Enhancing Classification of Imbalanced Data using Diffusion Process

Vikram Velankar, Sachin S. Patil

***Abstract*—Conventional deep learning algorithms have difficulties when imbalanced data is encountered where one class disproportionately outnumbers another. This work investigates how a novel deep learning approach called diffusion processes could enhance classification performance on unbalanced datasets. We study how well diffusion models can produce artificial information for underrepresented group, equalizing the spread of categories and reducing innate biases in favour of the majority class. The study also looks at how various diffusion model topologies and training approaches affect classification accuracy, with a focus on minority classes. Furthermore, the study intends to evaluate diffusion-based approaches against other cutting-edge strategies for managing unbalanced data. The results of this study should help build more reliable and accurate Deep Learning models for practical uses, as well as offer insightful information on how well diffusion processes work for the generation of unbalanced data.**

***Index Terms*—Diffusion Process, Deep Learning, Unbalanced Datasets, Balancing, Generative AI**

I. Introduction

WITH the increasing availability of large datasets, machine learning algorithms have become an indispensable tool for solving complex problems in various domains such as healthcare, finance, and security [1]. Yet, the effectiveness of these algorithms is significantly influenced by the caliber of the data utilized in their training. In numerous practical scenarios, the data frequently exhibits an imbalance, indicating that the quantity of instances within each category is uneven. Imbalanced data poses a considerable challenge as it can substantially impact the effectiveness of machine learning algorithms, resulting in biased and erroneous predictions [1]. To tackle this concern, scientists have devised diverse methods to manage uneven data, including over-sampling, under-sampling, and cost-aware learning approaches. However, these techniques have their own limitations and may not always be effective in improving the performance of classifiers [2-12].
Usage of Generative Adversarial Networks (GANs) for similar purposes provides acceptable results compared to traditional techniques to deal with the imbalanced dataset [1]. Though GANs are promising towards generative capabilities they pose some issues too[13]. The issue related to training of GANs and even the mode collapse that inhibits the actual generative capabilities .
In recent years, diffusion-driven methodologies have emerged as a promising strategy for bolstering classifier performance on datasets characterized by imbalances. Integration of Generative Adversarial Networks (GANs) has demonstrated to enhance both accuracy and ease of training within the diffusion process [14]. This article aims to provide a comprehensive analysis of imbalanced data, its ramifications on machine learning algorithms, and how diffusion-based methodologies can be instrumental in addressing this formidable challenge[15].
This article endeavors to offer insights into the utilization of diffusion-based methods, which have arisen as a promising strategy for refining classifier performance on imbalanced datasets. These techniques harness the intrinsic structure of the data and interconnections among data points to enhance classifier classification capabilities. The suggested approach entails employing the diffusion process as part of Diffusion-Based Oversampling.
Diffusion-based oversampling: Diffusion-driven oversampling is a technique centered on producing artificial instances associated with the minority group by employing the diffusion procedure. This approach aims to rectify discrepancies among classes within the dataset and enhance the recall precision of the underrepresented category.
To tackle the challenge of imbalanced datasets, a tailored framework is employed where the minority class is assigned to the Diffuser component. The Diffuser meticulously examines the dataset, absorbs its patterns, and generates new data that closely resembles the original. The quality of this synthesized data is validated through correlation analysis and heatmap distribution across the input set. As more instances of the minor class are incorporated, the dataset achieves improved balance. The newly created data is then inputted into the Classifier for classification purposes.

II. Literature Survey

Addressing the issues stemming from imbalanced datasets, where certain classes dominate others, is essential in data analysis. Data balancing involves employing different methods and algorithms to rectify this imbalance. This examination focuses on four primary categories of data balancing, delving into the latest algorithms within each category and assessing their efficacy in handling class imbalance across various fields of application [1-12].

A. Data Level Imabalancing

One prevalent approach to addressing imbalanced data involves manipulating the data itself to achieve a more balanced class distribution. These methodologies, grouped under the umbrella of data-level balancing techniques, seek to alter the initial dataset in such a manner that diminishes

Vikram Velankar from Department of CSE from RIT, Islampur, Maharashtra, India, 415414. email: vikram9623463690@gmail.com
Sachin Patil from Department of CSE from RIT, Islampur, Maharashtra, India, 415414. email: sachin.patil@ritindia.edu

the favoritism towards the majority class, enabling the model to glean more effectively from the samples of the minority class.

Implementing M-Centers, an undersampling tactic that generates data clusters equivalent to the no. of underrepresented groups (M) to counterbalance the influence of the Pervasive, enhanced the accuracy of categorizing imbalanced datasets[16].
An undersampling approach may be suggested by using a unique algorithm titled Condensed E nearest neighbours (Condensed-ENN). It classifies the whole set of data using ENN algorithm and then condenses the data points for additional passes, resulting in a cluster that can handle actual unbalanced data [17].
Synthetic Minority Oversampling technique (SMOTE) is employed to address dilemma of unbalanced data. Data augmentation being the major workforce for generation of synthetic data, that is used later for provision of minority samples to create sense of balanced dataset being closest to the real data [18].
Other SMOTE variants like Borderline SMOTE and Boundary SMOTE are other mechanisms to solve the same problem, where the boundary of generated data with respect to that of original data is taken into consideration [19, 20].
A significant improvement over SMOTE is Adaptive Synthetic Sampling (ADASYN), which takes into account data points that are challenging to learn. increases the variability for the creation of datasets across SMOTE [21]. In order to do this, ADASYN assigns weights to each data item based on how challenging it is to learn, uses an adaptive feature to fabricate a minute set of samples to mitigate the problem of overfitting..
Minority weighted Majority undersampling (MWMOTE) is a mix of best of both under and oversampling. Utilizes interpolation of data points with SMOTE generated synthetic data and real weighted datapoints [22]. MWMOTE is better as compared to SMOTE as it interpolates between synthetic data and real datapoint leading towards better complexity of synthetic data and helping in mitigating overfitting. Utilizes a 3 phase architecture in generating data samples, increasing complexity of generation at each phase and involving validation at 3rd phase.

In a manner akin to SMOTE or MWMOTE, Real-Value Negative Selection Oversampling (RSNO) utilizes an additional membership function for evaluating data. This supplementary function serves to enhance variability within the dataset, thereby addressing concerns related to overfitting [23].
SMOTE - E nearest neighbors (SMOTE-ENN) is an hybrid approach that actually utilizes SMOTE technique to generate Synthetic samples for oversampling and utilizes the ENN approach for removing the majority samples acting as an undersampling algorithm [24].
SMOTE - iterative Partitioning filter (SMOTEIPF) has been introduced to address the issue of misclassified boundary examples in data, which can negatively impact classifier performance [25]. The IPF technique is employed to filter out noisy samples, ultimately leading to an enhancement in classifier performance.

B. Error Function Based Balancing

Error-based balancing techniques present a potential avenue to overcome this limitation. By embedding information about class imbalance directly into the error function utilized during the training, favors the goal to alleviate the bias favoring the majority class and bolster the model's capability to glean valuable insights from the underrepresented data points. This approach strives to refine the model's learning dynamics, ensuring a more equitable treatment of all classes within the dataset and thereby fostering improved predictive performance across the board [26, 27].

Cost-based Rectification Loss (CRL) an activation function devised specially for imbalanced datasets, is utilised to mitigate the problem of imbalanced datasets in Images [26]. CRL utilises a major matrix comparative analysis that compares the randomly selected negative pair and positive pair. It checks for the actual distribution of both elements and tries to balance the same in a non-deterministic manner.
Focal loss another loss function devised specifically for handling the imbalanced datasets, utilizes a confidence value of each point Pt and decreases the same if Pt belongs to a major class to balance out the overall dataset on its confidence level resulting in a balanced learning dataset [27].

C. Classifier-Design based Balancing

An alternative innovative strategy to tackle this constraint: classifier design-oriented balancing. In contrast to conventional techniques altering data directly, this method centers on adapting the classifier structure to explicitly integrate imbalanced data insights during training. This entails a more thorough exploration into the blueprint of the suggested classifier design, elucidating its utilization of mechanisms while confronting the distinct hurdles of imbalanced data categorization[28, 29].

AUC is generally considered relatively insensitive to class imbalance. This means that even in datasets with a skewed class distribution, the AUC score might not be significantly affected. In contrast, other metrics like accuracy can become misleading due to the dominance of the majority class [28]. While not directly influenced by class imbalance, AUC can fail to capture the full picture of model performance in such scenarios. It primarily focuses on the ranking ability of the model, regardless of the specific class labels.
Kernelized Online Imbalanced Learning (KOIL) is another mitigation technique devised for the problem of Imbalanced Datasets. It is an architecture model with two buffer writers and one kernel that generates a penalty for each data point. Later this penalty is evaluated as a part of an adversarial heuristic function [29].

D. GANs Based Balancing

The potential of Generative Adversarial Networks (GANs) as a novel balancing strategy. By leveraging the synthetic sample generation capabilities of GANs. The theoretical foundation and implementation strategies of this GAN-based balancing approach, highlighting its potential to improve classification performance on all classes [1, 30].

Triple-GANs (TGAN) is a three character architecture implemented to solve the problem of imbalanced dataset using oversampling technique. It uses an adversarial relationship between classifier (a), generator (g) and discriminator (d) [30]. Though the generation process is benefited but the classification is unable to learn the things generated due to adversarial relationship.
This was solved by the introduction of Triple-Cooperative GAN (TCGAN). It promoted the same three player architecture during the generation process, but changed the adversarial relationship between classifier and generator to cooperative relationship [1]. Usage of cooperative relationship for the training process of generator and classifier enhances the actual usage of GAN for solution of problem and gives notable results.

E. Diffusion Based Balancing

In this paper we are proposing a methodology for balancing the dataset using the generative capabilities of Diffusion Process. Though the process is quite unexplored there are immense possibilities the process can be utilized for, one being tabular data generation [32,33].
Tabular- Denoising Diffusion Probabilitstic Model (Tab-DDPM) utilizes same diffusion model to predict the noise value to actual meaningful value. Though having greater generative capabilities, noisy output is inevitable leading to loss of accuracy [32].
The following problem was coped up by method proposed in this article, that uses technique of Diffusion by Latent calculation, that generates more meaningful outputs that can be utilized to actually improve performance metric accuracy for the diffusion model [33].

III. Methodology

Imbalanced datasets, where one class significantly outweighs others, pose a significant challenge for diffusion processes in classification tasks. The skewed distribution leads models to prioritize the majority class, neglecting the minority class, impacting overall performance. This section explores a novel tabular diffusion-based balancing method specifically designed to address this limitation within the diffusion process framework. This approach leverages the unique properties of tabular data and the power of diffusion processes to craft an equalized representation for training the classifier.
The novel strategy incorporates three layers, one being the actual diffusion process for the generation of tabular data, second being the quality of generated data compared to original and third being the actual implementation of classifier based model.
Fig.1 represents the complete work flow of the actual process that is being utilised. First some tabular data is generated using diffusion process later, the data is taken to visualization to actually map the distribution compared to original data. The accepted distribution is later passed on to classifier along with the original data to predict the classes.

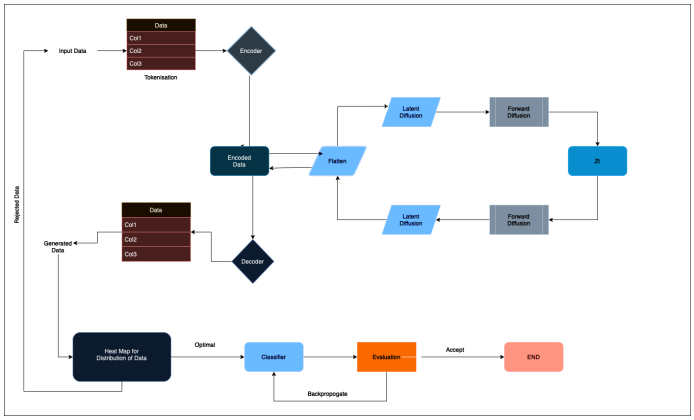

Fig. 1. This figure represents the Flow Diagram for complete process i.e. Tabular Diffusion followed by Visualisation followed by Classificaton

A. Diffusion Model

Tabular Synthetic generating Diffusion Process (Tab-Syn) provides a base for this research [34]. Fig.2 represents the general model architecture proposed [34] that has proven to be better in generating tabular samples. Works on principle of Diffusion process [14] that involves two-step learning process before it is capable of generating non existing samples.

1) Diffusion problem: The problem involves a data with set of features represented as $Feature_{num}$ and set of categorical features represented as $Feature_{cat}$. The total features that represent categorical features $Category_i$ where i = 0 ..... 1 and each category $Category_1$ ..... $Category_i$, represents different set of attributes. Collective categorical and numerical features define a set of features for an observation **a** = $[Feature^{num} + Feature^{cat}]$, and this paper represents the non-conditional generating solution for the particular set of tabular features.
The total tabular dataset $\Gamma = \{x + y\}$ is what is utilized for the generation of new data points for balancing problems.
2) Generation using Diffusion Process: Diffusion models are Markov chains that are utilized in a two-step procedure to generate usable data. Forward process has only one task of addition of random noise to the actual data that is to be generated using $f(a_T|a_0) = \prod_{t=1}^{T} f(a_t|a_{t-1})$, where $f(a_0)$ add samples random noise to the first element of the data distribution a_0. And the same is true when the function sample the noise from the predefined distribution of latent variables $f(a_t|a_{t-1})$ with variances **V** = $\{\alpha_1....\alpha_T\}$ for all T distributions[32].
On the other hand reverse diffusion process uses the log-likelihood of the sampling function to determine the actual

value of noise for a particular noisy sample belonging to a class required [33].
$r(a_{0:T}) = \prod_{t=1}^{T} r(a_{t-1}|a_t)$ is the required reverse process that is utilized to generate new high-quality data samples, for each latent variable $a_T \sim f(a_T)$.

$$\log f(a_0) \geq E_{f(a_0)}[\log d_\theta(a_0|a_1)] - D_{KL}(f(a_T|a_0)\ f(a_T)) - \sum_{t=2}^{T} D_{KL}(f(a_{t-1}|a_t, a_0)\ d_\theta(a_{t-1}|a_t)) \quad (1)$$

where D_{KL} is the KL divergence value for the distribution generated and the original existing.
The same can be utilized to get a gradient for reverse process using the Eq.2 mentioned for prediction of noisy input to actual values represented as $\mathbf{dz_t}$.

$$\mathbf{dz_t} = -2\sigma^{'}(t)\sigma(t)\nabla_{z_t} log\mathbf{d}(z_t)dt + \sqrt{2\sigma^{'}(t)\sigma(t)}\mathbf{dw_t} \quad (2)$$

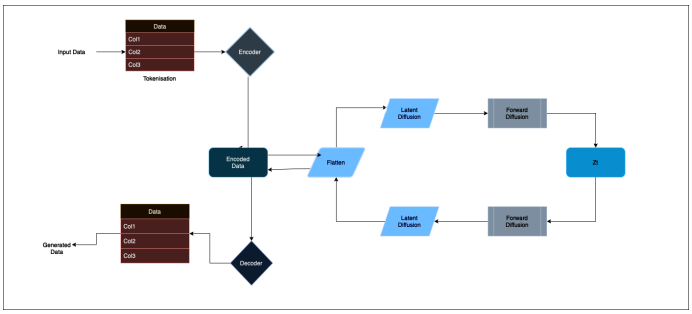

Fig. 2. Figure represents the actual working of the diffusion process for generation of new featured datapoints. Utilizes the Tabular Synthetic data generation process [33], for generation of new data points with better accuracy.

B. Algorithm for Generation process

1) TabSyn-Algorithm: Representation of a diffusion model algorithm that has capability to generate tabular data in its most complex form [33]. It also has the capability to generate multivariate data sequences with at most accuracy for the distribution. Divided into the following elements:

a) Variational AutoEncoder (VAE)
The capacity to extract significant patterns from intricate data stands as a fundamental aspect of contemporary machine learning. Variational Autoencoders (VAEs) have surfaced as an influential instrument in attaining this objective, presenting a distinctive fusion of reducing dimensionality and generating models. This overview delves into the fundamental principles of VAEs, emphasizing their benefits and utilization across diverse research fields.
VAE consists of two elements:
1.Encoder: Input Data is evolved onto a latent space with reduced dimensions, capturing the fundamental characteristics.
2.Decoder: Recreates the initial data from the latent representation, guaranteeing that the acquired features convey meaningful information.

A training algorithm for VAE for the particular solution is represented by Algo.1.
VAE requires two variables during training phase μ and $\log\sigma$ representing mean and log variance of dimension L.

$$\mu = Hidden_{pooled} * Weight_\mu \quad (3)$$

$$\log\sigma = Hiddem_{pooled} * Weight_\sigma \quad (4)$$

where, $Weight_\mu,\ Weight_\sigma \in R^L$ and $Hidden_{pooled} \in R^H$, where H is the Hidden vector of the encoder with mean pooling.

Decoder equation gives out decoded feature $\hat{d}$ using set of features $a = [Feature^{num}, Feature^{cat}]$

$$\hat{d}_i^{num} = \hat{Weight}_i^{num} * Feature_i^{num} + b_i^{num}$$

$$\hat{d}_i^{cat} = Softmax(\hat{Weight}_i^{cat} * Feature_i^{cat} + b_i^{cat})$$

$$\hat{d} = [\hat{d}_1^{num}, ..., \hat{d}_{M_{num}}^{num}, \hat{d}_1^{cat}, ..., \hat{d}_{M_{num}}^{cat}] \quad (5)$$

where $\hat{Weight}_i^{num},\ \hat{b}_i^{num} \in \mathbf{R^{1x1}}$
$\hat{Weight}_i^{cat},\ \hat{b}_i^{cat} \in \mathbf{R^{1x}}cat_i$ are detokenizing parameters.
where $M_{num} \in \mathbf{R}^{num}$ i.e. Number of Elements

Algorithm 1 An algorithm for Tabular Data generation[33] : Training Variational Auto Encoder (VAE)

1: Sample $a = (Feature^{num}, Feature^{cat}) \sim f(\Gamma)$
2: Get Token values for $Token^{cat} \rightarrow Feature_{cat}^{one\ hot}.W^{cat} + b^{cat}$
3: Get μ *and* σ using Eqns 3 and 4.
4: Reparameterize: $\hat{z} = \mu + \epsilon * \sigma$ where $\epsilon \in \mathbf{N}(0, I)$
5: Get $\hat{e}$ via VAEs Decoder
6: Get detokenized features $\hat{d}$ using Eq.5
7: Calculate Loss $\mathbf{L} = l\ (z,\ \hat{d}) + \beta_{KL}(\mu,\ \sigma)$
8: Update parameters using Adam
9: **If**: $\mathbf{L}$ fails to decrease in steps $\mathbf{M} \rightarrow \beta_{new} = \lambda * \beta$
10: **End**

Algo.1 represents the working of encoder and decoder that supplies information to the diffusion process which utilizes latent diffusion for generation process. TabSyn has demonstrated superiority over other cutting-edge methods in producing exceedingly intricate tabular data [33]. A different algorithm can be employed to represent this, which extends the approach used for generating tabular data. Algo.2 provides insights for the generation process of a tabular data.

Algorithm 2 An algorithm for Tabular Data generation[33] : Training Diffusion Process

1: Extract the vector a_0 from $\mathbf{f}(a_0) = \mathbf{f}(\mu)$
2: Extract timed instances t from $\mathbf{f}(t)$ to retrieve $\sigma(t)$
3: Extracting the noise(vect) retrieved from $\sim \mathbf{I}(0,\ \sigma_i^2)$
4: Retrieve Bartered vectors $\mathbf{z_t} = \mathbf{z_0} + \epsilon$
5: Calculating error using $l\ (\theta)\ = \| \epsilon_\theta(\mathbf{z_t}, \mathbf{t}) - \epsilon \|_2^2$
6: Update parameters using Adam
7: **End**

Using Algo.1 and Algo.2 process of generation can be initialized, though the sampling mechanism needs to be

developed. Algo.3 refers to the same sampling mechanism that utilizes the trained Diffusion framework and synthesized instances reflecting the distribution of the initially sampled data [33].

Algorithm 3 An algorithm for Tabular Data generation[33] : Sampling Mechanism

1: Retrieve $\mathbf{a_t} \sim \mathbf{I}(0,\ \sigma^2(TIME)\mathbf{I}),\ time_{max} = TIME$
2: **for** j = maximum, ... , 1 **do**
3: $\nabla_{b_{time_{j-1}}} \log \mathbf{f}(b_{time_j}) = -\epsilon_\theta(b_{time_j},\ time_j)/\sigma(time_j)$
4: Extract $b_{time_{j-1}}$ using Eq.2
5: **end for**
6: Put $\mathbf{b_0}$ as an input for VAEs decoder, and acquire $\hat{e}$.
7: Retrieve the unparsed vectors $\hat{d}$ using Eq.5
8: $\hat{d}$ is the vector that is generated.
9: **End**

The algorithms Algo.1 through Algo.3 represents the complete working of the diffusion model[33] that is utilized for actual generation of the samples in an imbalanced dataset.

C. Classifier

Achieving robust classification performance remains a challenge When handling datasets with disparate class distributions, where one class significantly outweighs others. Traditional methods often struggle to learn effective representations for the minority class, leading to biased models.
This section explores the different potential classifiers for classification of imbalanced datasets.

1) Logistic regression: Logistic Regression, a fundamental yet powerful classifier. It leverages a linear model in Eq.6 to establish The association between predictor variables (X) and the binary outcome variable (y), represented by:

$$\mathbf{h}(x) = \Gamma_0 + \Gamma_1 * a_1 + \Gamma_2 * a_2 + ... + \Gamma_n * a_n \quad (6)$$

Here, Γ_0 represents the bias term, and Γ_j (where j = 1 to number of elements) denotes the weight coefficients associated with each independent variable (X_i). The core of Logistic Regression lies in transforming these linear outputs (h(X)) into probabilities between 0 and 1 using the sigmoid function Eq.7:

$$\sigma(z) = 1/(1 + (e^{-z})) \quad (7)$$

Logistic Regression is better for Binary Classification and utilizes Binary Crossentropy Loss function for its improvement.

2) Decision Tree (DT): Decision Trees, a powerful non-parametric classifier known for their interpretability and robustness. Unlike Logistic Regression, Decision Trees do not rely on explicit equations. Instead, they follow a tree-like structure where internal nodes represent splitting criteria based on features (X) and terminal nodes represent class labels (y). At each internal node, the decision tree utilizes a splitting rule to separate the data based on a specific feature (X_i) that best discriminates between classes. This splitting rule can be formulated in various ways, with common options including Gini impurity Eq.8 for classification tasks:

$$Gini(t) = 1 - \sum_{i=1}^{n} (\mathbf{a}_j)^2 \quad (8)$$

3) Support-Vector-Machine (SVMs/SVs): SVs, A robust machine learning algorithm recognized for its ability to find optimal decision boundaries. Unlike Logistic Regression, SVMs don't rely on explicit probability equations. Instead, they focus on maximizing the margin between the separating hyperplane and the closest data points (support vectors) from each class.
Formally, the objective of an SVM can be expressed as maximizing the margin Eq.9:

$$\textbf{Maximize} = \| W \|^2 - 1 \quad (9)$$

where w represents the weight vector defining the hyperplane and $\| \cdot \|$ denotes the L2 norm.

However, for complex datasets, SVMs often employ kernel functions ($\phi(X)$) that map the input data (X) to a higher-dimensional space, allowing for non-linear separation. The decision function in this higher-dimensional space can be written as Eq.10:

$$\mathbf{f}(a) = Weight^T.\ \Phi(x) + \beta \quad (10)$$

4) Random Forest Classifier (RFs): While diffusion processes offer a promising avenue for data augmentation in imbalanced datasets, the choice of classifier remains crucial for exploiting the enriched data. This section explores the application of Random Forests, a powerful ensemble learning method known for its robustness to imbalanced data and interpretability. Unlike Logistic Regression, Random Forests don't rely on a single model or explicit equations. Instead, they combine multiple decision trees (often referred to as base learners) trained on different subsets of features and data points.

The final prediction of a Random Forest is typically made through majority voting Eq.11 across the individual predictions from each base learner:

$$arg.\max(\sum_{j=1}^{m} (v_j(a))) \quad (11)$$

where $v_j(a)$ represents the vote of the j-th base learner for a particular class given an input instance a.

5) Deep Neural Network (DNs): DNs have emerged as a frontrunner in this pursuit, demonstrating remarkable capabilities in recognizing complex patterns and achieving high classification accuracy.
DNN utilized for the study consisted of an Artificial Network layered into a model with optimizer Adam and loss function binary crossentropy with logits loss.

5.1) Architecture:

Fig.3 represents the architecture diagram for the Deep Neural Network utilized in the proposed study. The major

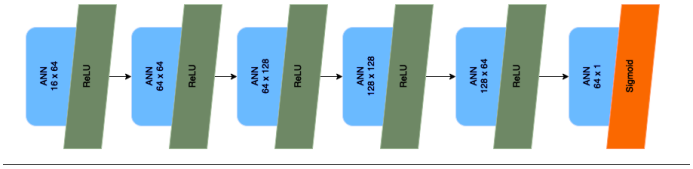

Fig. 3. The figure illustrates the actual architecture of the DNN model utilized in the study

parameters that were utilized can be found in Table.I:-

TABLE I
THIS TABLE REPRESENTS ALL THE HYPER PARAMETERS THAT WERE UTILIZED DURING STUDY FOR THE ENHANCEMENT

Attribute	Value
Learning Rate	0.01
Layers	6
Hidden Units	[64,128]
Regularization	L2
Optimizer	Adam
Loss Function	BCEWithLogitsLoss

The complete architecture Fig.1 represents the actual usage of the generated data directly for the classification process. The data generated is passed directly into the classifier to analyze the performance gap and the data is regenerated until performance is enhanced.

IV. RESULTS

A. Determination of Quality of Datasets:

1) Evaluation Metrics: 1.Baseline Statistics:-This step involves figuring out how spread out the data is in each column (density estimation) and how much each column influences the others (correlation)[33].

1.1) Column Density Estimation:-
1.1.1) Kolmogorov-Sirnov Test (KST):- Two distributions $D_o(a)$ *and* $D_s(a)$ where o represents original and s represents the synthetic version of the same dataset, finds the distance between the same using Cumulative Distribution Functions (CDFs) Eq.12:

$$KST = supremum_a \ \|CDF_o(a) - CDF_s(a)\| \tag{12}$$

where, $CDF_o(a)$ and $CDF_s(a)$ are CDFs of $D_o(a)$ and $D_s(a)$ given in Eq.13:

$$\mathbf{CDF}(a) = \int_{-\infty}^{a} \mathbf{f}(a)da \tag{13}$$

1.1.2) Total Variation Distance (TVD):- The computation of difference of the probabilities of the distribution groups that represent the original [O(.)] and synthetic [S(.)] data distributions given by Eq.14.

$$TVD = \frac{1}{2}\sum_{\omega \in \Omega} \|O(\omega) - S(\omega)\| \tag{14}$$

where ω explores every point from column Ω

1.2) Paiw-Wise Column Density Estimation:-
1.2.1) Pearson Correlation Coefficient:-
Metric to evaluate whether the data points are linearly correlated represented by Eq.15[33]:

$$\rho_{a,b} = \frac{Covar(a,b)}{SD_a.SD_b} \tag{15}$$

where a and b are sequentially moving columns. Covar is Co-variance and SD is Standard Deviation.

$$Pearson\ Score = \frac{1}{2}\mathbf{E}_{x,y}\|\rho^o_{x,y} - \rho^s_{x,y}\| \tag{16}$$

1.2.2) Contingency Similarity (CS):-
Calculated using Eq.14 for pair of categorical columns N, B summarized by Eq.17:

$$Contingency\ Score = \frac{1}{2}\sum_{a \in N}\sum_{b \in B} \|O_{a,\ b} - S_{a,\ b}\| \tag{17}$$

where $a,\ b$ represents everu category in N and B for Original (O) and Synthetic (S).

1.3) Specific Statistics:- These $\alpha - precision$ and $\beta - recall$ scores [34] help us assess how well synthetic data captures both the real data's accuracy (fidelity) and its variety (diversity).

1.4) Machine Learning Performance Utilization data fabricated by the process is essential to enhance a metric of the Machine Learning model for unbalanced datasets.

2) The Implemented Results: The results showcased in Tab.V are studies of over 20 datasets. That shows how TabSyn algorithm outperforms others in every aspect.

B. Usage for Enhancing Classification on Imbalanced Dataset

The usage of the TabSyn algorithm (Algo.3) for the study was done on Diabetes Dataset presented on Kaggle Datasets. The dataset was 35% balanced and and had imbalanced binary category provided in Table.II.

TABLE II
TABLE REPRESENTS THE ACTUAL DISTRIBUTION OF DATASET FOR BINARY CLASSIFICATION OF DIABETES DATASET

Category	Percentage
Non-Diabetic	65%
Diabetic	35%

1) Before Diffusion Process: The data from Table.II was passed into the classifiers from Methodlogy.c section.III-C Table.III depicts the results of the model before the utilization of the actual TabSyn for solving the issue.

The following observations were made before balancing the data.

TABLE III
THE TABLE REPRESENTS THE ACTUAL RESULTS AND THE CONFIGURATION FOR THE TEST BEFORE UTILIZATION OF DIFFUSION PROCESS ALGO.3.

Model	Accuracy	Configuration
Logistic Regression	75%	Max-Iterations = 1000
Decision Trees	80%	Max-Depth=5
SVM	76%	Kernel = Linear
Random Forest	77%	Max-Depth=2
DNN	train = 93%; test = 72%	epochs = 400

Implementing Diffusion process: The following data was passed into the Algo.3 for generation. The quality of data was verified using the metrics proposed in Results.Quality section.
Data generated wasn't just relied on the metrics provided but rather visualised as can be found out in Fig.4 and Fig.5.
The generated data was later concatenated using Pandas Dataframe with the original data to be passed into the classifier. The data merged followed the process that can be in the Fig.1.

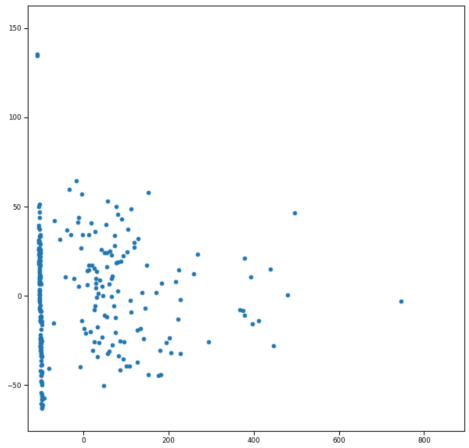

Fig. 4. This diagram depicts the factual spread of the authentic Diabetes dataset.

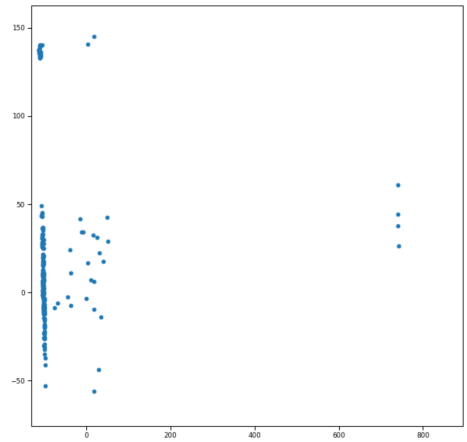

Fig. 5. This diagram depicts the factual spread of the artificial dataset that is fabricated using the Algo.3

The data then passed into the classifiers from Methodology.C section for testing the enhancements. The Table.IV represents the results from the actual test for the study.

TABLE IV
THE TABLE REPRESENTS THE ACTUAL RESULTS AND THE CONFIGURATION FOR THE TEST BEFORE UTILIZATION OF DIFFUSION PROCESS ALGO.3.

Model	Accuracy	Configuration
Logistic Regression	76%	Max-Iterations = 1000
Decision Trees	81%	Max-Depth=5
SVM	76%	Kernel = Linear
Random Forest	78%	Max-Depth=2
DNN	train = 100%; test = 83%	epochs = 400

V. FUTURE WORK

The future work for the proposed study includes usage of newer and better model architectures, that can be altered to check for other datasets. Even multimodal capabilities for generation of the imbalanced data to balance for different data type can be a next step forward. Automatic detection and utilization of proposed process can prove to be useful to for solving the proposed problem generally.

VI. CONCLUSION

This study set out on a quest to investigate the possibilities of diffusion mechanisms in addressing the difficulties linked with imbalanced classification assignments. The investigation centered on the efficacy of generating synthetic minority class samples using diffusion models. The proposed solution hypothesized that by enriching the dataset with these synthetic samples, there could be an improvement in the performance of various classification algorithms.

The exploration encompassed a range of established classifiers, including classifiers covered in the paper, e.g., DNs, RFs, SVs, etc. Through meticulous experimentation, It was observed by compelling evidence that diffusion-based data augmentation serves as a powerful tool for addressing the inherent bias present in imbalanced datasets. The incorporation of synthetic data demonstrably led to significant advancements in terms of overall classification accuracy. These findings highlight the effectiveness of diffusion processes in bolstering classification performance, particularly in scenarios where imbalanced data poses a significant obstacle.

Looking towards the future, this research opens doors for further exploration in several key areas. One promising avenue lies in delving deeper into the realm of advanced diffusion model architectures. Investigating the impact of these more intricate architectures on classification accuracy, particularly in highly imbalanced scenarios, holds immense potential for even greater advancements. Additionally, the integration of class-specific information into the diffusion process itself warrants exploration. By tailoring the data augmentation process to focus on the minority class, the proposed problem can potentially achieve even more targeted and impactful improvements. This specific strategy possesses the capability to notably enhance the recall of the minority group, an imperative measure in imbalanced classification assignments.

In summary, this study has effectively illuminated the effectiveness of diffusion mechanisms as a means to boost

classification accuracy in imbalanced datasets. The ability to generate synthetic minority class samples offers a significant advantage in overcoming the inherent bias of imbalanced data. As we move forward, the exploration of advanced diffusion model architectures and class-specific augmentation techniques promises to further refine this approach, paving the way for robust and reliable classification even in the face of imbalanced data.

REFERENCES

[1] Sukarna Barua, Md. Monirul Islam, Xin Yao, and Kazuyuki Murase. Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering*, 26(2):405–425, 2014.

[2] Davtyan-N. Wolfien M Bej, S. Loras: an oversampling approach for imbalanced datasets. *Mach Learn 110*, page 279–301, June 2021.

[3] P.K. Chan, W. Fan, A.L. Prodromidis, and S.J. Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and their Applications*, 14(6):67–74, 1999.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002.

[5] Hyun-Soo Choi, Dahuin Jung, Siwon Kim, and Sungroh Yoon. Imbalanced data classification via cooperative interaction between classifier and generator. *IEEE Transactions on Neural Networks and Learning Systems*, 33(8):3343–3356, 2022.

[6] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. 2021.

[7] Qi Dong, Shaogang Gong, and Xiatian Zhu. Class rectification hard mining for imbalanced deep learning. 2017.

[8] Georgios Douzas, Fernando Bacao, and Felix Last. Improving imbalanced learning through a heuristic oversampling method based on k-means and smote. *Information Sciences*, 465:1–20, 2018.

[9] Panagiotis Filippakis, Stefanos Ougiaroglou, and Georgios Evangelidis. Condensed nearest neighbour rules for multi-label datasets. page 43–50, 2023.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. 2014.

[11] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. *Adv Intell Comput*, 3644:878–887, 09 2005.

[12] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. pages 1322–1328, 2008.

[13] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. 2020.

[15] Feng Hu and Hang Li. A novel boundary oversampling algorithm based on neighborhood rough set model: Nrsboundary-smote. *Mathematical Problems in Engineering, vol.*, 2013.

[16] Junjie Hu, Haiqin Yang, Irwin King, Michael Lyu, and Anthony Man-Cho So. Kernelized online imbalanced learning with fixed budgets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), Feb. 2015.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

[18] Joanna Jedrzejowicz and Piotr Jedrzejowicz. Gep-based classifier with drift detection for mining imbalanced data streams. *Procedia Computer Science*, 176:41–49, 2020. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 24th International Conference KES2020.

[19] Chakraborty-S. Popescu M Khalilia, M. Predicting disease risks from highly imbalanced data using random forest. 2011.

[20] Salman H. Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous Sohel, and Roberto Togneri. Cost sensitive learning of deep feature representations from imbalanced data. 2017.

[21] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. 2022.

[22] Faezeh Movahedi, Rema Padman, and James Antaki. Limitations of roc on imbalanced data: Evaluation of lvad mortality risk scores. 10 2020.

[23] Clifton Phua, Damminda Alahakoon, and Vincent Lee. Minority report in fraud detection: classification of skewed data. *SIGKDD Explor. Newsl.*, 6(1):50–59, jun 2004.

[24] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. 2016.

[25] Dipankar Sarkar, Ankur Narang, and Sumit Rai. Fed-focal loss for imbalanced data classification in federated learning. 2020.

[26] José A. Sáez, Julián Luengo, Jerzy Stefanowski, and Francisco Herrera. Smote–ipf: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, 291:184–203, 2015.

[27] Xinmin Tao, Qing Li, Chao Ren, Wenjie Guo, Chenxi Li, Qing He, Rui Liu, and Junrong Zou. Real-value negative selection over-sampling for imbalanced data set learning. *Expert Systems with Applications*, 129:118–134, 2019.

[28] Carla Vairetti, José Luis Assadi, and Sebastián Maldonado. Efficient hybrid oversampling and intelligent undersampling for imbalanced big data classification. 2023.

[29] Hengrui Zhang, Jiani Zhang, Balasubramaniam Srinivasan, Zhengyuan Shen, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. 2024.

[30] Y. Zhang. Deep generative model for multi-class imbalanced learning. 2018.

[31] Xing-Ming Zhao, Xin Li, Luonan Chen, and Kazuyuki Aihara. Protein classification with imbalanced data. *Proteins: Structure, Function, and Bioinformatics*, 70(4):1125–1132, 2008.

[32] Chao Wang; Zhibin Yu; Haiyong Zheng; Nan Wang; Bing Zheng. Cgan-plankton: Towards large-scale imbalanced class generation and fine-grained classification. 2017.

[33] Qian Zhou and Bo Sun. Adaptive k-means clustering based under-sampling methods to solve the class imbalance problem. *Data and Information Management*, page 100064, 2023.

[5] [31] [3] [23] [13] [24] [2] [20] [8] [32] [18] [19] [30] [10] [6] [14] [33] [9] [4] [11] [15] [12] [1] [27] [28] [26] [7] [25] [22] [16] [17] [21] [29]

TABLE V
REPRESENTS THE ERROR RATE COLUMN-WISE FOR ALL THE DATASETS DURING THE GENERATION PROCESS FOR ALL DIFFERENT GENERATIVE ALGORITHMS [33], THE TABLE SHOWS THAT THE MODEL HAS AN AVERAGE 86% LESS ERROR RATE COMPARED TO OTHER TECHNIQUES

$Model(M)$	$Data_{adult}$	$Data_{Default}$	$Data_{Shoppers}$	$Data_{Magic}$	$Data_{Beijing}$	$DataNews$	$Average_{results}$
SMT	1.60	1.48	2.68	0.91	1.85	5.31	2.30
CTGAN	$16.84_{\pm 0.03}$	$16.83_{\pm 0.04}$	$21.15_{\pm 0.10}$	$9.81_{\pm 0.08}$	$21.39_{\pm 0.05}$	$16.09_{\pm 0.02}$	17.02
TVAE	$14.22_{\pm 0.08}$	$10.17_{\pm 0.05}$	$24.51_{\pm 0.06}$	$8.25_{\pm 0.06}$	$19.16_{\pm 0.06}$	$16.62_{\pm 0.03}$	15.49
GOOGLE	16.97	17.02	23.33	1.90	16.93	25.32	16.74
GReaT	$12.12_{\pm 0.04}$	$19.94_{\pm 0.06}$	$14.51_{\pm 0.12}$	$16.16_{\pm 0.09}$	$8.25_{\pm 0.23}$	Max Limit	14.20
STaSy	$11.29_{\pm 0.06}$	$5.77_{\pm 0.06}$	$9.37_{\pm 0.09}$	$6.29_{\pm 0.13}$	$6.71_{\pm 0.03}$	$6.89_{\pm 0.03}$	7.72
CoDi	$21.83_{\pm 0.06}$	$15.77_{\pm 0.07}$	$31.84_{\pm 0.05}$	$11.560_{\pm 0.26}$	$16.94_{\pm 0.02}$	$32.27_{\pm 0.04}$	21.63
TabDDPM	$1.75_{\pm 0.03}$	$1.57_{\pm 0.08}$	$2.72_{\pm 0.13}$	$1.01_{\pm 0.09}$	$1.30_{\pm 0.03}$	$78.75_{\pm 0.01}$	14.52
TabSyn	$0.58_{\pm 0.06}$	$0.85_{\pm 0.04}$	$1.43_{\pm 0.24}$	$0.88_{\pm 0.09}$	$1.12_{\pm 0.05}$	$1.64_{\pm 0.04}$	1.08
Improvement	$66.9\% \downarrow$	$45.9\% \downarrow$	$47.4\% \downarrow$	$12.9\% \downarrow$	$13.8\% \downarrow$	$76.2\% \downarrow$	$86.0\% \downarrow$

Vikram Velankar The author, an undergraduate researcher passionate about tackling real-world issues, aims to challenge misconceptions in the field of machine learning by investigating the use of diffusion processes for imbalanced data classification. Their experience working with Dr. Abderrahim Benslimane (Avignon University) and Dr. Ankit Kumar (Indian Institute of Management Calcutta) has fostered a strong desire to contribute to research groups and make a positive global impact.

Sachin S. Patil The manuscript's second author brings a wealth of experience to the table. A professor with over 20 years of teaching expertise and 25+ years in machine learning, their research focus lies in data balancing and big data applications for classification tasks. He has worked as head of Computer Science and Engineering department at Rajarambapu Institute of Technology, Rajaramnagar, MH – India. His research presentations have been awarded twice with "Best Paper Award" at Research Symposium conducted under ACM chapter at WCE, Sangli.