. This table shows the performance metrics (accuracy, precision, recall, F1 score, and ROC AUC) for models trained with different synthetic data generation techniques (SMOTE, ADASYN, GAN, VAE, SGA, EGA, and SVMGA) on the PHONEME dataset at varying percentages of data usage. Significant values are in bold.

Phoneme

The results from the experiments on the Phoneme dataset show notable improvements in performance across various metrics for several methods, particularly SVMGA. Overall, SVMGA is observed to outperform the other methods in most scenarios, indicating that more reliable and discriminative synthetic data is generated, as shown in Table 9.

SMOTE Results The model trained on the synthetic data generated by the SMOTE method shows a considerable improvement in recall and F1 Score, as presented in Table 9, as the percentage of sampled synthetic data increases up to 60%, with the highest precision being observed at this point compared to earlier stages. However, at higher percentages (80% and 100%), precision and F1 Score start to decline, particularly at 100%, where both metrics drop significantly. This indicates that the model's ability to accurately predict class labels decreases as the volume of generated synthetic data increases. Compared to SGA and EGA, SMOTE provides

better initial improvements in recall and F1 score; however, its performance declines at higher synthetic data percentages, suggesting sensitivity to the noise introduced by the larger volume of synthetic data.

ADASYN Results Similar to the SMOTE results, the results obtained from the ADASYN method show a gradual improvement in the accuracy of minority class predictions, but the performance of the model peaks at 40% data sampling. A notable decline in precision and F1 score is observed after more than 40% of the data is sampled. The ROC AUC remains relatively stable, with a slight improvement observed at 40%.

GAN Results The GAN-based approach on the PHONEME dataset shows mixed results as per Table 9. GAN achieves strong recall values (86.9%-89.9%) across all sampling levels and ties for the highest F1 Score (72.3%) at 80% sampling with EGA. It also shows competitive ROC AUC (82.8%) at 20% sampling. However, GAN consistently underperforms in precision, recording the lowest values at multiple sampling levels. Its accuracy remains below most other methods, particularly at 40% sampling (79.9%). These limitations suggest GAN generates synthetic samples that introduce classification errors, preventing it from achieving the balanced performance of GA-based methods, especially SVMGA, which demonstrates superior precision-recall trade-offs across sampling percentages.

VAE Results The VAE method on the PHONEME dataset demonstrates moderate strengths per Table 9. At 40% sampling, VAE achieves the highest recall (86.7%, tied with SGA), while at 60% sampling, it shows good accuracy (82.6%) and the second-highest precision (66.2%). VAE maintains relatively stable performance across metrics as sampling increases. However, VAE struggles to balance precision and recall simultaneously–when recall improves, precision often suffers. At 100% sampling, VAE's precision drops to 58.5%, significantly lower than SVMGA's 60.8%. While VAE achieves reasonable ROC AUC values throughout, it consistently falls short of the more balanced performance demonstrated by GA-based methods, particularly SVMGA, which better handles the precision-recall trade-off across different sampling levels.

Logistic Regression-based GAs Results When using SGA, accuracy scores remain relatively consistent across different sampling percentages, but precision and recall slightly decrease as the percentage increases, leading to a slight drop in F1 score and ROC AUC values. This suggests that while overall accuracy is maintained by SGA, it slightly struggles with precision and recall, particularly as more synthetic data is introduced. Interestingly, the highest accuracy for minority class predictions is observed at 80% data sampling, where the model appears to strike a balance between meaningful diversity and minimal noise.

The EGA results for the Phoneme dataset demonstrate relatively stable accuracy, with precision and recall showing modest variability as the percentage of synthetic data increases. The F1 score shows a slight decline at higher synthetic data percentages, reflecting a minor degradation in model performance, particularly in handling the trade-offs between precision and recall. Compared to the SGA results, EGA exhibits similar performance on precision and recall, though EGA achieves slightly higher accuracy than the SGA-based model.

SVM-based GA Result The results of the experiments conducted on synthetic data obtained from the SVM-based GA show a consistent improvement in the accuracy of minority class predictions, except for a slight dip at 100% data sampling. However, a slight drop in precision and F1 score is observed at higher synthetic data percentages, particularly at 80%, where both metrics display their lowest values. The ROC AUC remains relatively stable across different data percentages. Optimal model performance is observed at 20%-60% data sampling, with high accuracy, recall, and F1 score.

Models comparison

As observed from Table 10, the models perform comparably across all performance metrics, including accuracy, precision, recall, F1 score, and ROC AUC. The overall precision achieved ranges from 59 to 63%; however, the proposed SVMGA-based method outperforms the GAN, VAE, SMOTE and ADASYN methods. The most optimal performance appears to be achieved by the SVM-based GA method, with relatively high and stable values across all metrics. SVM-based GA demonstrates the highest F1 score (73.18) and ROC AUC (83.34), indicating a balance between precision and recall. While GAN shows highest recall (87.9), representing the percentage of class 1 samples correctly classified, ADASYN exhibits the lowest precision (59.18), accuracy (79.7), F1 score (70.0), and ROC AUC (80.0), indicating poor performance across all performance metrics.

The Precision-Recall Curve after 100% data sampling does not produce much variation in the outcome. The models are ranked in the same manner, with the SVM-based GA method leading the others, as observed in Fig. 9.

The ROC curve, illustrated in Fig. 9 indicates that the SVM-based GA method is the best performer, with an AUC of 0.93. This is followed by SGA and EGA, with AUCs of 0.92 and 0.91, respectively. The other models

Metric	SMOTE[9]	ADASYN[11]	GAN[71]	VAE[63]	SGA	EGA	SVMGA
Accuracy	79.98	79.7	80.1	80.8	80.18	81.34	**81.74**
Precision	63.1	59.18	60.8	62.4	61.4.	63.08	**63.48**
Recall	85.88	87.0	**87.9**	85.1	86.5	84.54	86.84
F1 Score	72.3	70.0	71.8	71.9	71.8	72.1	**73.18**
ROC AUC	81.68	80.0	82.5	82.1	82.4	82.41	**83.34**

Table 10. Average model performance on phoneme dataset. This table presents the average performance metrics (accuracy, precision, recall, F1 score, and ROC AUC) for models trained with different synthetic data generation techniques (SMOTE, ADASYN, GAN, VAE, SGA, EGA, and SVMGA) on the Phoneme dataset. Significant values are in bold.

Model	SMOTE[9]	ADASYN[11]	GAN[71]	VAE[63]	SGA	EGA	SVMGA
SMOTE	72.30	+ 1.68 (0.2710)	+ 0.46 (0.7211)	+ 0.40 (0.7078)	+ 0.50 (0.6284)	+ 0.14 (0.8744)	− 0.88 (0.4751)
ADASYN	− 1.68 (0.2710)	70.62	− 1.22 (0.3485)	− 1.28 (0.2558)	− 1.18 (0.1097)	− 1.54 (0.1769)	− 2.56 (0.0355)
GAN	− 0.46 (0.7211)	+ 1.22 (0.3485)	71.84	− 0.06 (0.8925)	+ 0.04 (0.9546)	− 0.32 (0.5565)	− 1.34 (0.0992)
VAE	− 0.40 (0.7078)	+ 1.28 (0.2558)	+ 0.06 (0.8925)	71.90	+ 0.10 (0.8311)	− 0.26 (0.5383)	− 1.28 (0.0146)
SGA	− 0.50 (0.6284)	+ 1.18 (0.1097)	− 0.04 (0.9546)	− 0.10 (0.8311)	71.80	− 0.36 (0.3974)	− 1.38 (0.0399)
EGA	− 0.14 (0.8744)	+ 1.54 (0.1769)	+ 0.32 (0.5565)	+ 0.26 (0.5383)	+ 0.36 (0.3974)	72.16	− 1.02 (0.1779)
SVMGA	+ 0.88 (0.4751)	+ 2.56 (0.0355)	+ 1.34 (0.0992)	+ 1.28 (0.0146)	+ 1.38 (0.0399)	+ 1.02 (0.1779)	73.18

Table 11. Pairwise comparison of mean F1-scores across various different models on PHONEME Dataset. Diagonal entries show the average F1-score for each model. Off-diagonal entries represent the mean difference in F1-score between the row and column models, followed by the p-value from a paired t-test in parentheses. Significant values are in bold.

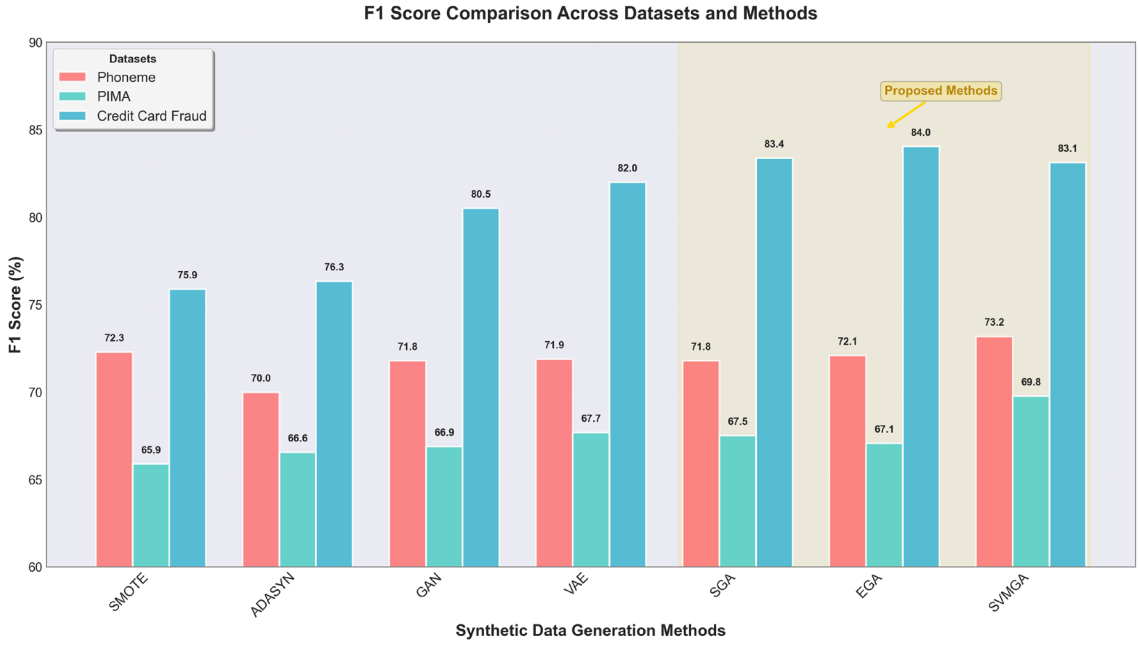

Fig. 10. F1 Score comparison across synthetic data generation methods on three benchmark datasets. The proposed genetic algorithm variants (SGA, EGA, SVMGA) exhibit superior performance compared to traditional approaches, with SVMGA achieving optimal F1 scores on Phoneme (73.18%) and PIMA (69.78%) datasets, while EGA exhibits superior performance on the Credit Card Fraud dataset (84.04%).

(ADASYN, SMOTE-based, Original) have lower AUC values of 0.89 and 0.9, suggesting slightly less effective classification performance compared to SVM-based GA, EGA, and SGA.

Statistical comparison of F1 scores on the PHONEME dataset Table 11 displays pairwise comparisons of mean F1-scores across models on the PHONEME dataset. Diagonal values indicate average F1-scores per model, and off-diagonal values show mean differences and associated p-values from paired t-tests. SVMGA attained the highest mean F1-score (73.18), outperforming several models with statistically significant improvements over ADASYN ($p = 0.0355$), VAE ($p = 0.0146$), and SGA ($p = 0.0399$). Although SVMGA also showed positive differences against other models such as GAN and EGA, the corresponding p-values indicate marginal or non-significant differences. Other GA-based models (SGA and EGA) remained competitive, though their relative advantages were not statistically significant. These findings further shows the effectiveness of GA-based methods, with SVMGA in particular showing statistically superior performance on the PHONEME dataset.

Comparative performance analysis

A comparative analysis of F1 scores across all synthetic data generation techniques and benchmark datasets is presented in Fig. 10. The analysis reveals the consistent superiority of the proposed genetic algorithm variants (SGA, EGA, and SVMGA) over conventional oversampling methods and deep learning-based approaches.

The experimental results indicate that the proposed GA-based methodologies consistently outperform baseline approaches across all evaluated datasets. SVMGA exhibits the most consistent performance characteristics, achieving optimal F1 scores on two of the three benchmark datasets. The comparative analysis

substantiates the effectiveness of integrating machine learning guidance within genetic algorithm frameworks for synthetic data generation, thereby validating the fundamental contributions of this research.

Feature importance analysis

A feature importance analysis was conducted using permutation importance across the three datasets: PIMA, Credit Card Fraud Detection, and Phoneme. This analysis identifies the most influential features contributing to synthetic data generation and overall model performance.

Key insights include:

- *PIMA Dataset*: Features such as Glucose (25%) and BMI (20%) were the most significant, reflecting their importance in determining diabetes risk. Pregnancies (15%) and Diabetes Pedigree Function (12%) also contributed substantially.
- *Credit Card Dataset*: Amount (20%), V3 (22%), and V2 (18%) emerged as the most critical features, highlighting their role in distinguishing fraudulent transactions.
- *Phoneme Dataset*: The feature Sh (30%) showed the highest influence in classifying sound categories, followed by Ao (25%) and Aa (20%).

The combined plot as shown in Fig. 11 illustrates the feature importance across datasets, emphasizing the diverse factors impacting classification performance. This analysis confirms the effectiveness of our GA-based approach in leveraging dataset-specific key features for improved model accuracy.

Ablation study: evaluating GA components

To thoroughly assess the contributions of the Genetic Algorithm (GA) components–namely, crossover and mutation–to the quality of synthetic data and model performance, we conducted an ablation study across the three benchmark datasets: Credit Card Fraud Detection, PIMA Indian Diabetes, and Phoneme. Each component was systematically excluded from the GA process, and the resultant effects on performance metrics were analyzed.

Impact of crossover

The crossover operator in GA plays a pivotal role in combining genetic information from parent solutions to generate diverse offspring. To evaluate its impact, experiments were conducted with the crossover operator entirely disabled. The absence of crossover resulted in a significant reduction in genetic diversity within the synthetic data. As a consequence, the generated data failed to adequately explore critical regions of the feature space. This limitation was reflected in diminished model performance across all datasets, particularly in metrics

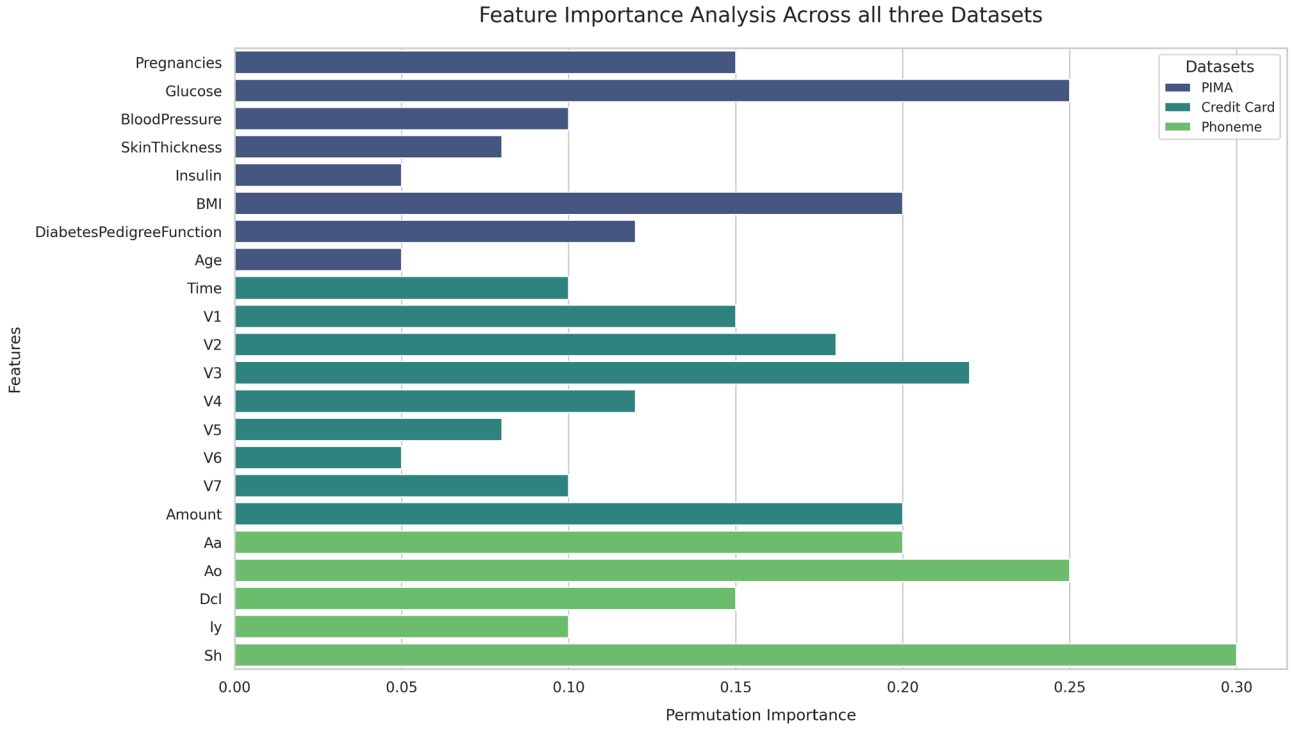

Fig. 11. Permutation importance analysis across the three datasets: PIMA, Credit Card Fraud Detection, and Phoneme. The graph highlights the relative contribution of each feature to the model's performance, as measured by the decrease in accuracy when the feature's values are permuted.

such as F1-score, precision, and ROC AUC. The PIMA Indian Diabetes dataset, in particular, exhibited a sharp decline in classification performance due to its inherently lower data diversity.

Impact of mutation
Mutation introduces random variations to offspring, enabling exploration beyond the local optima in the solution space. To isolate its effect, experiments were performed with mutation disabled, while retaining all other GA components. The absence of mutation led to premature convergence in the population, with the algorithm failing to explore potentially beneficial areas of the solution space. This resulted in suboptimal synthetic data generation, particularly evident in the Credit Card Fraud Detection dataset, where the F1-score decreased significantly. The lack of mutation restricted the diversity of generated samples, thereby limiting the classifier's ability to generalize to unseen data.

SVM vs. logistic regression initialization analysis
To evaluate the impact of different initialization strategies, we compared SVM-based initialization with logistic regression-based initialization and random initialization. The SVM-based approach uses decision boundary information to strategically position initial population members near classification boundaries, while logistic regression initialization uses probability-based positioning. Random initialization serves as the baseline approach. The results show that SVM-based initialization consistently outperforms both logistic regression and random initialization across all datasets. The geometric properties of SVM decision boundaries provide superior guidance for identifying critical regions in the feature space where synthetic samples can be most beneficial. Logistic regression initialization, while better than random initialization, lacks the explicit boundary information that makes SVM-based initialization particularly effective for minority class synthesis. ‘

Combined analysis
A comparative analysis was conducted between the full GA implementation and the ablated versions (no crossover and no mutation). Table 12 presents a detailed comparison of performance metrics across the three datasets. The results demonstrate the essential role of both components in improving synthetic data quality and enhancing model performance.

Discussion of findings
The ablation study results in Table 12 highlight the significant contributions of the Genetic Algorithm (GA) components–crossover and mutation–to synthetic data quality and model performance. In the Credit Card Fraud dataset, characterized by severe class imbalance, the removal of either crossover or mutation led to a marked decline in performance. The full GA implementation achieved the highest results (F1-score: 0.82, precision: 0.82, ROC AUC: 0.92). Similarly, in the PIMA Diabetes dataset, with moderate class imbalance, performance decreased when crossover was removed (F1-score: 0.62, precision: 0.50) and when mutation was excluded (F1-score: 0.61, precision: 0.53). The full GA improved all metrics (F1-score: 0.71, precision: 0.59, ROC AUC: 0.76). In the Phoneme dataset, with a more balanced class distribution, performance also decreased with the removal of either component, with the full GA achieving the best results (F1-score: 0.72, precision: 0.60, ROC AUC: 0.82). Across all datasets, the full GA consistently outperformed the ablated versions, underscoring the importance of both crossover and mutation. Crossover contributes to exploring diverse solutions, while mutation prevents premature convergence by maintaining genetic diversity. These findings emphasize the effectiveness of the GA approach in generating high-quality synthetic data and improving model performance across different datasets.

Computational cost analysis

In addition to classification performance, we also evaluated the computational cost of each synthetic data generation method in terms of runtime and memory usage. This comparison provides insight into the practical feasibility of each approach, especially when applied to large datasets.

Table 13 reports the average time and peak memory consumed during synthetic data generation for the Credit Card Fraud Detection dataset, which has the largest volume among the three benchmarks.

As expected, interpolation-based methods such as SMOTE and ADASYN are the most efficient in terms of both time and memory, making them suitable for real-time or resource-constrained environments. However,

Dataset	Metric	No crossover	No mutation	LR init	Random init	Full SVM-GA
Credit card fraud	F1-score	0.70	0.74	0.78	0.73	0.82
	Precision	0.67	0.72	0.76	0.71	0.82
	ROC AUC	0.84	0.82	0.88	0.85	0.92
PIMA diabetes	F1-score	0.62	0.61	0.68	0.64	0.71
	Precision	0.50	0.53	0.56	0.52	0.59
	ROC AUC	0.65	0.69	0.73	0.70	0.76
Phoneme	F1-score	0.65	0.68	0.70	0.67	0.72
	Precision	0.49	0.51	0.57	0.53	0.60
	ROC AUC	0.72	0.71	0.78	0.75	0.82

Table 12. Performance metrics for comprehensive ablation study across datasets.

Method	Runtime (s)	Peak memory (MB)
SMOTE	4.1	84
ADASYN	4.4	96
GAN	5.6	106
VAE	7.4	112
GA (SGA)	12.3	147
Elitist GA (EGA)	12.9	155
SVM-based GA (SVMGA)	16.7	186

Table 13. Runtime and memory usage comparison (Phoneme Dataset).

these methods offer limited data diversity and may contribute to overfitting. In contrast, GA-based methods require more computational resources due to their iterative nature and fitness-based optimization. Among them, the SGA method is the most efficient, while the SVM-guided GA shows the highest cost due to model-driven evaluation. Nonetheless, the overhead remains acceptable for offline preprocessing, especially when weighed against the significant gains in classification performance and generalization. For large-scale applications, runtime can be mitigated using parallel or GPU-accelerated fitness evaluations, and memory usage can be reduced through batch-wise evolution strategies. This makes the GA-based approach a scalable and effective option for data augmentation in high-stakes domains.

Discussion and conclusion

This study addresses the challenge of class imbalance in datasets, a major issue in various fields facing imbalance data classification problems. Various techniques are investigated, including the Synthetic Minority Over-sampling Technique (SMOTE)[9], Adaptive Synthetic Sampling Approach (ADASYN)[11], Generative Adversarial Networks (GANs)[71], Variational Autoencoders (VAEs)[63] and Genetic Algorithms (GAs), to generate synthetic data and improve classification performance. Our findings reveal that while SMOTE, ADASYN, VAEs and GANs can balance datasets, they often fail to significantly enhance model performance metrics when the class ratio is high. These methods inadequately address severe imbalances, resulting in limited improvements in classification accuracy due to unwanted noise during synthetic data generation. Advanced variations like RN-SMOTE (with DBSCAN for noise reduction[72]) and ADASYN-LOF[62] offer improvements but still face challenges in eliminating noise completely. These advanced variations of SMOTE and ADASYN present interesting avenues for future comparative studies against GA-based approaches. In contrast, our proposed approaches, Simple Genetic Algorithm (SGA), Elitist Genetic Algorithm (EGA) optimized through logistic regression, and Support Vector Machine-based Genetic Algorithm (SVMGA) demonstrate notable improvements across all performance metrics: accuracy, precision, recall, F1-score, ROC-AUC, and Precision-Recall curves. These GA-based models achieve significant results by generating high-quality synthetic data, thereby improving classification of minority class samples. Comparative analysis with all other techniques shows that the GA-based synthetic data generation approach significantly outperforms these traditional methods, particularly in scenarios with limited noise and extreme imbalance.

While GA-based methods are effective in generating high-quality synthetic data, their scalability to large datasets or real-time applications requires consideration. Due to their iterative nature and reliance on population-wide fitness evaluations, GAs are best suited for offline preprocessing in moderate-sized datasets. However, scalability can be improved through parallelization, as fitness evaluations across individuals are independent and can be executed concurrently on modern multi-core or GPU architectures. For large-scale applications, the GA framework can be adapted using mini-batch processing or by limiting the population and number of generations based on available computational resources. In real-time scenarios where latency is critical, traditional methods like SMOTE or ADASYN may be preferable due to their low overhead. Nonetheless, the proposed GA-based approach remains a viable solution for high-stakes domains (e.g., fraud detection or medical diagnosis) where offline augmentation can substantially enhance model performance. Beyond binary classification, many real-world problems involve multiclass imbalance, where one or more classes have significantly fewer instances. The proposed GA-based framework can be extended to such settings by evolving synthetic samples separately for each minority class, using one-vs-all strategies or class-specific fitness functions that target the boundaries between each minority class and all others. To preliminarily validate this extension, we propose an experiment using the UCI Letter Recognition dataset or CIFAR-10 with artificially induced class imbalance. For each minority class, the GA would generate synthetic data using a targeted fitness function (e.g., SVM trained in a one-vs-rest fashion), and classifier performance would be compared to multiclass SMOTE and ADASYN baselines. Key metrics such as macro-averaged F1 score and per-class recall can be used to evaluate improvement. This experiment would offer early evidence of the method's adaptability to more complex imbalanced scenarios and pave the way for future work on multiclass GA-guided augmentation.

While the GA-based synthetic data generation method has demonstrated effectiveness, it is essential to recognize that any data augmentation technique, including those based on genetic algorithms, carries the potential for bias. Specifically, biases inherent in the original data or any models used for guiding the process–such as classifiers like logistic regression or SVM–could be inadvertently transferred or even amplified in the synthetic data. In sensitive domains like healthcare or finance, such biases could lead to skewed decision-making or perpetuate existing disparities, raising concerns about fairness and equity. To address this, it's crucial that synthetic data generation techniques, including those based on GAs, incorporate robust mechanisms to assess

and control for biases. This could involve the careful design of fitness functions that are sensitive to fairness considerations and the implementation of post-generation validation processes to check for disproportionate representation of subgroups. Additionally, integrating fairness-aware objectives or constraints could further enhance the ethical deployment of synthetic data, especially in high-risk applications. In addition to bias concerns, privacy is another critical aspect when using GA for synthetic data generation, particularly when augmenting real-world data. In applications involving sensitive personal information, such as healthcare or finance, the use of synthetic data must be carefully considered to avoid inadvertent leakage of private or identifiable information. While synthetic data is intended to reduce risks related to privacy by creating non-real data points, there remains the possibility that patterns in the synthetic data could be traced back to the original data. To mitigate these risks, it is crucial to implement privacy-preserving techniques, such as differential privacy, during the synthetic data generation process. This could include adding noise or ensuring that the generated data cannot be reverse-engineered to reveal sensitive individual information.

While our GA-based approach demonstrates significant improvements in handling imbalanced datasets, it has certain limitations. The computational complexity of GAs may pose challenges for very large datasets, and the effectiveness of the method depends on the quality of the ML models used for initialization. Future work could explore optimizing the computational efficiency of the GA process, extending the approach to multi-class imbalance problems, and applying it to complex data types such as images or time-series data. Our current evaluation is based on three datasets: the Credit Card Fraud Detection dataset (a cyber-security application) and two medical datasets (PIMA Indian Diabetes and Phoneme). Expanding to more diverse domains and complex data types would improve generalization in future work. In conclusion, this research demonstrates the potential of genetic algorithms, particularly when optimized using SVM-based fitness functions and population initialization, to generate high-quality synthetic data that enhances classification performance. These findings open opportunities for further exploration of advanced GA techniques in various applications, ultimately contributing to more accurate and reliable predictive models in imbalanced datasets. The proposed GA-based methods provide a better alternative by ensuring greater diversity and variability in the generated synthetic data. The insights gained from this study are expected to benefit researchers and practitioners in domains where data imbalance is a critical challenge, offering a resilient solution for improving model performance on minority classes without compromising overall accuracy.

Data availability

The datasets utilized in this study are publicly available and sourced from established repositories. The Credit Card Fraud Detection dataset was obtained from[59] (https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud), published on Kaggle. The Pima Indian Diabetes dataset[60] (https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/XFOZQR) is accessible via Harvard Dataverse. Furthermore, the phoneme dataset[61], originally collected for the European ESPRIT 5516 project (ROARS) for French and Spanish speech recognition, was retrieved from Kaggle (https://www.kaggle.com/datasets/timrie/phoneme). All datasets are available through their respective original sources, ensuring full reproducibility of our research findings.

Received: 30 January 2025; Accepted: 27 June 2025
Published online: 07 October 2025

References

1. Li, C., Li, Z., Jun, X. & Pi, W. The impact of data quality on neural network models. *Cyber Secur. Intell. Anal.* **928**, 657–665. https://doi.org/10.1007/978-3-030-15235-2_91 (2019).
2. El-Rashidy, N., Tarek, Z., Elshewey, A. M. & Shams, M. Y. Multitask multilayer-prediction model for predicting mechanical ventilation and the associated mortality rate. *Neural Comput. Appl.* **37**, 1321–1343 (2025).
3. Elshewey, A. M. & Osman, A. M. Orthopedic disease classification based on breadth-first search algorithm. *Sci. Rep.* **14**, 23368 (2024).
4. Tarek, Z., Alhussan, A. A., Khafaga, D. S., El-Kenawy, E.-S.M. & Elshewey, A. M. A snake optimization algorithm-based feature selection framework for rapid detection of cardiovascular disease in its early stages. *Biomed. Signal Process. Control* **102**, 107417 (2025).
5. Elkenawy, E.-S.M., Alhussan, A. A., Khafaga, D. S., Tarek, Z. & Elshewey, A. M. Greylag goose optimization and multilayer perceptron for enhancing lung cancer classification. *Sci. Rep.* **14**, 23784 (2024).
6. Riston, T. et al. Oversampling methods for handling imbalance data in binary classification. In Gervasi, O. *et al.* (eds.) *Computational Science and Its Applications*–ICCSA 2023 Workshops*, 3–23 (Springer Nature Switzerland, Cham, 2023).
7. Mostafaei, S. H. & Tanha, J. Ouboost: Boosting based over and under sampling technique for handling imbalanced data. *Int. J. Mach. Learn. Cybern.* **14**, 3393–3411. https://doi.org/10.1007/s13042-023-01839-0 (2023).
8. Mohammed, R., Rawashdeh, J. & Abdullah, M. Machine learning with oversampling and undersampling techniques: Overview study and experimental results. In *Proceedings of the 11th International Conference on Information and Communication Systems (ICICS)*, https://doi.org/10.1109/ICICS49469.2020.239556 (IEEE, Irbid, Jordan, 2020).
9. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357. https://doi.org/10.1613/jair.953 (2002).
10. Han, H., Wang, W. Y. & Mao, B. H. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In Huang, D. S., Zhang, X. P. & Huang, G. B. (eds.) *Advances in Intelligent Computing (ICIC 2005)*, vol. 3644 of *Lecture Notes in Computer Science*, 878–887 (Springer, Berlin, Heidelberg, 2005).
11. He, H., Bai, Y., Garcia, E. A. & Li, S. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* https://doi.org/10.1109/IJCNN.2008.4633969 (IEEE, Hong Kong, 2008).
12. Wang, A. X., Chukova, S. S. & Nguyen, B. P. Synthetic minority oversampling using edited displacement-based k-nearest neighbors. *Appl. Soft Comput.* **148**, 110895. https://doi.org/10.1016/j.asoc.2023.110895 (2023).
13. Munguía, M. J. C., Lara, R. E., Eleuterio, A. R., Gutirrez, E. E. G. & López, F. D. R. Density-based clustering to deal with highly imbalanced data in multi-class problems. *Mathematics* https://doi.org/10.3390/math11184008 (2023).

14. Ling, C. X. & Sheng, V. S. Cost-sensitive learning. In Sammut, C. & Webb, G. I. (eds.) *Encyclopedia of Machine Learning*, 231–235, https://doi.org/10.1007/978-0-387-30164-8_181 (Springer, Boston, MA, 2011).
15. Chen, Z., Lin, T., Xia, X., Xu, H. & Sha, D. A synthetic neighborhood generation based ensemble learning for the imbalanced data classification. *Appl. Intell.* **48**, 2441–2457. https://doi.org/10.1007/s10489-017-1088-8 (2018).
16. Cinquini, M., Giannotti, F. & Guidotti, R. Boosting synthetic data generation with effective nonlinear causal discovery. *In IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)* https://doi.org/10.1109/CogMI52975.2021.00016 (IEEE, Atlanta, GA, USA 2021).
17. Kamalov, F. Kernel density estimation based sampling for imbalanced class distribution. *Inf. Sci.* **512**, 1192–1201. https://doi.org/10.1016/j.ins.2019.10.017 (2020).
18. Aggarwal, U., Popescu, A. & Hudelot, C. Active learning for imbalanced datasets. *In IEEE Winter Conference on Applications of Computer Vision (WACV)* https://doi.org/10.1109/WACV45572.2020.9093475 (IEEE, Snowmass, CO, USA 2020).
19. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. Preprint at https://doi.org/10.48550/arXiv.1312.6114 (2014).
20. Liu, J. Importance-smote: A synthetic minority oversampling method for noisy imbalanced data. *Soft Comput.* **26**, 1141–1163. https://doi.org/10.1007/s00500-021-06532-4 (2022).
21. Yang, Y., Khorshidi, H. A. & Aickelin, U. A diversity-based synthetic oversampling using clustering for handling extreme imbalance. *SN Comput. Sci.* https://doi.org/10.1007/s42979-023-02249-3 (2023).
22. Araf, I., Idri, A. & Chairi, I. Cost-sensitive learning for imbalanced medical data: A review. *Artif. Intell. Rev.* https://doi.org/10.1007/s10462-023-10652-8 (2024).
23. Swana, E. F., Doorsamy, W. & Bokoro, P. Tomek link and smote approaches for machine fault classification with an imbalanced dataset. *Sensors* **22**, 3246. https://doi.org/10.3390/s22093246 (2022).
24. Holland, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (MIT Press, Cambridge, 1992).
25. Alaoui, N., Adamou-Mitiche, A. B. H. & Mitiche, L. Effective hybrid genetic algorithm for removing salt and pepper noise. *IET Image Process.* **14**, 289–296. https://doi.org/10.1049/iet-ipr.2019.0566 (2020).
26. Arkhipov, D. I., Wu, D., Wu, T. & Regan, A. C. A parallel genetic algorithm framework for transportation planning and logistics management. *IEEE Access* https://doi.org/10.1109/ACCESS.2020.2997812 (2020).
27. AlKhafaji, B. J., Salih, M. A., Nabat, Z. M. & Shnain, A. S. Segmenting video frame images using genetic algorithms. *Period. Eng. Nat. Sci.* **8**, 1106–1114. https://doi.org/10.21533/pen.v8i2.1351 (2020).
28. Abbasi, M. et al. An efficient parallel genetic algorithm solution for vehicle routing problem in cloud implementation of the intelligent transportation systems. *J. Cloud Comput.* https://doi.org/10.1186/s13677-020-0157-4 (2020).
29. Li, T., Yin, Y., Yang, B., Hou, J. & Zhou, K. A self-learning bee colony and genetic algorithm hybrid for cloud manufacturing services. *Computing* **104**, 1977–2003. https://doi.org/10.1007/s00607-022-01079-0 (2022).
30. Elshewey, A. M., Alhussan, A. A., Khafaga, D. S., Elkenawy, E.-S.M. & Tarek, Z. EEG-based optimization of eye state classification using modified-BER metaheuristic algorithm. *Sci. Rep.* **14**, 24489 (2024).
31. Gang, L. Genetic algorithm and its application in software test data generation. *In Proceedings of the International Conference on Applied Intelligence and Sustainable Computing (ICAISC)* https://doi.org/10.1109/ICAISC58445.2023.10200303 (Dharwad, India 2023).
32. Rodrigues, D. S., Delamaro, M. E., Corrêa, C. G. & Nunes, F. L. S. Using genetic algorithms in test data generation: A critical systematic mapping. *ACM Comput. Surveys* **51**, 1–23. https://doi.org/10.1145/3182659 (2018).
33. Kim, J. W. & Jang, B. Deep learning-based privacy-preserving framework for synthetic trajectory generation. *J. Netw. Comput. Appl.* **206**, 103459. https://doi.org/10.1016/j.jnca.2022.103459 (2022).
34. Marefat, A., Nematollahi, M. A., Oyelere, S. S. & Hussain, S. A review of generative models in generating synthetic attack data for cybersecurity. *Electronics* https://doi.org/10.3390/electronics13020322 (2024).
35. Srivastava, A., Sinha, D. & Kumar, V. Wcgan-GP based synthetic attack data generation with GA based feature selection for ids. *Comput. Secur.* **134**, 103432. https://doi.org/10.1016/j.cose.2023.103432 (2023).
36. Batista, G. E. A. P. A., Prati, R. C. & Monard, M. C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **6**, 20–29. https://doi.org/10.1145/1007730.1007735 (2004).
37. Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn.* **20**, 273–297. https://doi.org/10.1007/BF00994018 (1995).
38. Cox, D. R. The regression analysis of binary sequences. *J. R. Stat. Soc. Ser. B (Methodological)* **20**, 215–232 (1958).
39. Johnson, J. M. & Khoshgoftaar, T. M. Survey on deep learning with class imbalance. *J. Big Data* **6**, 1–54. https://doi.org/10.1186/s40537-019-0192-5 (2019).
40. Pereira, J. & Saraiva, F. A comparative analysis of unbalanced data handling techniques for machine learning algorithms to electricity theft detection. *In Proceedings of the IEEE Congress on Evolutionary Computation (CEC)* https://doi.org/10.1109/CEC48606.2020.9185822 (Glasgow, UK 2020).
41. Pereira, P. J., Pereira, A., Cortez, P. & Pilastri, A. A comparison of machine learning methods for extremely unbalanced industrial quality data. In Marreiros, G., Melo, F. S., Lau, N., Cardoso, H. L. & Reis, L. P. (eds.) *Progress in Artificial Intelligence*, vol. 12981 of *Lecture Notes in Computer Science*, 561–572, https://doi.org/10.1007/978-3-030-86230-5_44 (Springer, Cham, 2021).
42. Khoshgoftaar, T. M., Golawala, M. & Hulse, J. V. An empirical study of learning from imbalanced data using random forest. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*,https://doi.org/10.1109/ICTAI.2007.46 (Patras, Greece, 2007).
43. Chen, Q., Zhang, X., Wang, Y., Zhai, Z. & Yang, F. Applying a random forest approach to imbalanced dataset on network monitoring analysis. In Lu, W., Zhang, Y., Wen, W., Yan, H. & Li, C. (eds.) *Cyber Security*, vol. 1699 of *Communications in Computer and Information Science*, 28–37, https://doi.org/10.1007/978-981-19-8285-9_2 (Springer, Singapore, 2022).
44. Lu, C., Lin, S., Liu, X. & Shi, H. Telecom fraud identification based on adasyn and random forest. In *Proceedings of the 5th International Conference on Computer and Communication Systems (ICCCS)*,https://doi.org/10.1109/ICCCS49078.2020.9118521 (Shanghai, China, 2020).
45. Afreen, S., Bhurjee, A. K. & Aziz, R. M. Cancer classification using RNA sequencing gene expression data based on game shapley local search embedded binary social ski-driver optimization algorithms. *Microchem. J.* **205**, 111280 (2024).
46. Esnaashari, M. & Damia, A. H. Automation of software test data generation using genetic algorithm and reinforcement learning. *Exp. Syst. Appl.* **183**, 115446. https://doi.org/10.1016/j.eswa.2021.115446 (2021).
47. Kim, J. H. & Hwang, Y. Gan-based synthetic data augmentation for infrared small target detection. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–12. https://doi.org/10.1109/TGRS.2022.3179891 (2022).
48. Joshi, A. A. & Aziz, R. M. Soft computing techniques for cancer classification of gene expression microarray data: A three-phase hybrid approach. In *Computational Intelligence for Data Analysis*, 92–113 (Bentham Science Publishers, 2024).
49. Yaqoob, A., Musheer Aziz, R. & Verma, N. K. Applications and techniques of machine learning in cancer classification: A systematic review. *Human Centric Intell. Syst.* **3**, 588–615 (2023).
50. Huang, S. W. *et al.* Auggan: Cross domain adaptation with GAN-based data augmentation. In Ferrari, V., Hebert, M., Sminchisescu, C. & Weiss, Y. (eds.) *Computer Vision – ECCV 2018*, vol. 11213 of *Lecture Notes in Computer Science*, 731–744, https://doi.org/10.1007/978-3-030-01234-5_44 (Springer, Cham, 2018).
51. Zareapoor, M., Shamsolmoali, P. & Yang, J. Oversampling adversarial network for class-imbalanced fault diagnosis. *Mech. Syst. Signal Process.* **149**, 107175. https://doi.org/10.1016/j.ymssp.2020.107175 (2021).

52. Wu, Y., Ding, Y. & Feng, J. Smote-boost-based sparse Bayesian model for flood prediction. *EURASIP J. Wirel. Commun. Netw.* **1–15**, 2020. https://doi.org/10.1186/s13638-020-01689-2 (2020).
53. Wang, R. Adaboost for feature selection, classification and its relation with SVM: A review. *Phys. Procedia* **25**, 800–807. https://doi.org/10.1016/j.phpro.2012.03.160 (2012).
54. Hubin, A. & Storvik, G. Sparse Bayesian neural networks: Bridging model and parameter uncertainty through scalable variational inference. *Mathematics* **12**, 788. https://doi.org/10.3390/math12060788 (2024).
55. Nguyen, Q., Vu, T., Tran, A. & Nguyen, K. Dataset diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. In Oh, A. et al. (eds.) *Advances in Neural Information Processing Systems. Proceedings of the NeurIPS Conference*, vol. 36, 76872–76892 (2023).
56. Ge, Y. F., Wang, H., Cao, J., Zhang, Y. & Jiang, X. Privacy-preserving data publishing: An information-driven distributed genetic algorithm. *World Wide Web* **27**, 144–151. https://doi.org/10.1007/s11280-024-01241-y (2024).
57. Rahman, R. U., Kumar, P., Mohan, A., Aziz, R. M. & Tomar, D. S. A novel technique for image captioning based on hierarchical clustering and deep learning. *SN Comput. Sci.* **6**, 360 (2025).
58. Sharma, A., Kumar, P., Ben, D., Bikhani, M. & Aziz, R. M. Improved GA based clustering with a new selection method for categorical dental data. In *Swarm Optimization for Biomedical Applications*, 172–192 (CRC Press, 2025).
59. Benchaji, I., Douzi, S. & Ouahidi, B. E. Credit card fraud detection dataset (2021). Published on Kaggle, used for developing a fraud detection model. Available from: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.
60. Bartley, C. *Replication data for: Pima Indians diabetes* https://doi.org/10.7910/DVN/XFOZQR (2016).
61. Cappel, D. V. & Sintra, T. Phoneme dataset (1994). Used in the European ESPRIT 5516 project: ROARS, for French and Spanish speech recognition. Available from: https://www.kaggle.com/datasets/timrie/phoneme.
62. Qing, Z. et al. Adasyn-lof algorithm for imbalanced tornado samples. *Sensors* **13**, 544. https://doi.org/10.3390/atmos13040544 (2022).
63. Kingma, D. P., Welling, M. *et al.* Auto-encoding variational Bayes (2013).
64. Zou, X., Hu, Y., Tian, Z. & Shen, K. Logistic regression model optimization and case analysis. In *Proceedings of the 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, https://doi.org/10.1109/ICCSNT47585.2019.8962457 (Dalian, China, 2019).
65. Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L. & Lopez, A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing* **408**, 189–215. https://doi.org/10.1016/j.neucom.2019.10.118 (2020).
66. Rosales-Pérez, A., García, S. & Herrera, F. Handling imbalanced classification problems with support vector machines via evolutionary bilevel optimization. *IEEE Trans. Cybern.* **53**, 4735–4747. https://doi.org/10.1109/TCYB.2022.3163974 (2022).
67. Katoch, S., Chauhan, S. S. & Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **80**, 8091–8126. https://doi.org/10.1007/s11042-020-10139-6 (2021).
68. Luque, A., Carrasco, A., Martín, A. & Heras, A. D. L. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recogn.* **91**, 216–231. https://doi.org/10.1016/j.patcog.2019.02.023 (2019).
69. Bradley, A. P. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recogn.* **30**, 1145–1159. https://doi.org/10.1016/S0031-3203(96)00142-2 (1997).
70. McKnight, S. et al. A comparison of methods for generating synthetic training data for domain adaptation of deep learning models in ultrasonic non-destructive evaluation. *NDT E Intern.* **141**, 102978. https://doi.org/10.1016/j.ndteint.2023.102978 (2024).
71. Goodfellow, I. et al. Generative adversarial networks. *Commun. ACM* **63**, 139–144 (2020).
72. Arafa, A., El-Fishawy, N., Badawy, M. & Radad, M. Rn-smote: Reduced noise smote based on dbscan for enhancing imbalanced data classification. *J. King Saud Univ. Comput. Inf. Sci.* **34**, 5059–5074. https://doi.org/10.1016/j.jksuci.2022.06.005 (2022).

Author contributions

Muhammad Usman Safder: conceptualization, formal analysis, methodology implementation, writing—original draft. Syed Sarib Naveed: methodology pipeline implementation, original draft writing, validation, investigation. Khawar Khurshid: supervision, validation, methodology, investigation, writing—original draft. Ahmad Salman: supervision, validation, investigation. Imran Fareed Nizami: supervision, validation, investigation.

Declarations

Competing interests

The authors declare no competing financial or non-financial interests that could have influenced the research presented in this manuscript.

Declaration of the AI-assisted technologies

During the preparation of this manuscript, the authors have utilized the basic feature of the OpenAI's GPT model, solely for the purpose of eliminating grammatical errors and improving the overall readability of the document. The authors have thoroughly reviewed the final manuscript and take full responsibility for the content of the published article.

Additional information

Correspondence and requests for materials should be addressed to K.K.