Mohammed Abdalraheem[1]*,
Yagoub Abbker Adam[1],
Mohammad Khamruddin[1],
Mostafa Mehanawi[2],
Ahamed Ali Meeran[1],
Shaik Rizwan[2],
Ali Douik[3]

A Learning Method for Class Imbalance Problem: A Case Study of Churn Prediction

Journal of Electrical Systems

Abstract : The issue of class imbalance, particularly in relation to Machine Learning (ML) models, are common challenge in practical applications. ML is a field in which class imbalance is problematic, as it slows down the best learning process even by the best ML models. This issue significantly impacts performance, as these methods often prioritize learning the majority class while neglecting the distribution of the minority classes. In this paper, a novel oversampling method based on the stratification of Pascal's triangle is introduced to tackle the problem of class imbalance. The method is designed to enhance the learning process of ML models by facilitating a more effective representation of minority classes. To evaluate effectiveness of the developed method, we conduct experiments on six benchmark datasets for the application of churn prediction. The results indicate that the developed method consistently outperforms SMOTE, ADASYN, G-SMOTE, and Gaussian oversampling techniques. Also, this approach appears to be not only an effective but also an efficient option for improving ML models' learning processes under the conditions of class imbalance.

Keywords: Class imbalance, machine learning, oversampling, churn prediction, optimization, Pascal's tringle.

1. Introduction

Class imbalance can also lead to biased models that perform poorly in real-world scenarios where both classes are equally important. Various techniques, such as oversampling, undersampling, and ensemble methods, have been developed to address this issue and improve the performance of classification algorithms on imbalanced datasets [1]. This issue has become increasingly serious for researchers due to its potential impact on several applications, including malware detection [2], medical diagnosis [3, 4], financial crisis prediction [5], and churn prediction [6].

Data balancing techniques are essential when working with highly imbalanced datasets, as such distributions can cause models to incorrectly classify the majority of instances as negative, thereby increasing the number of false negatives [7, 8]. Thus, it is crucial to adopt an effective balancing strategy with a strong interpretability component during the preprocessing phase to accurately identify churn customers. As is often the case in predictive analytics, handling false positives—particularly in churn prediction—poses considerable risks and costs; therefore, the need to address class imbalance is critical [9].

Although ML models exhibit generalization, scalability, and interpretability, they face considerable challenges when dealing with imbalanced data. Many solutions were presented to tackle this problem which can be broadly classified into three groups: data level, algorithm level and hybrid methods [10, 11]. Data level strategies addressed the issue by enhancing the under-represented classes in the dataset by generating synthetic minority samples. The simplest of these techniques is random sampling, which aims to improve the quality of the training set before classification algorithms are trained. Random sampling can be divided into random undersampling and oversampling [12].

The Synthetic Minority Oversampling Technique (SMOTE) [13], and the Adaptive Synthetic Sampling Approach (ADASYN) [14], are among the most widely used methods for addressing class imbalance. In these

[1]Department of Computer Science, College of Engineering and Computer Sciences, Jazan University, Jazan, Saudi Arabia.
[2]Department of Electrical and Electronics Engineering, College of Engineering and Computer Sciences, Jazan University, Jazan, Saudi Arabia.
[3]National Engineering School of Sousse, NOCCS-ENISO Lab, University of Sousse, Sousse, Tunisia
* Correspondent author: Mohammedh@jazanu.edu.sa

techniques, new minority-class instances are generated by identifying nearest neighbors and interpolating between a given sample and one randomly selected neighbor [14, 15]. These approaches help balance positive and negative examples in the dataset; however, the generation of synthetic samples also introduces randomness, which can detract from a clear understanding of the minority class. Consequently, learning algorithms may face challenges in accurately modeling and classifying minority-class instances [13].

Many techniques have been suggested to remedy this problem [11, 15, 17, 18]. In [11], the HEOMGA approach is developed, which uses Heterogeneous Euclidean-Overlap Metric (HEOM) and genetic algorithm (GA) combinations, especially in the area of over sampling minority class. HEOM is used as a fitness function that facilitates the GA to create new data points. The HEOMGA technique was found to be more efficient than a number of popular oversampling techniques. In [15], the authors propose a novel SMOTE based method called Range Controlled SMOTE (RCSMOTE) for handling three major issues that arise in oversampling. This method relies on a sample classification scheme to determine suitable samples from the minority class and an improved filling process which synthesizes samples from the input data within the 'safe zone'. This enables pattern space over-sampling to be done both safely and efficiently. The findings verified that RCSMOTE gained better results than the classical SMOTE method. In [17] a new and basic approach convex hull based SMOTE' or CHSMOTE has been suggested to counter problems faced by SMOTE technologies. The results suggest that the algorithm implemented in this case is comparatively more effective than SMOTE. Furthermore, the authors in [18] introduce another novel technique called Multi-vector Stochastic Exploration Oversampling (MSEO) that is an extension of the conventional SMOTE technique. The technique has the advantage of producing synthetic samples with random direction and scaling vectors, instead of the traditional approach whereby vectors are determined by neighboring samples, hence more generalized oversampling technique is offered. The results indicate MSEO improves the classification efficiency even more so than SMOTE.**[**5]

This work presents a data-oriented oversampling approach that addresses class imbalances by utilizing the Pascal Triangle logic to generate new data points from available minority classes instead of relying on random sampling. The primary goal is to assess the effectiveness of the proposed method in achieving optimal performance and enhancing the learning process for imbalanced datasets. Initially, the technique chooses a pair of data points in the minority class that have similar feature values and combines them to create a new data point. Next, we use the newly created data points to generate additional points, repeating this process until we achieve a balance between the minority and majority class data points.

The rest of the paper is structured as follows: The section 2 reviews SMOTE and ADASYN oversampling methods. Proposed method is presented in section 3. The datasets used to validate the proposed method are described in section 4, while section 5 provides the evaluation measures used to assess the proposed method and its efficiency. The section 6 presents the experimental results obtained by the used methods. Finally, section 7 concludes the paper along with possible future research directions.

2. Methods and materials

This section provides and overview for SMOTE, ADASYN, G-SMOTE and Gaussian method and proposed method and used datasets.

2.1. SMOTE

Chawla et al. initially developed SMOTE [13], a popular method to address class imbalance in datasets. It works by generating synthetic samples for the minority class through interpolation between existing minority instances rather than simply duplicating them. It improves the performance of classification algorithms by balancing the class distribution, which reduces bias toward the majority class. However, SMOTE may introduce noise if synthetic instances are generated in overlapping regions. A new synthetic minority class example is generated on the line segment between X_i and $\widetilde{X}$. both $X_i, \widetilde{X} \in N_{min}$ is computed as follows:

$$X_{new} = X_i + \left(\widetilde{X} - X_i\right) * rand(0,1), \tag{1}$$

where X_i is the oversampled minority sample, $\check{X}$, is another minority sample, which is usually selected from N_{min} samples near to X_i, $rand(0,1)$ is a random number in the range of [0,1].

2.2. ADASYN

He, Bai, Garcia, and Li [14] introduced ADASYN to address the issue of class imbalance. It was developed as an extension of SMOTE and it is designed to reduce the generation of noisy data along the decision boundaries. ADASYN effectively minimizes the learning bias caused by the original imbalanced data by shifting the decision boundary to focus on the harder-to-learn examples in the dataset. The new synthetic examples are generated through the following process:

$$s_i = x_i + (x_{zi} - x_i)\lambda, \tag{2}$$

where s_i is the new synthetic example, x_i and x_{zi} are two minority examples within and λ is a random number between [0,1].

2.3. G-SMOTE

G-SMOTE is an enhanced version of SMOTE that tackles the problem of addresses class imbalances by generating synthetic samples for the minority class. Unlike traditional SMOTE, G-SMOTE considers the geometric properties of data, such as distances and angles, to create more representative samples. This approach helps improve model performance and reduce overfitting in imbalanced datasets, particularly those with complex distributions [19]

2.4. Gaussian method

This method is used to generate synthetic samples for the minority class by modeling the data distribution with a Gaussian (normal) distribution. It assumes that the minority class follows a Gaussian distribution and then generate new samples by sampling from this distribution. This method helps balance the class distribution by producing realistic minority class data points [20]

2.5. Proposed method

Pascal's triangle is an algebraic arrangement of numbers that was known long before the Italian mathematician Niccolò Fontana Tartaglia (1500–1577) and the French scientist Blaise Pascal (1623–1662). It had also been explored by various mathematicians in regions such as China, India, and Iran prior to their work [21]. Binomial coefficient is a simple type of algebraic expression and its properties was widely explored by many over the history of mathematics [22, 23, 24]. Binomial coefficient has just two terms operated on only by addition, subtraction, multiplication. Pascal's triangle in mathematics is a triangular array of binomial coefficients (Fowler, 1996), where each number is the sum of the two numbers directly (right and left) above it as shown in Figure 1.

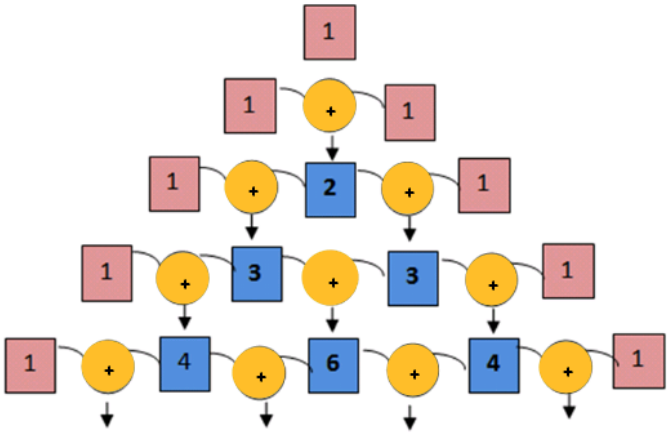


Figure 1. Pascal's triangle

Pascal's triangle can be used to find out the number of possible combinations of n items using the following:

$$\binom{n}{k} = (n,k) = \frac{n!}{(n-k)!\,k!}$$

where n presents number of items and k is the number of k items can be ordered.

It is amazing to note how this formula works and its relationship with Pascal's triangle. For example, one can choose 3 balls from 16 or can choose 13 balls out of 16 have the same number of combinations as shown in figure 6. This can be achieved using the follows:

$$\binom{16}{3} = \frac{16! * 15! * 14! * 13!}{(16-3)!\,3!} = 560$$

The idea behind the proposed strategy is to deal with the problem of class imbalance and restrict the possibility of negative effects which can be caused by generating random examples on the predictive performance. Multiple solutions may exist to resolve class imbalance problem by adding random synthetic instances. However, the proposed method aims to generate consistent data points in order to improve learning process.

The proposed method aims to generate consistent data points in order to improve learning process. In the first step, the proposed strategy selects two points (samples) from the minority class in the dataset within the same attribute values to produce a real point. The newly generated data point presents the sum of the two data points above it. This can be achieved using the logic of Pascal triangle. In the next step, the produced real data points will be used to generate new ones. Subsequently, the produced points that have been generated from two similar points will be ignored in order to remove duplication and to make sure that the point has not been selected twice during this step. This procedure will be repeated and once the total number of the new generated points equals the number of majority classes, the procedure stops. Finally, all the generated data points are normalized within the range of the attributes values by utilizing equation (3). Algorithm 2 provides the pseudocode of the suggested method:

$$X_{new} = \left(\mathrm{X_i} - \mathrm{X_j}\right)\frac{\mathrm{S} - \mathrm{X_j}}{\mathrm{max_x} - \mathrm{X_j}} + \mathrm{X_j} \tag{3}$$

where, $\mathrm{X_i}$ and $\mathrm{X_j}$ is two minority data points within the same feature values, $\mathrm{X_i}$ presents the maximum value while $\mathrm{X_j}$ is the minimum value, S is the sum of $\mathrm{X_i}$ and $\mathrm{X_j}$, $\mathrm{max_x}$ is the maximum number from the original minority data points and the generated ones.

Algorithm 2 Pseudocode of the proposed method

1**: Input:** Collect dataset related to customer churn
2: **Output:** Balanced dataset
3: **Set**:
4; N: Number of minority class samples
5: M: Number of majority class samples
6: c: Counter
7: **Method**:
8: **Step 1:** Compute number of combinations that can be produced from N
9: K=2; this step to choose two samples from the minority N
10: $\binom{n}{k} = \frac{n!}{(n-k)!k!}$
11: **Step 2:** Produce new real examples
12: A=$\{a_0,\ a_1 \cdots a_n\}$
13: B= $\{b_0,\ b_1 \cdots b_n\}$
14: While c <= N
15: If $a_i \in$ to A and $b_i \in$ to B then
16. D= A+B this step will select two real examples and produce a new one

17: If the new examples (D) produced from two similar examples ($a_i = b_i$) then ignore
18: End while
19: **Step 3:** Use the new real examples produced in Step 2 to produce new examples and repeat this procedure until reaching to N=M
20: End of Pseudocode

2.6. *Datasets*

A set of publicly available datasets are used, and they obtained from Kaggle repository [25]. The datasets with their characteristics are provided in Table 2, [26,27].:

Table 1. Summary of the dataset's characteristics

Dataset	No. of instances	No. of features	No. of class	No. of churners	No. of non-churners
DS-1	3,333	21	2	483	2,850
DS-2	71,047	58	2	14,210	56,837
DS-3	100,000	100	2	49,562	50,438
DS-4	3,333	11	2	483	2,850
DS-5	3,150	16	2	495	2,655
DS-6	50,375	10	2	20,331	30,044

3. Experimental results

3.1. Experimental setup

To assess the performance of the developed methods, Support Vector Machine (SVM) learner is employed. The SVM seeks to identify the hyperplane that separates instances of two classes by maximizing the distance between them. This approach is particularly effective due to its ability to operate in high-dimensional feature spaces, allowing it to capture complex nonlinear relationships with relatively high accuracy between input and output [28, 29, 30, 31, 32, 33]. The SVM with a radial basis function (SVM_{rbf}) kernel is utilized for this analysis.

For the SMOTE, ADASYN, G-SMOTE, and Gaussian methods, the input parameter settings are based on their original implementations. All experiments are conducted using Python's scikit-learn library, with SVM_{rbf} learners constructed using default parameters on a Windows 7 system equipped with a Duo CPU running at 3.13 GHz and 44.25 GB of RAM.

3.2. Evaluation metrics

Several evaluation metrics are used to evaluate the methods, and these metrics are widely adopted in churn prediction [11, 26, 29].

$$\text{Recall} = \frac{TP}{TP+FN}\text{,,} \tag{2}$$

$$Gmean = \sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}} \tag{3}$$

$$AUC = \frac{\left(1 + \frac{TP}{TP + FN} - \frac{FP}{FP + TN}\right)}{2} \quad (4)$$

where, True Positive (TP) and True Negative (TN) refer to the counts of positive and negative examples that are correctly classified, while False Negative (FN) and False Positive (FP) indicate the counts of misclassified positive and negative examples, respectively.

3.3. Results and discussion

A 10-fold cross-validation approach is employed to ensure that no specific portions of the dataset are exclusively used for training or testing. The value of k is set to 10, resulting in the dataset being divided into 10 parts. The process begins by allocating 90% of the data for training and 10% for testing. This procedure is repeated 10 times, allowing each part of the data to serve as the test set for each algorithm utilized in this study. Ultimately, the average results across the 10 partitions are considered. The results, including those obtained without any balancing method (i.e., 0% balancing), as well as those from the proposed and other methods, are summarized in Table 2.

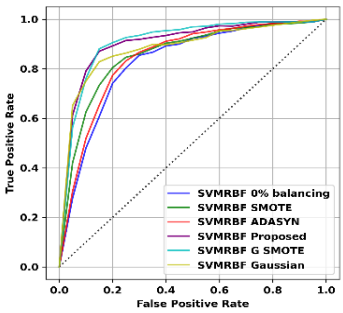
Table 2. Summary of the performance results using the Datasets

Dataset	Method	Recall	G mean	AUC
DS-1	0% balancing	0.719	0.706	0.724
	SMOTE	0.814	0.816	0.817
	ADASYN	0.676	0.739	0.739
	Proposed	**0.845**	**0.919**	**0.919**
	G SMOTE	**0.845**	**0.919**	0.918
	Gaussian	0.837	0.913	0.914
DS-2	0% balancing	0.581	0.607	0.636
	SMOTE	0.605	0.640	0.681
	ADASYN	0.539	0.586	0.642
	Proposed	**0.674**	**0.705**	**0.740**
	G SMOTE	0.673	0.697	0.724
	Gaussian	0.668	0.699	0.734
DS-3	0% balancing	0.443	0.537	0.547
	SMOTE	0.395	0.524	0.545
	ADASYN	0.402	0.535	0.544
	Proposed	**0.502**	**0.557**	**0.560**
	G SMOTE	0.500	0.529	0.530
	Gaussian	0.498	0.551	0.553
DS-4	0% balancing	0.763	0.765	0.767
	SMOTE	0.836	0.838	0.839
	ADASYN	0.730	0.755	0.781
	Proposed	**0.864**	**0.889**	**0.916**
	G SMOTE	**0.864**	0.885	0.912
	Gaussian	0.852	0.879	0.907
DS-5	0% balancing	0.806	0.808	0.811
	SMOTE	0.859	0.860	0.861

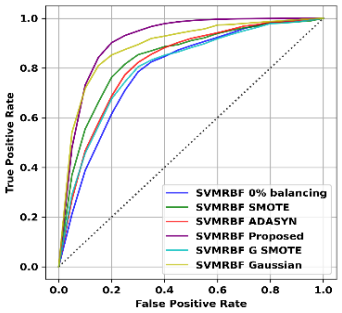
	ADASYN	0.785	0.803	0.823
	Proposed	**0.884**	**0.898**	**0.913**
	G SMOTE	**0.884**	**0.889**	0.905
	Gaussian	0.868	0.883	0.900
DS-6	0% balancing	0.893	0.895	0.897
	SMOTE	0.903	0.904	0.905
	ADASYN	0.893	0.900	0.907
	Proposed	**0.922**	**0.914**	**0.906**
	G SMOTE	0.903	0.897	0.892
	Gaussian	0.898	0.891	0.885

Table 2 demonstrates that the proposed method outperforms 0% balancing, SMOTE, ADASYN, G-SMOTE, and the Gaussian method in terms of Recall across all datasets used in the study. This indicates that the proposed approach is more effective at identifying true positive instances from the minority class, leading to better classification results. The superior Recall values highlight the method's ability to capture a higher proportion of the minority class, which is critical in applications such as customer churn prediction. By improving the identification of at-risk customers, the proposed method improves the churn rate prediction compared to traditional oversampling techniques. As a result, this method offers a more reliable solution for dealing with class imbalance, ensuring that minority class instances are better represented and contributing to a more balanced and accurate learning process.

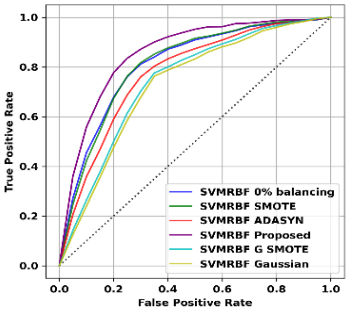
The findings indicate that the suggested method surpassed other oversampling techniques regarding both G-mean and AUC across the datasets. The proposed technique attained the highest G-mean and AUC values across the six datasets. This enhancement is due to the comprehensive information supplied by the proposed approach, which improved both the predictive outcomes and the learning process as well. The ROC graph evaluates the SVM_{rbf} and adjusts its confidence level scores to derive unique values for the True Positive Rate (TP rate) and False Positive Rate (FP rate), as depicted in Figure 2.



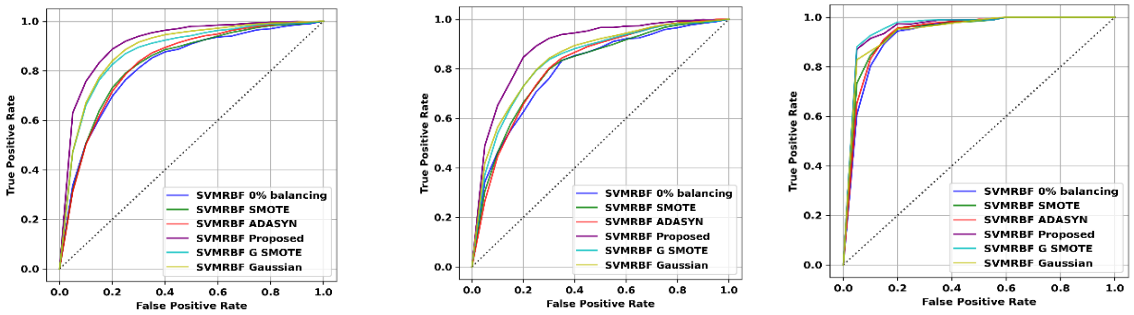
(a) DS-1



(b) DS-2



(c) DS-3



(d) DS-4 (e) DS-5 (f) DS-6

Figure 2. ROC carver results obtained by SVM_{rbf} on the datasets

4. Conclusion

This work tries to tackle class imbalance problem in order to achieve a better prediction accuracy in the application of churn prediction. In this study, a preprocessing approach as an alternative solution to mitigate the class imbalance issue is presented, helping to enhance the learner's generalization capacity and performance. While oversampling methods generate instances randomly, our proposed method produces authentic data-points from the existing minority class samples, which in turns provides richer information about the new generated data-paints. Through experiments, the proposed method behaved excellent and exhibited superior performance over existing oversampling techniques in overcoming class imbalance, demonstrating both effectiveness and efficiency. Future work could benefit from assessing its performance across other applications and datasets where class imbalance remains a significant challenge

Conflicts Of Interest

The author's affiliations, financial relationships, or personal interests do not present any conflicts in the research.

Datasets availability:

References

[1] Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence, 23(04), 687-719.

[2] Sikri, A., Jameel, R., Idrees, S. M., & Kaur, H. (2024). Enhancing customer retention in telecom industry with machine learning driven churn prediction. Scientific Reports, 14(1), 13097.

[3] Zhou, Q., & Sun, B. (2024). Adaptive K-means clustering based under-sampling methods to solve the class imbalance problem. Data and Information Management, 8(3), 100064.

[4] AlShourbaji, I., Kachare, P., Zogaan, W., Muhammad, L. J., & Abualigah, L. (2022). Learning features using an optimized artificial neural network for breast cancer diagnosis. SN Computer Science, 3(3), 229.

[5] Abhishake Reddy Onteddu, Rahul Reddy Bandhela; RamMohan Reddy Kundavaram. Enhancing E-Commerce Product Recommendations through Data Engineering and Machine Learning. ES 2024, 20 (1), 171-183. https://doi.org/10.69889/vqgzw857.

[6] Guo, H. (2023). The design of early warning software systems for financial crises in high-tech businesses using fusion models in the context of sustainable economic growth. Peerj Computer Science, 9, e1326.

[7] AlShourbaji, I., Helian, N., Sun, Y., & Alhameed, M. (2021). Customer churn prediction in telecom sector: A survey and way a head. International Journal of Scientific & Technology Research (IJSTR)..

[8] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[9] Liu, X. Y., Wu, J., & Zhou, Z. H. (2009). Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39(2), 539-550.

[10] Wang, S., Hussien, A. G., Kumar, S., AlShourbaji, I., & Hashim, F. A. (2023). A modified smell agent optimization for global optimization and industrial engineering design problems. Journal of Computational Design and Engineering, 10(6), 2147-2176.
[11] Zhu, B., Qian, C., vanden Broucke, S., Xiao, J., & Li, Y. (2023). A bagging-based selective ensemble model for churn prediction on imbalanced data. Expert Systems with Applications, 227, 120223.
[12] AlShourbaji, I., Helian, N., Sun, Y., & Alhameed, M. (2021). A novel HEOMGA approach for class imbalance problem in the application of customer churn prediction. SN Computer Science, 2, 1-12.
[13] Ghosh, K., Bellinger, C., Corizzo, R., Branco, P., Krawczyk, B., & Japkowicz, N. (2024). The class imbalance problem in deep learning. Machine Learning, 113(7), 4845-4901.
[14] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.
[15] He, H., Bai, Y., Garcia, E. A., & Li, S. (2008, June). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on (pp. 1322-1328). IEEE.
[16] Soltanzadeh, P., & Hashemzadeh, M. (2021). RCSMOTE: Range-Controlled synthetic minority over-sampling technique for handling the class imbalance problem. Information Sciences, 542, 92-111.
[17] Islam, A., Belhaouari, S. B., Rehman, A. U., & Bensmail, H. (2022). KNNOR: An oversampling technique for imbalanced datasets. Applied soft computing, 115, 108288.
[18] Yuan, X., Chen, S., Zhou, H., Sun, C., & Yuwen, L. (2023). CHSMOTE: Convex hull-based synthetic minority oversampling technique for alleviating the class imbalance problem. Information Sciences, 623, 324-341.
[19] Li, H., Wang, S., Jiang, J., Deng, C., Ou, J., Zhou, Z., & Yu, D. (2024). Augmenting the diversity of imbalanced datasets via multi-vector stochastic exploration oversampling. Neurocomputing, 583, 127600.
[20] Douzas, G., Rauch, R., & Bacao, F. (2021). G-SOMO: An oversampling approach based on self-organized maps and geometric SMOTE. Expert Systems with Applications, 183, 115230.
[21] Xie, Y., Qiu, M., Zhang, H., Peng, L., & Chen, Z. (2020). Gaussian distribution-based oversampling for imbalanced data classification. IEEE Transactions on Knowledge and Data Engineering, 34(2), 667-679.
[22] Coolidge, J.L., 1949. The story of the binomial theorem. The American Mathematical Monthly, 56(3), pp.147-157.
[23] Knuth, D.E., 2014. Art of computer programming, volume 2: Seminumerical algorithms. Addison-Wesley Professional.
[24] Zheng, C. and Zheng, J., 2019. Triangular Numbers and Their Inherent Properties. In Variant Construction from Theoretical Foundation to Applications (pp. 51-65). Springer, Singapore.
[25] Rajasekaran, A., Shallit, J., & Smith, T. (2020). Additive number theory via automata theory. Theory of Computing Systems, 64, 542-567.
[26] Kaggle, https://www.kaggle.com/, Accessed on 12th august 2024.
[27] Katrawi, A. H., Abdullah, R., Anbar, M., AlShourbaji, I., & Abasi, A. K. (2021). Straggler handling approaches in mapreduce framework: a comparative study. International Journal of Electrical & Computer Engineering (2088-8708), 11(1).
[28] AlShourbaji, I., Helian, N., Sun, Y., Hussien, A. G., Abualigah, L., & Elnaim, B. (2023). An efficient churn prediction model using gradient boosting machine and metaheuristic optimization. Scientific Reports, 13(1), 14441.
[29] Verbiest, N., Ramentol, E., Cornelis, C., & Herrera, F. (2014). Preprocessing noisy imbalanced datasets using SMOTE enhanced with fuzzy rough prototype selection. Applied Soft Computing, 22, 511-517.
[30] Ganaie, M. A., Tanveer, M., & Alzheimer's Disease Neuroimaging Initiative. (2022). KNN weighted reduced universum twin SVM for class imbalance learning. Knowledge-based systems, 245, 108578.
[31] Al-Janabi, S., & Al-Shourbaji, I. (2017). A smart and effective method for digital video compression. 2017 International Conference on Information and Digital Technologies (IDT). https://doi.org/10.1109/dt.2017.8012161
[32] Ganaie, M. A., Tanveer, M., & Lin, C. T. (2022). Large-scale fuzzy least squares twin SVMs for class imbalance learning. IEEE Transactions on Fuzzy Systems, 30(11), 4815-4827.

[33] Alberto Fernández et.al. (2018). Learning from imbalanced data sets. Springer Nature Switzerland. https://doi.org/10.1007/978-3-319-98074-4

[34] Alberto Fernández et.al. (2023). Machine learning for imbalanced data. Packt Publishing Ltd. Birmingham. UK. www.packtpub.com. ISBN 978-1-80107-083-6.