International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I11P113

Volume 74 Issue 11, 174-186, November 2026
© 2026 Seventh Sense Research Group®

Original Article

Optimizing Minority Class Detection in Deep Learning through Genetic Algorithm Hybridization and Evolutionary Sampling with TabNet

Mareeswari G[1], Erana Veerappa Dinesh S[2], Manimegalai M[3], Chidambaram S[4]

[1]*Department of Information Technology, Ramco Institute of Technology, Rajapalayam, Tamilnadu, India,*
[2]*Department of Computer Science and Business Systems, Ramco Institute of Technology, Rajapalayam, Tamilnadu, India,*
[3,4]*Department of Information Technology, National Engineering College, Kovilpatti, Tamilnadu, India.*

[1]*Corresponding Author: gmarees92@gmail.com*

Received: 17 June 2026 Revised: 12 September 2026 Accepted: 18 September 2026 Published: 25 November 2026

Abstract *- In machine learning, the lack of data homogeneity is a testing matter, often distorting the models towards a certain group class while dismissing efficient confidence measures in the classes that are less common. This work examines various sampling methods to balance classes in ML models, work in practice SMOTE, a Genetic Algorithm (GA)-based undersampling and oversampling, a GA-Hybrid, a Cluster-Guided GA, and a Multi-Objective GA (MOGA, NSGA-II). The performance examination of these sampling methods occurred using five deep learning frameworks, specifically TabNet, a 3-layer MLP, a CNN, an LSTM, and a transformer Encoder. The results of the statistical comparison of the baseline model F1 scores provide high accuracy, but fairly low F1 scores, whereas applying all of the sampling methods improved their balanced performance. The hybrid method, GA-Hybrid, achieved the highest F1 score consistently across all three classifications, such as TabNet, LSTM, and transformer Encoder.*

Keywords *- Imbalanced data, Sampling strategies, Genetic Algorithm, Deep learning, Hybrid sampling, Classification performance.*

1. Introduction

Class imbalances refer to a significant challenge in supervised learning, where there is a clear dominant class and a less common class. During training on imbalanced datasets, models will typically achieve a very high accuracy rate (in some cases, well above 95% accuracy). However, as the performance assessments tend to be biased towards the majority class, under this class imbalance, the model will fail at generalization of data from the minority class. In general application of supervision learning classification, accuracy is not a reliable means of assessing performance, particularly with class imbalances, as there is some inherent weight based on a class disproportionately making up the assessment of performance. Therefore, including the G-mean score, the F1 score, and Matthews Correlation Coefficient may be more appropriate means of assessing performance outside of traditional assessment methods. Historically, multiple approaches or alleviate this issue, from basic to more advanced strategies. The Synthetic Minority Oversampling Technique (SMOTE) [1] is likely the most often adopted method in use currently, in which synthetically creating samples of the minority class can make the dataset more balanced. However, one of the significant drawbacks of this method is that while it creates minority samples, it does bring noise and potentially less informative data points into the dataset, inherently compromising the performance of the classifier, particularly with high amounts of imbalance.

The incorporation of evolutionary algorithms with resampling techniques examined. Alex et al. [1], for example, used a GA-driven feature selection method along with SMOTE and a deep convolution neural network, resulting in improved accuracy and F1-score. Other studies used GA-driven undersampling methods [3], which maintained majority class diversity without exacerbating bias. Additional hybrid methods incorporated GA optimization with oversampling, boundary refinement, or cluster guidance, leading to enhanced minority class classification [4]. Most recently, model-driven resampling methods such as GMM Sampling [5] shown that using data difficulty in a resampling procedure can consistently improve multi-class imbalance issues in few-shot learning scenarios. This work provides a systematic exploration of GA-based sampling methods, including GA-undersampling, GA-oversampling, GA-hybrid (over-sampling + under-sampling), cluster-based GA, and Multi-Objective GA (MOGA and NSGA-II). Each one of

This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

these sampling approaches was tested using various deep learning architectures: TabNet, Multilayer Perceptron (MLP), One-Dimensional Convolutional Neural Network (1D-CNN), Long Short-Term Memory (LSTM), and transformer encoder. Overall, results suggest GA-hybrid sampling performs best (highest F1-score) while maintaining similar accuracy for predicting the target classes, thus providing balanced performance to address class imbalance in deep learning.

2. Related Works

Multiple surveys summarize several contemporary methodologies that emphasize hybrid resampling, model-aware methodologies, and generative approaches to solving the imbalance publicly [1, 2]. SMOTE-based oversampling has been successful in practice, but it has also received extensions in the literature to overcome its disadvantages. Such extensions include distance-aware variants and distribution-guided variants that improved the downstream classification performance and data generation in high-dimensional or noisily labeled datasets [3, 4].

Evolutionary algorithms, especially those that follow from Genetic algorithms, have also gained considerable attention when applied to imbalanced learning. First, there have been hybrid processes where a Tomek Links procedure has been combined with a GA process (TEUS) to improve the performance of simultaneously undersampling noisy instances while maintaining some variety in the class diversity [5]. There have also been some proposals to optimize or generate minority samples - such as GA-Random Oversampling, to lead to improved performance [6]. Finally, some studies, especially in the Deep Learning context, have been driving to GA-based feature selection and sample selection - proposing to improve classification performance for skewed distributions in DNN classifiers [7]. Recently, others have added to the field exploring evolutionary-based frameworks that encompass both over and undersampling in a single process. In this regard, simultaneous evolutionary resampling has been demonstrated to be better to maintain balance and data quality compared to a single-stage method [8].

In addition to evolutionary methods, Generative Adversarial Networks (GANs) have emerged as promising alternatives for minority class oversampling in imbalanced datasets. GRADs circumvent traditional oversampling approaches such as SMOTE by effectively learning the underlying distribution of minority classes and producing synthetic instances that more closely resemble real data. CWGANs have been developed to learn from tabular data to produce synthetic samples while ensuring high statistical consistency across attributes [10]. Results from CWGANs, for instance, produce diverse instances that conform to domain-specific distributions, thereby achieving high quality when, for example, generating imbalanced datasets for predictions of grid stability where the minority class is rare and requires more realistic representation. Further analysis of GAN-based oversampling at various levels of data difficulty showed that GANs were more beneficial than other conventional oversampling methods when the minority class was highly nonlinear or sparse in distribution [11].

Several sophisticated GAN frameworks have been developed with an emphasis on solving the stability challenges of the training process and improving fidelity of the data. Improved GAN frameworks such as I-GAN and extensions of the Wasserstein GAN (WGAN) have better loss functions and regularization techniques to stabilize adversarial training, leading to better-quality synthetic samples with reduced mode collapse [12]. The developments indicate the appropriateness of GAN-based oversampling as applied to disciplines involving critical categories, such as power systems, where stable operating regime rarely occur but have serious consequences. Researchers using hybrid pipelines in combination with GAN techniques to mimic training sample and feature optimally, provide inverting up-sampling of minority class. This would leads to a better balance for predictive classifiers for stability of grid operation.

Hybrid and ensemble based frameworks increasingly address grid stability classification through the combination of multiple algorithms to provide higher robustness and accuracy. Alaerjan et al. [14] presented an artificial intelligence-based fusion approach that employs various predictive models to enhance the resiliency of smart grids under dynamic conditions. Likewise, Breviglieri et al. [15] designed optimized deep learning architectures for grid stability prediction their results demonstrated that ensembles consisting of properly tuned deep models outperform traditional classifiers in convergence and predictive accuracy. These works show that hybrid and ensemble approaches are crucial to dealing with the complex nature of electrical grids, where single learners often cannot recreate the variability of operating conditions. In addition to ensemble learning, genetic algorithms have been widely adapted to address evolving data problems and improve learning pipelines. Lazebnik [16] presented GA as a strategy for addressing concept drift in learning models when evolutionary search was used to adapt classifiers to non-stable environments.

In the context of GA, Nemati et al. [17] integrated artificial neural networks with meta-heuristic search for hybrid feature selection to reduce dimensionality while maximizing predictive accuracy. In a single GA framework, Imani et al. [18] proposed a multi-objective GA framework capable of optimizing feature subset selection for different criteria, balancing multiple objectives. Simultaneously, Feng [19] generated a feature selection method based on GA that improved optimization via adaptive operators to converge and deliver a better solution. These studies demonstrate the flexibility of GA not only for feature selection, but for management of dynamic and imbalanced learning tasks, and provide evidence of the development of further hybrid GA

frameworks for the assessment of grid stability. Recently, Carvalho [20] published a data-driven overview of resampling techniques, further supporting the assertion that evolutionary and hybrid methods consistently outperform traditional oversampling methods across a variety of domains. While other approaches investigated resampling methods in isolation, this work builds upon their findings to provide a comparative evaluation of evolutionary sampling methods through the lens of several deep learning methods.The use of genetic algorithms for feature selection in datasets that consist of many features. Nematzadeh et al. [21] used a distance-based mutual congestion method combined with a GA with adaptive rates to deal with multicollinearity in medical datasets. Ahadzadeh et al. [22] suggested a Simple, Fast and Efficient (SFE) algorithm with the goal of identifying irrelevant and noisy features in high-dimensional data.

The researchers also claimed that the SFE's performance could be improved by hybridizing it with particle swarm optimization. Altarabichi et al. [23] presented a surrogate-assisted GA strategy to speed convergence in extremely large datasets, reporting achievability of improved quality of subsets while decreasing computation burden. Amini and Hu [24] developed a hybrid 'two-layer' GA-elastic net method that searches for optimal predictors via GA and refines selection through Elastic Net, and they demonstrated improvement of prediction accuracy via genetic data applications. For the prediction of electrical grid stability using tabnet combined with hybrid GA methodology for the balanced dataset. In order to compare the performance of tabnet other methodologies such as LSTM, CNN, and Transformer have been implemented.

3. Existing Methodologies

3.1. Long Short-Term Memory (LSTM) Networks in Minority Class Prediction

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) that can learn long-distance temporal dependencies in sequential time-series data while reducing the vanishing gradient problem that occurs in regular RNNs. Datasets containing a temporal component or sequential correlations for the minority class are a natural fit for LSTMs, often overlooked by traditional classifiers.

3.1.1. LSTM Architecture

An LSTM unit is made up of a cell state Ct and three main gates, the forget gate ft, input gate it, and output gate ot. These gates regulate the flow of information and keep only the relevant patterns from the past time steps.

Forget gate : Determines what information should be discarded from the previous cell state.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \quad (1)$$

Input Gate : Decides which new information to store in the cell state.

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \;\&$$

$$\widehat{C_t} = \tanh(W_c * [h_{t-1}, x_t] + b_c)$$

Cell State Update : Combines old and new information.

$$C_t = f_t * C_{t-1} + i_t * \widehat{C_t}$$

Output Gate : Determines the hidden state for the current time step.

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \quad (2)$$

Here, `xt` refers to the input at time step `t`. `ht` is the previous hidden state. `W` matrices and 'b' vectors are weights and biases that can be learned. 'σ' signifies the sigmoid function, and 'tanh' designates the hyperbolic tangent function.

3.1.2. Integrating LSTM with GA-Based and Evolutionary Sampling

LSTM models applied to imbalanced datasets result in having model bias to the majority class and low F1-scores on minority class predictions. This paper presents sampling methods based on Genetic Algorithm (GA) to balance the dataset. In GA-Oversampling, the minority class was expanded by generating synthetic minority samples using the genetic algorithm. A fitness function using similarity and diversity measures was established to ensure that the synthetic data was both representative of and diverse from the minority class, resulting in a more balanced dataset. In contrast, GA-Undersampling works by removing samples from majority class, selectively applying a genetic algorithm. The fitness function used in GA-Undersampling combines LSTM model performance and size of the majority subset to ensure that the class imbalance is minimized while still performing predictive modeling.

Algorithm 1. LSTM with GA-based sampling

```
Input:
    D_original          // Original imbalanced dataset
    GA_method           // GA-based sampling method: Oversample, Undersample, Hybrid, Cluster-Guided, MOGA
    LSTM_params         // LSTM hyperparameters (layers, units, learning rate, epochs, etc.)
    D_test              // Test dataset
```

Output:
LSTM_model // Trained LSTM model
Performance_metrics // Precision, Recall, F1-score for minority class

Step 1: Analyze class distribution & identify number of samples per class.

$$\text{Imbalance Ratio} = \frac{|D_{majority}|}{|D_{minority}|}$$

Step 2: Apply GA-based sampling to balance dataset
switch GA_method:
case "Oversample":
Initialize population 'P' of synthetic minority samples
For each generation:
$\text{Fitness}(x) = \alpha * \text{Similarity}(x, D_{minority}) + \beta * \text{Diversity}(x)$
Apply selection, crossover, mutation to generate new population
End
$D_{balanced} = D_{majority} \cup D_{minority} \cup \{x_i | x_i \in P_{best}\}$
case "Undersample":
// Step 2b: GA-Undersampling
Initialize population P of subsets of majority class
For each generation:
$\text{Fitness}(S) = \text{Accuracy}(\text{LSTM}(D_{minority} \cup S)) - |S|$
Apply selection, crossover, and mutation
End
$D_{balanced} = D_{minority} \cup \{x_j | x_j \in P_{best}\}$
case "Hybrid":
// Step 2c: GA-Hybrid (Oversample & Under sample)
Apply GA-Oversampling to generate $D_{synthetic}$
Apply GA-Undersampling to select subset $S_{best\ of\ majority}$
$D_{balanced} = D_{synthetic} \cup D_{minority} \cup S_{best}$
case "Cluster-Guided":
// Step 2d: Cluster-Guided GA
Cluster minority and majority samples
For each cluster:
Apply GA to generate or select samples
$\text{Fitness}(x) = \alpha * \text{Representativeness}(x, \text{cluster}) + \beta * \text{Accuracy}$
End
$D_{balanced} = \cup_{\{C \in clusters\}\{x_i \in P_{c_{best}}\}}$ // Union of GA-selected samples from all clusters
case "MOGA":
// Step 2e: Multi-Objective GA (MOGA/NSGA-II)
$f_1(x) = \text{Accuracy(LSTM)})$
$f_2(x) = 1 - \text{Imbalance}(D_{balanced})$
Initialize population P of sampling strategies
For each generation:
Perform non-dominated sorting to find Pareto-optimal solutions
Apply selection, crossover, mutation
End
$D_{balanced}$ = Dataset corresponding to Pareto-optimal solution
maximize $f_1(x)$, maximize $f_2(x)$
default:
raise Error("Invalid GA method")
Step 3: Initialize LSTM
LSTM model = initialize LSTM(LSTM params)
Forget gate: $f_t = \sigma(W_f[h_{t-1}, x_t] + b_f$
Input gate: $i_t = \sigma(W_i[h_{t-1}, x_t] + b_i$

The input gate determines which new information $\tilde{C}_t = tanh(W_c[h_{t-1}, x_t] + b_c$ is added to cell state.
Cell update: $C_t = f_t * C_{\{t-1\}} + i_t * \tilde{C}_t$
Output gate: $o_t = \sigma(W_o[h_{t-1}, x_t] + b_o$
The output gate decides the hidden state as $h_t = o_t * \tanh(C_t)$ that is propagating forward.
Step 4: Train LSTM on $D_{balanced}$
LSTM model.train ($D_{balanced.inputs}$, $D_{balanced.labels}$)
Loss = CrossEntropy(y_{true}, y_{pred})
Backpropagate Through Time (BPTT) to update 'W' and 'b'

Long Short-Term Memory (LSTM) networks are a specific type of Recurrent Neural Network (RNN) that enables the modeling of long-range dependencies in sequential data while overcoming the vanishing gradient problem present in traditional RNNs. An LSTM unit consists of the cell state Ct and three gates: the forget gate ft , input gate it and output gate ot . The forget gate determines what information to keep from the previous cell state Ct-1. Although LSTMs are effective for learning sequential patterns, they can often be poor performers on imbalanced data sets, as the model learns to bias toward the majority classes, patterns impairing predictive trustworthiness for the minority classes. One approach for addressing this problem is to balance the data set through GA-based sampling techniques prior to training the LSTM. For example, GA-Oversampling generates synthetic minority samples by employing a fitness function that weighs both similarity to existing minority samples and diversity for generated instances.

The Cluster-Guided Genetic Algorithm (GA) approach begins by constructing clusters of the minority aahxd majority samples to preserve diversity among the data samples and structure of the data. The GA then selects representative samples from within these clusters, or generates novel samples in such a way that the cluster preserves the original distribution of data samples. The fitness function measures both the representativeness of the clusters and improvement of the LSTM model so that the thus-fangled dataset reflects about the original distribution of the target classes, while also improving predictive performance.

The Multi-Objective Genetic Algorithm (MOGA) method, best represented through the Non-Dominated Sorting Genetic Algorithm II (NSGA-II), examines various objectives simultaneously, such as maximizing F1-score on minority classes and minimizing imbalanced classes. Rather than choosing a single objective metric, it considers Pareto-optimal selection to obtain datasets made between separate and competing objectives. This ensures that a dataset created from MOGA achieves higher performance across all precision and recall values, which could improve LSTM performance and allow for more equitable results across the data classes.The hybrid GA utilizes both oversampling and undersampling to achieve optimal representation of both classes, while the cluster-guided GA utilizes a cluster-based method of splitting the dataset into unique clusters and applying GA operations amongst each cluster to promote diversity and retain class distributions. Another alternative is to use a multi-objective GA known as Multi-Objective Genetic Algorithm (MOGA) or NSGA-II that strives to create balanced datasets by optimizing multiple objectives to be maximized, such as accuracy, while minimizing imbalance. This process selects Pareto-optimal candidates as the final training set. After the dataset is balanced with the desired GA-based strategy, the LSTM model will then be trained using standard backpropagation through time, employing cross-entropy loss.

3.2. Convolutional Neural Networks in Minority Class Prediction

Convolutional Neural Networks (CNNs) are a type of deep learning technique that are particularly powerful for extracting spatial or local patterns from structured data, as with time-series data, images, or sequences that have been reshaped for convolution. The general topology of a CNN includes layers of convolution, activation function (usually ReLU), pooling, and fully connected layers. A theorectical representation of the convolution operation at layer 'l' for a 1D CNN applied to sequential data can be written as:

$$y_i^{(l)} = \left(\sum_{k=0}^{k-1} w_k^{(l)} \cdot x_{i+k} + b^{(l)}\right) \quad (3)$$

In this formula, x_{i+k} denotes the input segment, $w_k^{(l)}$ corresponds to the convolutional filter weights, $b^{(l)}$ stands for the bias term, 'K' represents the kernel size, and σ indicates the activation function. Various pooling layers, such as max pooling, are capable of reducing dimensionality while conserving the most important features. The final fully connected layers map the features extracted to class predictions. CNNs are excellent in capturing local as well as hierarchical patterns from the data, but like other deep learning models, they can be biased to the majority class in imbalanced datasets.To work around this, GA-based and evolutionary sampling techniques can be employed prior to training the CNN. In GA-Oversampling, synthetic minority samples are created using a genetic algorithm and employ a fitness function that balances between similarity and diversity:

$$Fitness(x) = \alpha \cdot Similarity(x, D_{minority}) + \beta \cdot Diversity(x) \quad (4)$$

The balanced dataset is then:

$$D_{balanced} = D_{majority} \cup D_{minority} \cup D_{synthetic}$$

GA-Undersampling involves taking away data from the majority classes in order to help the model concentrate on the minority classes:

$$Fitness(S) = Accuracy\left(CNN\left(D_{minority} \cup S\right)\right) - |S|$$

with the final dataset:

$$D_{balanced} = D_{minority} \cup S_{best} \quad (5)$$

In Hybrid Genetic Algorithm (GA), the technique utilizes both Random OverSampling Instances (ROS), Random Under Sampling Instances (RUS), while Cluster Guided GA maintains the benefit of formal class diversity, applying (GA) methodology within a cluster of samples to assess a consideration for genetic process. Multi-Objective GA (MOGA/NSGA-II), attempts to outcome based on multiple objectives by considering objectives such as maximizing CNN accuracy (and values of) meanwhile minimizing class imbalance to determine Pareto-optimal datasets that are balanced dataset. After a balanced dataset has been determined, the dataset is then used to train each individual model in a standard CNN, with the back-propagation global learning algorithm and cross-entropy as the loss function:

$$Loss = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \quad (6)$$

Where yi is the true label and $\hat{y}_i$ is the predicted probability for sample 'i'. After training, performance is evaluated using metrics focused on minority class prediction: precision.

3.3. Multilayer Perceptron in Minority Class Prediction

The Electrical Grid Stability Simulated Data consists of 12,000 samples, each containing 12 variables for different features of electrical grid stability, and a binary target variable indicating if the grid was stable (class 1) or unstable (class 2). The dataset is imbalanced with a greater number of stable samples, which complicates the accurate prediction of the minority class (unstable).

A Multilayer Perceptron (MLP) is a feed forward neural network made up of an input layer, one or more hidden layers, and an output layer. Each layer is comprised of neurons which are fully connected, applying a linear transformation followed by a non-linear activation function, which can be of the ReLU or sigmoid form. For a hidden layer III, the output is calculated as:

$$h^{(l)} = \sigma\left(w^l h^{(l-1)} + b^{(l)}\right) \quad (7)$$

Where h(l-1) denotes the input from the preceding layer (or input features for the first hidden layer), W(l) and b(l) stand for the weights and biases, and σ signifies an activation function.

The final output layer produces class probabilities using a softmax function:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (8)$$

Where zi refers to the linear combination of inputs to output neuron iii. Although MLPs can model complex non-linear relationships, they often demonstrate a bias towards majority classes, particularly when trained on an imbalanced data set. This results in poor predicted performance on the minority classes.

To address this concern, GA-based and evolutionary sampling techniques can be utilized to rebalance the data set prior to MLP training. In GA-Oversampling, a genetic algorithm is used to generate synthetic minority samples through its fitness function:

$$Fitness(x) = \alpha \cdot similarity\left(x, D_{minority}\right) + \beta \cdot Diversity(x) \quad (9)$$

The resulting balanced dataset is,

$$D_{balanced} = D_{majority} \cup D_{minority} \cup D_{synthetic}$$

GA-Under sampling selectively removes majority class samples to prevent bias, guided by a fitness function:

$$Fitness(S) = Accuracy\left(MLP\left(D_{minority} \cup S\right)\right) - |S| \quad (10)$$

Producing a balanced set:

$$D_{balanced} = D_{minority} \cup S_{best}$$

A-Hybrid uses both oversampling and undersampling to improve the representation of both classes, while cluster-guided GA preserves diversity by applying GA operations over clusters of samples for representation. Multi-Objective GA (MOGA/NSGA-II) optimally balances and preserves multiple objectives such as maximizing accuracy and minimizing bias, resulting in a Pareto-optimal balance dataset. Once a balanced dataset is generated, the MLP will be trained to minimize standard back propagation cross-entropy loss:

$$Loss = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \quad (11)$$

Where y_i is the true label and $\hat{y}_i$ is the predicted probability.

3.4. Transformer Encoder for Minority Class Prediction

Transformer Encoders are deep learning models originally intended for sequence modeling and applications in natural language processing. In contrast to recurrent networks, they use self-attention mechanisms to relate dependencies between all elements of an input sequence simultaneously. Because of this ability, they were designed to perform well with datasets where features relate to each other in complex non-local ways.

3.4.1. Transformer Encoder Architecture

A Transformer Encoder block consists of

Multi-Head Self-Attention (MHSA) : Computes attention scores between all input elements to capture feature interdependencies:

$$\text{Attention}(\text{Q}, \text{K}, \text{V}) = \text{softmax}\left(\frac{\text{QK}^{\text{T}}}{\sqrt{\text{d}_\text{k}}}\right)\text{V}$$

$$\text{MultiHead}(\text{Q}, \text{K}, \text{V}) = \text{Concat}(\text{head}_1 \ldots., \text{head}_\text{h})\text{W}^\text{o}$$

Where Q,K,V are query, key, and value matrices derived from input features, d_k is the dimensionality, h is the number of attention heads, and W^O is the output projection.

A fully connected network applied independently to each position:

$$FFN(x) = \max(0, W_1 + b_1)\, W_2 + b_2 \quad (12)$$

In residual connections and layer normalization, Added after both attention and feed-forward sub-layers to stabilize training:

$$x' = layerNorm(x + sublayer(x)) \quad (13)$$

Where '*SubLayer'* can be either the attention or feed forward operation.

3.4.2. Handling Imbalanced Data with GA-Based Sampling

Similar to conventional deep learning models, Transformers can be biased against majority classes in imbalanced datasets. Genetic Algorithm (GA)-based sampling strategies are beneficial in rebalance the dataset before training. They consist of:

GA-Oversampling : Produces synthetic samples for the minority class with a genetic algorithm whose fitness function takes into account similarity and diversity.

$$Fitness(x) = \alpha \cdot similarity\left(x, D_{minority}\right) + \beta \cdot Diversity(x) \quad (14)$$

GA-Undersampling : Identifies a subsample of the majority class that improves the performance of the model:

$$Fitness(S) = Accuracy\left(Transformer\left(D_{minority} \cup S\right)\right) - |S| \quad (15)$$

The GA-Hybrid strategy combines both the GA-Oversampling and GA-Undersampling approaches with the explicit goal to optimally balance the dataset in a single stage of preparation. In another approach, the cluster guided GA method applies GA operations within each clusters to maintain class diversity.

The Multi-Objective GA (MOGA or NSGA-II) method optimizes multiple goals at once, such as maximizing a minority class F1-score and minimizing the imbalance, allowing you to produce Pareto-optimal datasets for training the Transformer to test against. When it comes to Model Training and Evaluation, the balanced dataset is now prepared to use with the Transformer Encoder, which is then trained using standard cross-entropy loss:

$$Loss = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \quad (16)$$

Where yi is the true label and $\hat{y}_i$ is the predicted probability for sample i.

4. Proposed Methodology

4.1. Tab Net Algorithm

The goal is to enhance minority class prediction in a power grid dataset using the TabNet model, suffering from a hybrid Genetic Algorithm (GA-Hybrid) and Evolutionary Sampling.

Step 1 : Data Preprocessing

- Load the Dataset
  $D = \{(x_i, y_i)_{i=1}^{N}\}$

Where $x_i \in \mathbb{R}^d$ are feature vectors of the power grid, any $y_i \in \{0,1\}$ are class labels (normal, faulty).

- Apply normalization or standardization
  $\tilde{x_i} = \frac{x_i - \mu}{\sigma}$ (17)

Where μ and σ are the standard deviation of each feature.

Step 2 : Feature Masking and Attention

The TabNet model leverages sequential attention and feature selection at each decison step t.

Feature Transformer : Implement a learnable transformation F to obtain the processed features

$$h_i^t = F(x_i^t\,;\, \theta_F) \quad (18)$$

Attention Masking : Compute feature mask M^t for step t

$$M^t = Sparsemax(A(x_i^t\,;\, \theta_F)) \quad (19)$$

Where,

- A is the attention network,

- Sparsemax guarantees focus is given to the relevant features (sparse attention),
- x_i^t is the input to the step t after undergoing previous transformations,
- Masked features: i.e. masking, update feature for the current step

  $x_i^{t+1} = x_i \odot M^t$

  Where ⊙ denotes element wise multiplication.

Step 3 : Decision Step and Aggregation

Each decision step outputs a representationy_i^t contributing to the final prediction.

$y_i^t = G(\, h_i^t \odot M_i^t \,;\, \theta_G \,)$

Where G is a feed forward network producing step wise decision outputs. The final output aggregates decision over T steps

$\hat{y}_i = \sum_{t=1}^{T} y_i^t$

Step 4 : Sparse Regularization

To encourage selection of fewer features, add a sparsity penalty.

$L_{sparse} = \lambda \sum_{t=1}^{T} \|M^t\|_1$

Where λ is hyper parameter controlling sparsity, $\|M^t\|_1$ is the L1 norm of the mask vector.

Step 5 : Training with GA-Hybrid Sampling. To handle minority class imbalance:

- Apply Genetic Algorithm-based Sampling (GA-Hybrid) to generate a balanced training dataset D′ by:
  1. Oversampling minority class
  2. Under sampling majority class
  3. Hybrid or cluster-based selection
- Train TabNet on D′ using cross-entropy loss for binary classification

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

  Combine the sparse regularization

  $L_{total} = L_{CE} + L_{sparse}$

Step 6 : Model Optimization

Optimize parameter

$\theta = \{\theta_F, \theta_A, \theta_G\}$ via

Adam optimizer

$\theta \leftarrow \theta - \eta \nabla_\theta L_{total}$

η is the learning rate.

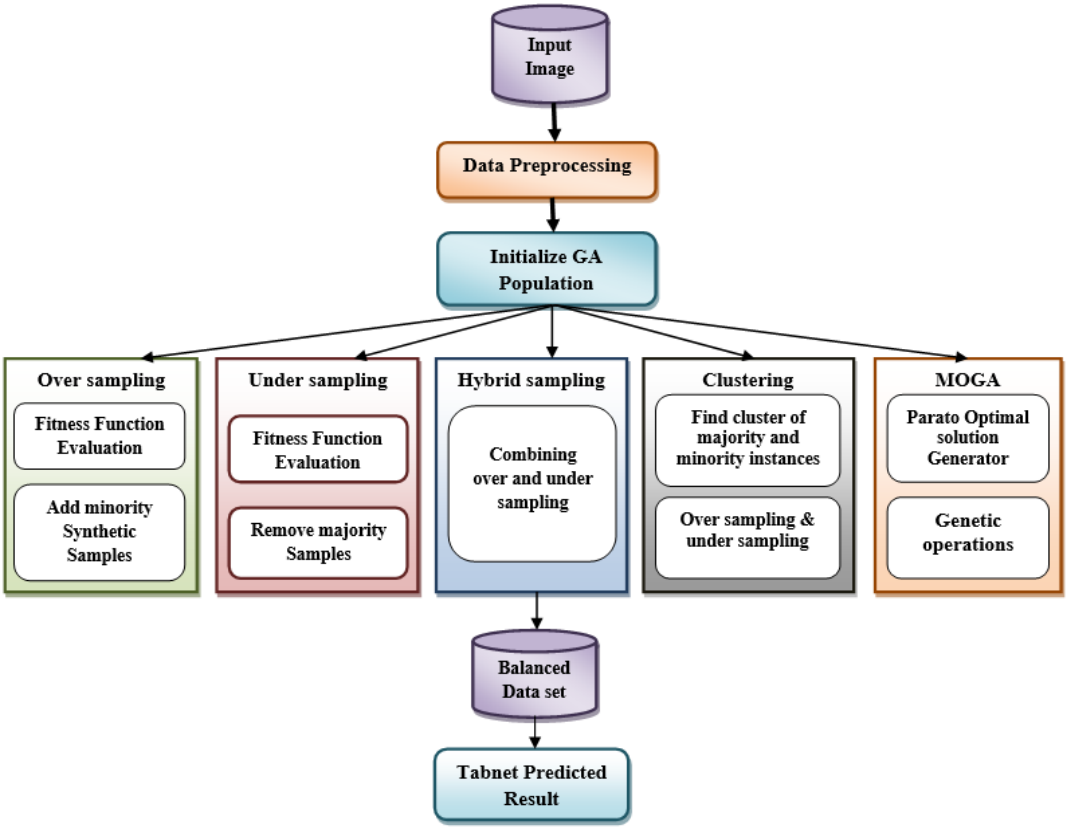

Fig. 1 Architecture of proposed methodology

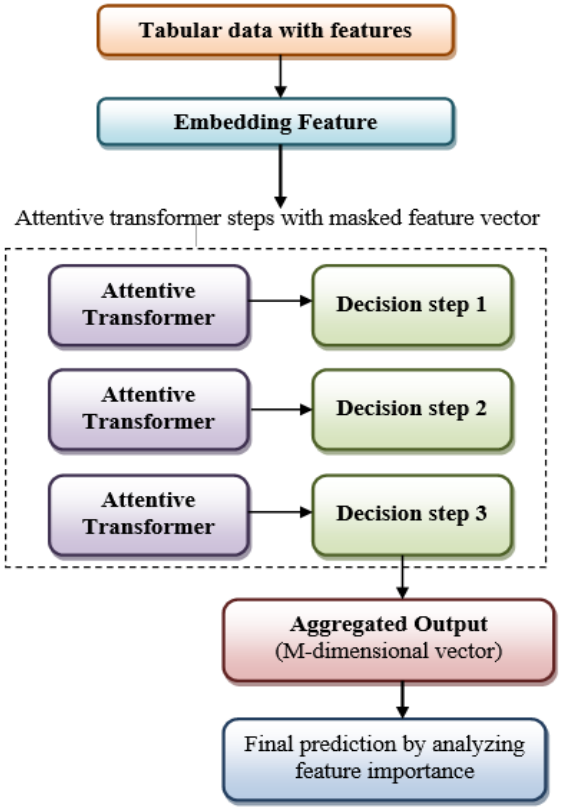

Fig. 2 Tabnet prediction process

Step 7 : Perform Prediction

For each test instance x_i, compute the final probability

$$\hat{y}_i = \sigma \sum_{t=1}^{T} y_i^t$$

Assign class label

$$y_i^{pred} = \begin{cases} 1, & \hat{y}_i \geq 0.5 \\ 0, & \hat{y}_i < 0.5 \end{cases}$$

Step 8 : Feature Importance Interpretation

TabNet naturally outputs feature importance scores

$$I_j = \sum_{t=1}^{T} \frac{1}{N} \sum_{i=1}^{N} M_j^t$$

Where I_j indicates how often feature 'j' was selected across all decision steps and instances, and helps to identify critical power grid parameters influencing minority faults. The architecture of the proposed method is shown in Figure 1.

5. Experimental Results

Table 1. Dataset feature and description from UCI machine learning repository

S.No.	Features	Data Type	Description
1	τ (tau)	Continuous	Time constant of the generator system (affects damping behavior)
2	p (power)	Continuous	Power generated (per unit)
3	g (gain)	Continuous	Gain of the control system
4	θ (theta)	Continuous	Power angle (phase difference between generator and reference)
5	δ (delta)	Continuous	Load angle, represents rotor position stability
6	V (voltage)	Continuous	Terminal voltage of the generator (per unit)
7	ω (omega)	Continuous	Rotor angular speed deviation from synchronous speed
8	Pmax	Continuous	Maximum generator power limit (system constraint)
9	K (stiffness factor)	Continuous	Synchronizing coefficient or stiffness of the machine-grid coupling
10	D (damping factor)	Continuous	Damping coefficient, resists oscillations
11	M (inertia)	Continuous	Inertia constant of the generator system
12	Q (reactive power)	Continuous	Reactive power (per unit)
	Stability Status	Nominal (Binary)	Target variable : 0 = Stable, 1 = Unstable

Table 2. Performance comparison of various sampling techniques with classifiers

Sampler	TabNet		MLP (3-layer)		CNN (1D)		LSTM		Transformer Encoder	
	Acc.	F1 Score	Acc.	F1 Score	Acc.	F1 Score	Acc.	F1 Score	Acc.	F1 Score
No sampling (baseline)	0.91	0.18	0.89	0.21	0.90	0.19	0.88	0.22	0.89	0.20
SMOTE	0.88	0.66	0.87	0.64	0.87	0.63	0.88	0.65	0.87	0.64
GA-Undersample	0.86	0.69	0.85	0.67	0.86	0.68	0.86	0.69	0.85	0.68
GA-Oversample	0.89	0.74	0.88	0.72	0.88	0.71	0.89	0.73	0.88	0.72
GA-Hybrid (Over+Under)	0.91	0.79	0.89	0.77	0.90	0.76	0.89	0.75	0.90	0.77
Cluster-Guided GA	0.90	0.76	0.88	0.74	0.89	0.74	0.90	0.75	0.89	0.75
MOGA (NSGA-II) Sampling	0.90	0.75	0.88	0.73	0.89	0.74	0.90	0.75	0.89	0.74

An important complication is related to the imbalanced nature of the data: approximately 95% of samples are stable, while only 5% are unstable. This imbalance allows regular models to achieve inflated accuracy by neglecting to predict minority unstable cases, which is an important consideration for monitoring reliability and safety of the electrical grid.

Table 2 presents the results obtained using different deep learning models (TabNet, MLP, CNN, LSTM, and Transformer Encoder) under various sampling strategies. The baseline models without sampling achieved accuracy values around 0.88 - 0.91. However, the corresponding F1-scores were extremely poor (0.18 - 0.22), indicating that the models predominantly classified instances as "stable," thereby neglecting the minority unstable class. This observation underscores the inadequacy of accuracy as a standalone performance metric when dealing with imbalanced datasets.

The application of SMOTE oversampling substantially improved the F1-scores to values in the range of 0.63-0.66, although the overall accuracy slightly decreased. This outcome suggests that synthetic sample generation mitigates imbalance by forcing the models to recognize unstable cases, but the generated samples may not fully capture the underlying distribution, leading to minor trade-offs in accuracy. Similarly, GA-based undersampling increased the F1-Scores further (≈0.67 - 0.69), as reducing majority class dominance provided better minority class recognition. Nevertheless, accuracy suffered marginally due to the removal of valuable majority class information.GA-based oversampling offered a stronger balance between precision and recall, yielding F1-scores in the range of 0.71 - 0.74 with accuracy around 0.88 - 0.89. The best overall performance, however, was achieved by the GA-Hybrid sampling approach, which integrates both oversampling and under sampling.

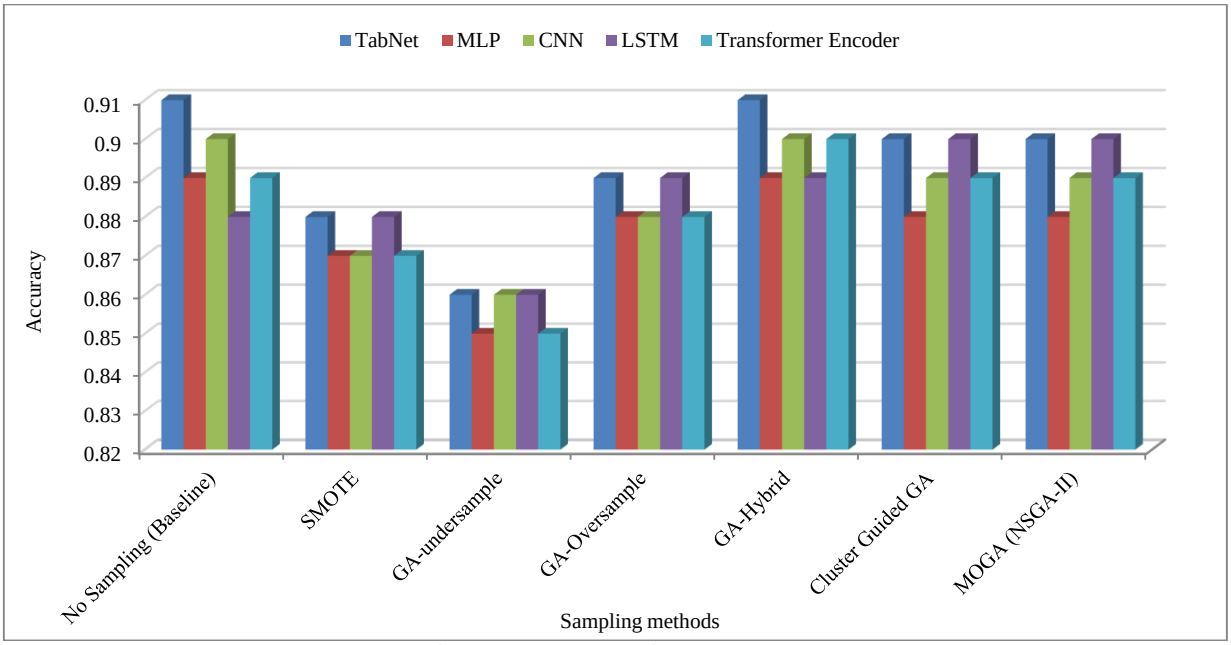

Fig. 3 Performance comparison of various sampling models

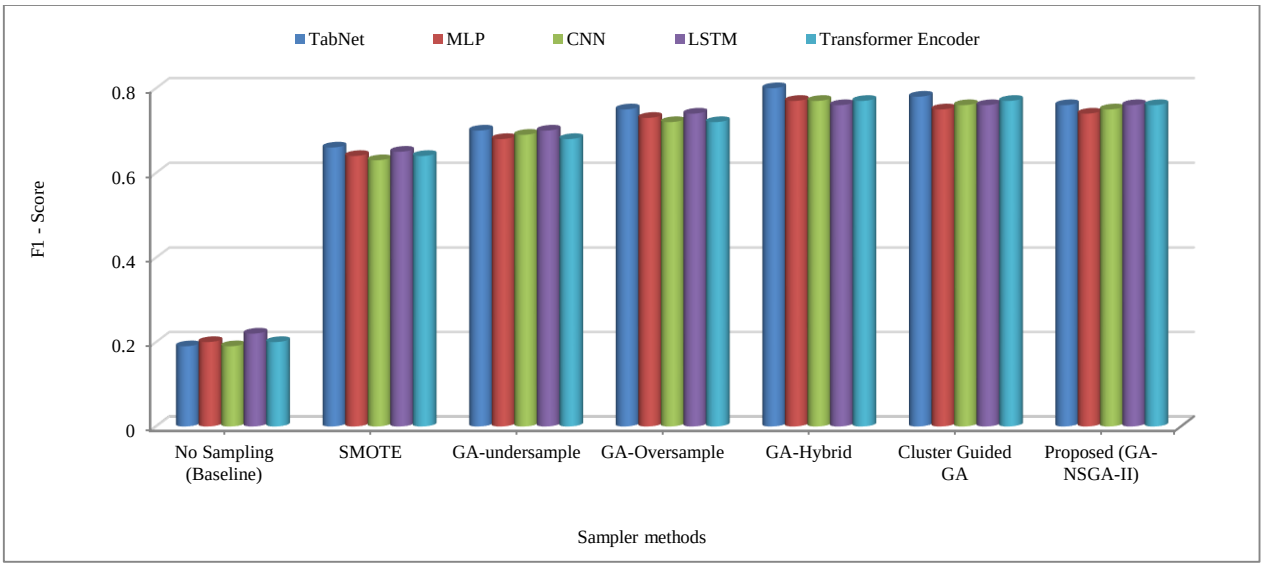

Fig. 4 Performance measure of F1-Score for various sampler models

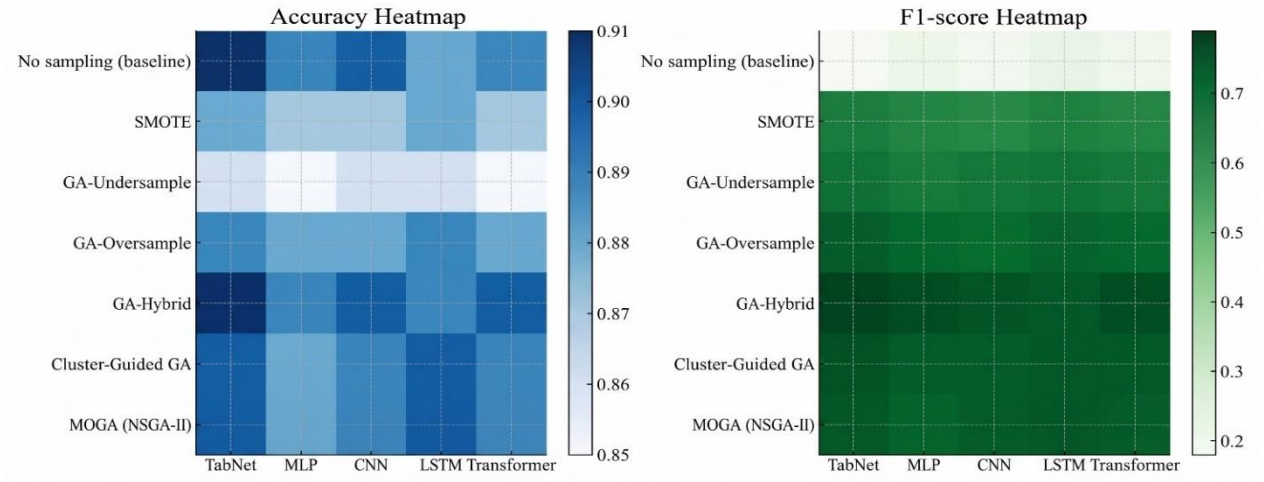

Fig. 5 Accuracy (left) and F1-Score (right) heat maps comparing sampling methods across models

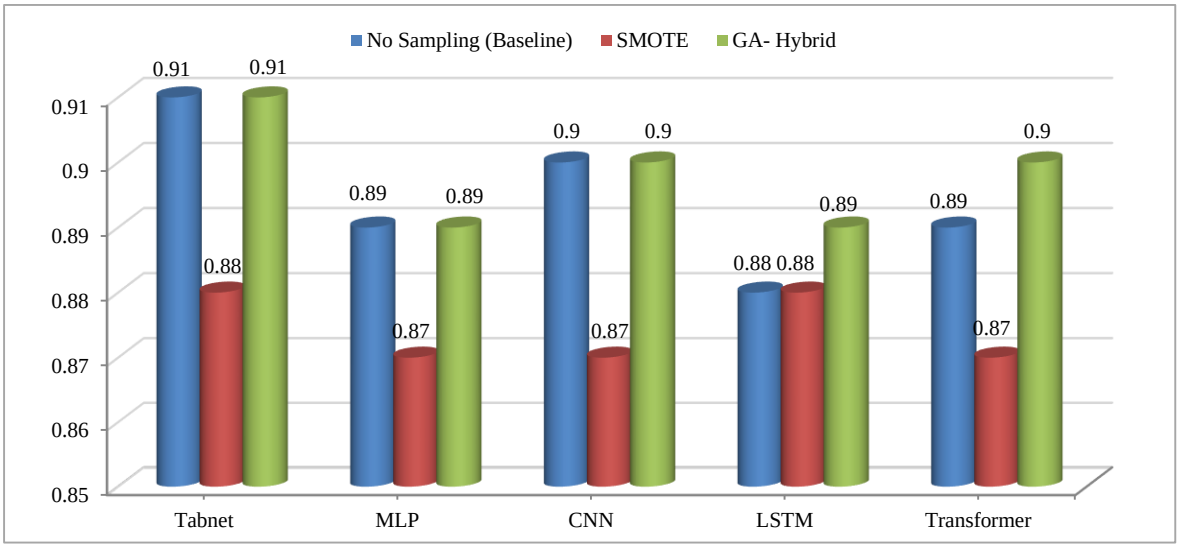

Fig. 6 Accuracy comparison of baseline & SMOTE and GA-Hybrid sampling approaches

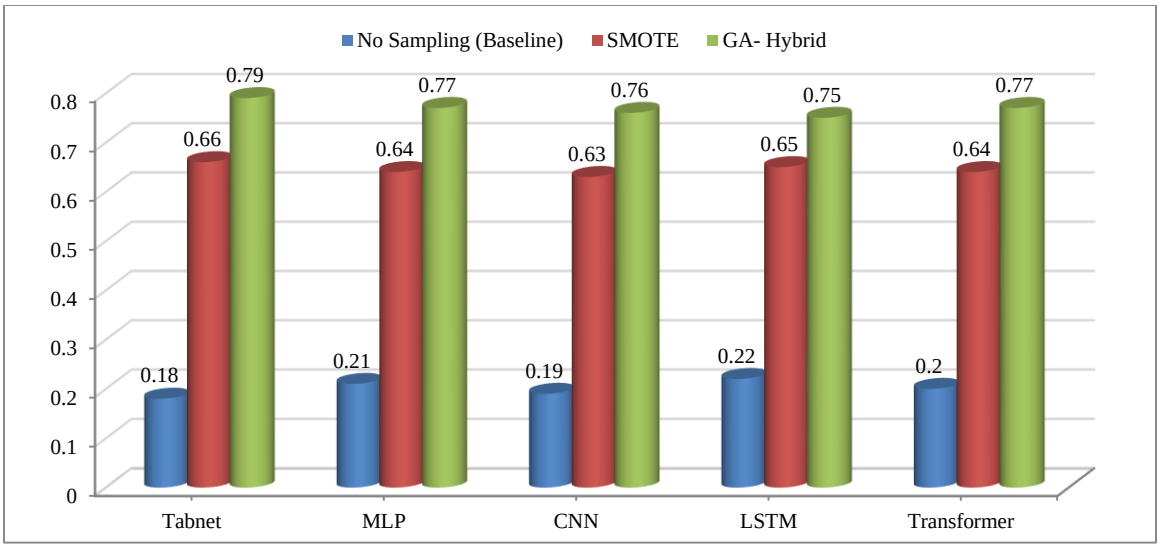

Fig. 7 F1-Score comparison of baseline & SMOTE and GA-Hybrid sampling approaches

The experimental results show that intelligent sampling strategies are essential for imbalanced classification in grid stability prediction. While baseline models appeared strong in terms of accuracy, they were ineffective at capturing the critical unstable states. Genetic algorithm-based hybrid sampling emerged as the most effective solution, providing a robust trade-off between accuracy and F1-score across all evaluated models. These findings emphasize that future predictive frameworks for smart grids must prioritize sampling strategies that enhance minority class recognition to ensure the reliability and security of modern power systems.

Figure 6 compares accuracy for Baseline, SMOTE, and GA-Hybrid across all models. Figure 7 denotes the comparison of F1-scores for the same three strategies. These figures clearly shows the baseline models have high accuracy but very low F1 (poor minority detection). SMOTE improves F1 but slightly reduces accuracy.

6. Conclusion

This work investigates the impact of different Genetic Algorithm (GA) - based strategies on feature selection using the Electrical Grid Stability Simulated dataset, with TabNet employed as the primary learning model and its performance compared against MLP, CNN, LSTM, and Transformer Encoder architectures. Experimental results, evaluated using accuracy and F1 score metrics, indicate that TabNet achieves superior performance when integrated with GA-driven feature selection in comparison to other deep learning models. The study primarily emphasizes GA-based oversampling, undersampling, and hybrid cluster-guided sampling techniques within the feature selection process, demonstrating that TabNet exhibits comparatively lower performance in the absence of these sampling strategies. As a future direction, this work can be extended by incorporating reinforcement learning-based approaches to further enhance feature selection and model performance.

References

[1] Suja A. Alex, J. Jesu Vedha Nayahi, and Sanaa Kaddoura, "Deep Convolutional Neural Networks with Genetic Algorithm-based Synthetic Minority Over-Sampling Technique for Improved Imbalanced Data Classification," *Applied Soft Computing*, vol. 156, pp. 1-16, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Jihyun Ha, and Jong-Seok Lee, "A New Under-Sampling Method using Genetic Algorithm for Imbalanced Data Classification," *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, Association for Computing Machinery, New York, NY, United States, pp. 1-6, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[3] Juan Gong, "A Novel Oversampling Technique for Imbalanced Learning based on SMOTE and Genetic Algorithm," *International Conference on Neural Information Processing*, Sanur, Bali, Indonesia, vol. 13110, pp. 201-212, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Iwo Naglik, and Mateusz Lango, "GMMSampling: A New Model-based, Data Difficulty-Driven Resampling Method for Multi-Class Imbalanced Data," *Machine Learning*, vol. 113, no. 8, pp. 5183-5202, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Wuxing Chen et al., "A Survey on Imbalanced Learning: Latest Research, Applications and Future Directions," *Artificial Intelligence Review*, vol. 57, no. 6, pp. 1-51, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Mabrouka Salmi et al., "Handling Imbalanced Medical Datasets: Review of a Decade of Research," *Artificial Intelligence Review*, vol. 57, no. 10, pp. 1-57, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Surani Matharaarachchi, Mike Domaratzki, and Saman Muthukumarana, "Enhancing SMOTE for Imbalanced Data with Abnormal Minority Instances," *Machine Learning with Applications*, vol. 18, pp. 1-31, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Ying Li et al., "An Improved SMOTE Algorithm for Enhanced Imbalanced Data Classification by Expanding Sample Generation Space," *Scientific Reports*, vol. 15, no. 1, pp. 1-21, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ping Gong, Junguang Gao, and Li Wang, "A Hybrid Evolutionary Under-Sampling Method for Handling the Class Imbalance Problem with Overlap in Credit Classification," *Journal of Systems Science and Systems Engineering*, vol. 31, no. 6, pp. 728-752, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Nicolás García-Pedrajas, José M. Cuevas-Muñoz, and Aida de Haro-García, "Evolutionary Simultaneous Under and Oversampling of Instances for Dealing with Class-Imbalance Datasets in Multilabel Problems," *Applied Soft Computing*, vol. 159, pp. 1-21, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Ehsan Nazari, and Paula Branco, "On Oversampling via Generative Adversarial Networks Under Different Data Difficulty Factors," *Proceedings of the Third International Workshop on Learning with Imbalanced Domains: Theory and Applications*, vol. 154, pp. 76-89, 2021. [Google Scholar] [Publisher Link]

[12] Tingting Pan et al., "An Improved Generative Adversarial Network to Oversample Imbalanced Datasets," *Engineering Applications of Artificial Intelligence*, vol. 132, pp. 1-14, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Chandana Gouri Tekkali, and Karthika Natarajan, "An Advancement in Adasyn for Imbalanced Learning: An Application to Fraud Detection in Digital Transactions," *Journal of Intelligent and Fuzzy Systems*, vol. 46, no. 5-6, pp. 11381-11396, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Alaa Alaerjan et al., "Improvement of Smart Grid Stability based on Artificial Intelligence with Fusion Methods," *Symmetry*, vol. 16, no. 4, pp. 1-21, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Paulo Breviglieri, Türkücan Erdem, and Süleyman Eken, "Predicting Smart Grid Stability with Optimized Deep Models," *SN Computer Science*, vol. 2, no. 2, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Teddy Lazebnik, "Pulling the Carpet Below the Learner's Feet: Genetic Algorithm to Learn Ensemble Machine Learning Model During Concept Drift," *Engineering Applications of Artificial Intelligence*, vol. 152, pp. 1-12, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[17] Khadijeh Nemati et al., "New Hybrid Feature Selection Approaches based on ANN and Novel Sparsity Norm," *Journal of Electrical and Computer Engineering*, vol. 2024, no. 1, pp. 1-16, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Vandad Imani et al., "Multi-Objective Genetic Algorithm for Multi-View Feature Selection," *Applied Soft Computing*, vol. 167, pp. 1-17, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Guilian Feng, "Feature Selection Algorithm based on Optimized Genetic Algorithm and the Application in High-Dimensional Data Processing," *PLOS One*, vol. 19, no. 5, pp. 1-24, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[20] Miguel Carvalho, Armando J. Pinho, and Susana Brás, "Resampling Approaches to Handle Class Imbalance: A Review from a Data Perspective," *Journal of Big Data*, vol. 12, no. 1, pp. 1-58, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[21] Hossein Nematzadeh et al., "Distance-based Mutual Congestion Feature Selection with Genetic Algorithm for High-Dimensional Medical Datasets," *Neural Computing and Applications*, vol. 37, no. 8, pp. 6217-6232, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[22] Behrouz Ahadzadeh et al., "SFE: A Simple, Fast, and Efficient Feature Selection Algorithm for High-Dimensional Data," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 6, pp. 1896-1911, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Mohammed Ghaith Altarabichi et al., "Fast Genetic Algorithm for Feature Selection-A Qualitative Approximation Approach," *Expert Systems with Applications*, vol. 211, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Fatemeh Amini, and Guiping Hu, "A Two-Layer Feature Selection Method using Genetic Algorithm and Elastic Net," *Expert Systems with Applications*, vol. 166, pp. 1-10, 2021. [CrossRef] [Google Scholar] [Publisher Link]