Computer Science (CO-SCIENCE)
Volume 6 Issue 2 July 2026
Accreditation **Sinta 3** No. SK : 295/C/C3/KPT/2026

Performance of Distance Metrics in SMOTE for Binary Imbalanced Classification

Fandi Yulian Pamuji[1*], Luthfi Indana[2], Mohammad Dwi Irfan Affandi[3]

[1,2,3]Universitas Merdeka Malang
Jalan Terusan Dieng. 62-64 Klojen, Pisang Candi, Kec. Sukun, Kota Malang, Jawa Timur 65146, Indonesia

e-mail: [1]fandi.pamuji@unmer.ac.id, [2]luthfi.indana@unmer.ac.id, [3]22083000105@student.unmer.ac.id

(*) Corresponding Author

Article Info: Received: 02-04-2026 | Revised : 09-07-2026 | Accepted : 21-07-2026

Abstracts - Skewed class distribution continues to be one of the central obstacles in binary classification, since a learning model tends to lean toward the dominant class and consequently overlooks observations belonging to the under-represented class. The purpose of this research is to examine how the choice of distance measure inside SMOTE, specifically Euclidean, Manhattan, Chebyshev, and Hamming, affects predictive quality on imbalanced binary data. Ten publicly available binary datasets drawn from the KEEL repository, whose imbalance ratios span from 1.86 up to 15.80, were used in the experiment. Every dataset was preprocessed and partitioned into 80% for training and 20% for testing; oversampling with SMOTE was carried out on the training portion only, after which four learners, namely Naive Bayes, Decision Tree, Logistic Regression, and k-Nearest Neighbor, were assessed. Model quality was judged through the Matthews Correlation Coefficient (MCC) together with the G-Mean, as these two indicators describe imbalanced performance more faithfully than plain accuracy. The comparison revealed that pairing Euclidean-based SMOTE with Logistic Regression yielded the strongest average scores (MCC = 0.72; G-Mean = 0.79); Manhattan-based SMOTE reached its top MCC again with Logistic Regression (MCC = 0.68) and its top G-Mean with the Decision Tree (G-Mean = 0.79); Chebyshev-based SMOTE delivered the best overall combination together with the Decision Tree (MCC = 0.74; G-Mean = 0.84); and Hamming-based SMOTE performed best alongside Logistic Regression (MCC = 0.73; G-Mean = 0.81). Taken together, these outcomes suggest that the distance function chosen within SMOTE shapes the quality of the generated synthetic points and, in turn, the behavior of the trained classifier.

Keywords : Binary Imbalanced Classification, Distance Metrics, SMOTE, MCC, G-Mean.

INTRODUCTION

As digital information keeps expanding rapidly, data mining and machine learning have grown increasingly central to uncovering useful patterns hidden within large collections of data. Within classification, how dependable a model turns out to be rests not merely on the chosen learning algorithm but equally on the way the training data are distributed and represented. When the classes are evenly represented, a model can shape its decision boundaries in a balanced manner; conversely, an uneven distribution pushes the learning process to be governed mainly by the dominant class (Misdram et al., 2023; Chen et al., 2024; Yulian Pamuji et al., 2024).

Situations with skewed classes appear regularly in real applications, including clinical diagnosis, detection of fraudulent transactions, identification of machine faults, and prediction of software defects. In such settings the rare class usually carries the most critical meaning even though it is backed by far fewer samples than the common class. Because of this, ordinary accuracy may give a misleading impression: a model can score highly simply by favoring the majority label while remaining unable to detect the rare cases. For that reason, indicators like MCC and G-Mean fit imbalanced problems better, since they reflect how well a model handles both the majority and the minority side (Cruz Huayanay et al., 2025; Imani et al., 2026; Wong & Chung, 2025).

A number of resampling techniques have been introduced to counter class imbalance, and SMOTE ranks among the most popular because, rather than copying existing records, it synthesizes new minority examples. Even so, earlier work has pointed out several notable shortcomings of the method. The synthetic points may fall within noisy or overlapping areas, the neighborhood definition can be vulnerable to outliers, and the standard distance formulation is not necessarily suitable for every kind of feature distribution (Hairani et al., 2024; Matharaarachchi et al., 2024; Widiyaningtyas et al., 2025; Zhang et al., 2024). These limitations indicate that the distance metric guiding the construction of SMOTE neighborhoods is far from a minor technicality; rather, it is a methodological

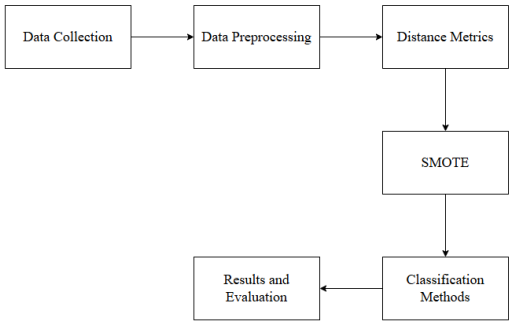
element capable of shaping how good the generated samples turn out to be.

The gap this study targets concerns the scarce comparative examination of distance measures inside the SMOTE procedure for binary imbalanced data. A large share of prior research has concentrated on contrasting classifiers, weighing SMOTE against alternative resampling schemes, or introducing fresh SMOTE variants. Comparatively few investigations, however, place Euclidean, Manhattan, Chebyshev, and Hamming distances side by side under one consistent experimental design that spans several binary datasets and relies on imbalance-aware indicators. Addressing this gap matters because every distance measure embodies a distinct geometric assumption about how neighborhoods are formed.

What distinguishes this work is its structured assessment of four distance measures within SMOTE, carried out over ten publicly accessible binary imbalanced datasets with four learning algorithms and two imbalance-focused evaluation indicators. In concrete terms, the study contributes by examining how Euclidean, Manhattan, Chebyshev, and Hamming distances influence SMOTE-based binary classification, by pinpointing the classifier-distance pairing that yields the highest MCC and G-Mean, and by offering a comparative overview clarifying which measure stays more consistent across different classifiers. Building on the gap described above, the study sets out to appraise and contrast the effectiveness of distance-metric-driven SMOTE for binary imbalanced classification. Its precise aim is to establish which pairing of distance measure, SMOTE, and learning algorithm delivers the most balanced predictive outcome as gauged by MCC and G-Mean.

RESEARCH METHOD

The present research was structured as an experimental comparison aimed at understanding how distance measures embedded in SMOTE shape binary imbalanced classification results. The framework put forward proceeds through seven stages: gathering ten binary imbalanced datasets from the KEEL repository; preparing and normalizing the features; dividing each dataset into 80% training and 20% testing portions through a stratified scheme; oversampling the training portion with SMOTE under the Euclidean, Manhattan, Chebyshev, and Hamming measures; building classifiers with Naive Bayes, Decision Tree, Logistic Regression, and k-Nearest Neighbor; assessing the results with MCC and G-Mean; and finally contrasting the mean performance obtained by each method.



Source: Research Result (2025)
Figure 1. Proposed Research Framework

The research framework is depicted in Figure 1. A key point of the design is that balancing is performed exclusively on the training subset once the split has been made, whereas the testing subset stays untouched. Handling the data in this way limits the risk of data leakage and guarantees that the reported MCC and G-Mean genuinely represent how the model behaves on data it has never seen.

1. Data collection

For the data-collection stage, ten binary imbalanced datasets were retrieved from the publicly available KEEL repository.

Table 1. Binary Imbalanced Dataset

Dataset	IR (Imbalance Ratio)	Instance	Attribute
ecoli-0-3-4-7_vs_5-6	9.28	257	7
ecoli-0-6-7_vs_5	10.0	220	6
ecoli-0-4-6_vs_5	9.15	203	6
ecoli-0-6-7_vs_3-5	5.14	215	5
ecoli-0-1_vs_2-3-5	9.17	244	7
ecoli4	15.8	336	7
pima	1.87	768	8
new-thyroid1	5.14	215	5

vehicle0	3.25	846	18
wisconsin	1.86	220	7

Source: KEEL-dataset (2025)

2. Data Preprocessing

Preprocessing was carried out so that every dataset would fit the experimental pipeline. To keep differences in feature scale from dominating the distance computation, the numerical attributes were normalized. Each dataset was subsequently divided into 80% training and 20% testing subsets by means of a stratified approach that keeps the class proportions intact, and SMOTE was invoked on the training subset alone. In the Hamming-based variant, the numerical attributes were first turned into interval-based categorical indicators using cut points obtained solely from the training data, allowing Hamming distance to capture mismatch patterns instead of the raw magnitude of the numbers.

3. *Distance Metrics*

Euclidean distance

Grounded in the Pythagorean theorem, Euclidean distance quantifies how far apart two points lie within a Euclidean space, whether on a 2D plane or in 3D. For a pair of points X=(X1,X2,…,Xn) and Y=(Y1,Y2,…,Yn), the Euclidean distance is written as follows.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (1)$$

Notation used:
d(x,y) = the distance separating the two points x and y
xi = the i-th attribute value of point x
yi = the i-th attribute value of point y
n = the total count of attributes (features)
∑ = summation taken from i = 1 through n

Manhattan distance

Manhattan distance gauges the separation between two points in a multidimensional space by summing the absolute gaps between their matching Cartesian coordinates. For a pair of points X=(X1,X2,…,Xn) and Y=(Y1,Y2,…,Yn), the Manhattan distance is written as follows:

$$d(x,y) = \sum_{i=1}^{n}|x_i - y_i| \quad (2)$$

Where the symbols denote:
d(x,y) = the distance separating the two points x and y
x_i y_i = the i-th attribute value
n = how many attributes are involved
$| x_i - y_i |$ = absolute value of the difference

Chebyshev distance

Chebyshev distance describes the separation of two points in a vector space as the single largest absolute gap found across their coordinate components. Put differently, it retains only the greatest deviation among the matching elements of the two vectors. For points X=(X1,X2,…,Xn) and Y=(Y1,Y2,…,Yn), the Chebyshev distance is written as follows:

$$d(x,y) = max_i(|x_i - y_i|) \quad (3)$$

Definition of the notation:
d(x,y) = the distance separating the two points x and y
x_i y_i = the i-th attribute value
$| x_i - y_i |$ = absolute value of the difference
max_i = the largest value across all dimensions

Hamming distance

Hamming distance evaluates dissimilarity by tallying how many positions hold differing values between two equally long vectors. Here it is brought in as an exploratory point of comparison, since it embodies a mismatch-oriented notion of neighborhood that contrasts with magnitude-oriented measures such as Euclidean, Manhattan, and Chebyshev. To keep the methodology sound, the Hamming-based SMOTE run relies on discretized or categorical indicator versions of the features. Consequently, the Hamming outcome is read as an illustration of categorical-style neighborhood formation instead of a straightforward substitute for continuous-distance measures.

$$d(x, y) = \sum_{i=1}^{n} \delta\,(x_i, y_i) \quad (4)$$

The symbols are read as:
d(x,y) = the distance separating the two points x and y
x_i y_i = the i-th attribute value
n = how many attributes are involved
$\delta(x_i,y_i)$ = an indicator flagging a mismatch

4. SMOTE

The Synthetic Minority Over-sampling Technique (SMOTE) balances a dataset by combining an enlargement of the minority class with a trimming of the majority class. During the trimming step, the count of majority-class records is lowered through a general routine so that the class distribution becomes more even (Nasaruddin et al., 2025). In the enlargement step, by contrast, SMOTE fabricates new records by exploiting how close instances sit to one another within the feature space: it selects minority-class points, locates their nearest neighbors, and places new samples along the line segments joining each point to those neighbors (Husain et al., 2025). This mechanism yields a more representative spread of the data and lets classifiers draw more general decision boundaries. As a result, applying SMOTE tends to curb overfitting and sharpens a model's ability to recognize patterns belonging to the minority class (Jain et al., 2025).

O is the original data set
P is the set of positive instances (minority class instances)
For each instance x in P
Find the k-nearest neighbors (minority class instances) to x in P
Get y by randomly selecting one of the k instances
Difference = x – y
Gap = a random number between 0 and 1
*n = x + difference * gap*
Add n to O
End

Source: Research Result (2025)
Figure 2. SMOTE Method

5. Classification Methods

Logistic Regression

Logistic Regression serves, in machine learning, to model binary outcome data for classification, chiefly by estimating the probability tied to a categorical dependent variable(Swana et al., 2022). The model equation of the *Logistic Regression* method is as follows:

$$P(y = 1|x) = \frac{1}{1+e-(\beta o+\sum_{i=1}^{n} \beta ixi)} \quad (5)$$

Meaning of each symbol:
P(y=1|x) = probability of positive class
σ(z) = the sigmoid (logistic) function
e = Euler's number (≈ 2.718)
β_0 = the intercept (bias) term
β_i = the weight of feature i
x_i = the value of feature i
n = how many attributes are involved

Decision Tree

A *Decision Tree* is a predictive structure in which every internal node applies a test to an attribute, each branch reflects a possible test outcome, and each leaf carries a class label. Because it is transparent and easy to follow, the technique is broadly adopted for classification. When a tree is grown, entropy becomes a key quantity, expressing the degree of impurity or uncertainty present in a dataset(Chandra et al., 2023). Entropy is computed with the formula below

$$Entropy(S) = \sum_{i=1}^{c} p_i log_2(p_i) \quad (6)$$

Symbol definitions:
S = the dataset under study

c = the count of distinct classes
p_i = the fraction of samples in class i
A = a chosen attribute (feature)
S_v = the data subset whose attribute equals v
|S| = the size of the dataset
$|S_v|$ = the size of that subset

k-Nearest Neighbor
As a supervised technique, k-Nearest Neighbor (k-NN) labels a new instance according to the class held by the majority of its k closest neighbors, where k denotes how many nearest neighbors enter the decision. In this work, k-NN adopts Euclidean distance as its default measure, so Equation (7) is included to make the classifier's neighbor-search step explicit. This classifier-level distance is separate from the distance measures examined within SMOTE, whose role is to create synthetic minority points during preprocessing. Choosing k carefully matters, because a value that is too small can be swayed by noise while a value that is too large may blur the class boundaries (Hairani et al., 2024). The default distance used by k-NN is written as follows:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (7)$$

Explanation of the notation:
d(x,y) = the distance separating the two points x and y
xi = the i-th attribute value of point x
yi = the i-th attribute value of point y
n = the total count of attributes (features)
$\sum$ = summation taken from i = 1 through n

Naïve Bayes
Naïve Bayes draws on statistical reasoning to estimate a class probability from earlier observations. Its core assumption is that whether a given feature appears within a class is independent of the remaining features(Lubis et al., 2024). The mathematical form of *Naïve Bayes* is expressed as follows:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \quad (8)$$

Notation and meaning:
P(C|X) = probability of class C for data X
P(C) = the the prior probability of class C of class C
P(X|C) = the likelihood term
P(X) = the evidence (marginal probability)
X_i = i-th feature
n = the total count of attributes (features)

RESULTS AND DISCUSSION

1. Data Sets

All experiments were run in Jupyter Notebook, with Python as the programming language. Ten binary imbalanced datasets were obtained from the KEEL repository for the analysis. These datasets differed considerably in their imbalance ratios, ranging from 1.86 to 15.8, and, being binary, each involved two classes.

2. MCC and G-Mean

MCC captures how strongly the predicted labels agree with the true ones by drawing on every cell of the confusion matrix. Because it folds true positives, true negatives, false positives, and false negatives into one coefficient, it fits imbalanced classification well. Its value lies between -1 and 1, and larger figures point to a better-balanced classification. The equation is given below:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (9)$$

Where the terms stand for:
TP (True Positive) = positive cases that the model classifies correctly as positive
TN (True Negative) = negative cases that the model classifies correctly as negative
FP (False Positive) = negative cases the model wrongly labels as positive
FN (False Negative) = positive cases the model wrongly labels as negative

The G-Mean reflects the equilibrium between sensitivity and specificity; a high value signals that the classifier copes well with the minority and the majority class alike. For this reason the present study treats G-Mean as an evaluation indicator rather than an accuracy figure. The corresponding equation is presented below:

$$G_{mean} = \sqrt{\left(\frac{TP}{TP+FN}\right) \times \left(\frac{TN}{TN+FP}\right)} \quad (10)$$

Description of the notation:
TP, TN, FP, FN : the four cells of the confusion matrix

3. Evaluation Test

Table 2. MCC and G-Mean Results for Binary Euclidean + SMOTE

Dataset	Naive Bayes		Decision Tree		Logistic Regression		KNN	
	MCC	G-Mean	MCC	G-Mean	MCC	G-Mean	MCC	G-Mean
ecoli-0-3-4-7_vs_5-6	0.54	0.75	0.37	0.57	0.69	0.70	0.54	0.57
ecoli-0-6-7_vs_5	0.85	0.86	0.88	0.89	0.80	0.81	0.69	0.70
ecoli-0-4-6_vs_5	0.77	0.88	0.46	0.62	0.68	0.70	0.72	0.75
ecoli-0-6-7_vs_3-5	0.62	0.93	0.31	0.49	0.83	0.91	0.80	0.81
ecoli-0-1_vs_2-3-5	0.80	0.81	0.47	0.62	0.77	0.88	0.85	0.86
ecoli4	0.56	0.94	0.70	0.70	0.68	0.75	0.55	0.57
pima	0.25	0.45	0.38	0.71	0.53	0.71	0.50	0.75
new-thyroid1	0.82	0.89	0.82	0.84	0.81	0.85	0.68	0.70
vehicle0	0.40	0.70	0.83	0.92	0.9	0.95	0.73	0.77
wisconsin	0.82	0.87	0.84	0.91	0.52	0.61	0.69	0.79
Average	**0.64**	**0.81**	**0.61**	**0.73**	**0.72**	**0.79**	**0.68**	**0.73**

Source: Research Result (2025)

As Table 2 indicates, the Euclidean + SMOTE setup reached its strongest average performance together with Logistic Regression, giving MCC = 0.72 and G-Mean = 0.79. Although Naive Bayes attained the top G-Mean (0.81), its MCC (0.64) trailed that of Logistic Regression. This pattern suggests that, under the Euclidean-based configuration, Logistic Regression offered a more evenly balanced agreement between the predicted and the true classes.

Table 3. MCC and G-Mean Results for Binary Manhattan + SMOTE

Dataset	Naive Bayes		Decision Tree		Logistic Regression		KNN	
	MCC	G-Mean	MCC	G-Mean	MCC	G-Mean	MCC	G-Mean
ecoli-0-3-4-7_vs_5-6	0.56	0.57	0.47	0.62	0.69	0.70	0.48	0.50
ecoli-0-6-7_vs_5	0.63	0.76	0.37	0.57	0.54	0.69	0.56	0.57
ecoli-0-4-6_vs_5	0.64	0.80	0.56	0.69	0.72	0.85	0.68	0.70
ecoli-0-6-7_vs_3-5	0.46	0.81	0.60	0.63	0.68	0.70	0.68	0.70
ecoli-0-1_vs_2-3-5	0.81	0.90	0.85	0.98	0.62	0.79	0.69	0.70
ecoli4	0.48	0.93	0.81	0.99	0.64	0.70	0.69	0.70
pima	0.19	0.32	0.38	0.69	0.57	0.77	0.32	0.62
new-thyroid1	0.79	0.97	0.91	0.92	0.87	0.90	0.75	0.77

vehicle0	0.41	0.68	0.83	0.92	0.85	0.92	0.78	0.86
wisconsin	0.80	0.87	0.85	0.91	0.58	0.67	0.87	0.91
Average	**0.58**	**0.76**	**0.66**	**0.79**	**0.68**	**0.77**	**0.65**	**0.70**

Source: Research Result (2025)

Table 3 shows that, with Manhattan + SMOTE, the best average MCC came from Logistic Regression (MCC = 0.68), whereas the best G-Mean was delivered by the Decision Tree (0.79). This outcome implies that Manhattan distance can reinforce different classifier strengths: Logistic Regression excels at overall correlation, while the Decision Tree is better at balancing sensitivity against specificity.

Table 4. MCC and G-Mean Results for Binary Chebyshev + SMOTE

Dataset	Naive Bayes		Decision Tree		Logistic Regression		KNN	
	MCC	**G-Mean**	**MCC**	**G-Mean**	**MCC**	**G-Mean**	**MCC**	**G-Mean**
ecoli-0-3-4-7_vs_5-6	0.45	0.68	0.84	0.92	0.68	0.70	0.69	0.70
ecoli-0-6-7_vs_5	0.69	0.70	0.77	0.88	0.75	0.77	0.60	0.63
ecoli-0-4-6_vs_5	0.85	0.98	0.56	0.57	0.92	0.93	0.85	0.86
ecoli-0-6-7_vs_3-5	0.72	0.75	0.85	0.86	0.80	0.81	0.68	0.63
ecoli-0-1_vs_2-3-5	0.69	0.70	0.69	0.87	0.75	0.77	0.75	0.77
ecoli4	0.49	0.91	0.78	0.88	0.70	0.86	0.81	0.81
pima	0.22	0.45	0.33	0.66	0.41	0.63	0.49	0.75
new-thyroid1	0.87	0.97	0.92	0.94	0.91	0.98	0.68	0.70
vehicle0	0.43	0.72	0.74	0.88	0.77	0.85	0.63	0.71
wisconsin	0.84	0.89	0.93	0.96	0.43	0.51	0.62	0.74
Average	**0.63**	**0.78**	**0.74**	**0.84**	**0.71**	**0.78**	**0.68**	**0.73**

Source: Research Result (2025)

Table 4 makes clear that Chebyshev + SMOTE performed best on average when teamed with the Decision Tree, recording MCC = 0.74 and G-Mean = 0.84. This suggests that the maximum-coordinate gap emphasized by Chebyshev distance generates synthetic points that align well with the hierarchical splitting logic of the Decision Tree across the binary imbalanced datasets examined.

Table 5. MCC and G-Mean Results for Binary Hamming + SMOTE

Dataset	Naive Bayes		Decision Tree		Logistic Regression		KNN	
	MCC	**G-Mean**	**MCC**	**G-Mean**	**MCC**	**G-Mean**	**MCC**	**G-Mean**
ecoli-0-3-4-7_vs_5-6	0.54	0.79	0.72	0.85	0.80	0.81	0.75	0.77
ecoli-0-6-7_vs_5	0.80	0.98	0.54	0.75	0.68	0.70	0.69	0.70
ecoli-0-4-6_vs_5	0.46	0.78	0.85	0.86	0.85	0.86	0.80	0.81
ecoli-0-6-7_vs_3-5	0.31	0.49	0.79	0.81	0.88	0.89	0.42	0.44
ecoli-0-1_vs_2-3-5	0.46	0.50	0.39	0.61	0.92	0.93	0.73	0.75
ecoli4	0.26	0.91	0.76	0.98	0.72	0.88	0.85	0.86
pima	0.35	0.68	0.20	0.59	0.42	0.71	0.34	0.68
new-thyroid1	0.91	0.98	0.60	0.63	0.69	0.87	0.34	0.37

vehicle0	0.32	0.67	0.82	0.91	0.67	0.70	0.37	0.42
wisconsin	0.80	0.88	0.78	0.89	0.70	0.75	0.50	0.58
Average	**0.52**	**0.77**	**0.65**	**0.79**	**0.73**	**0.81**	**0.58**	**0.64**

Source: Research Result (2025)

According to Table 5, Hamming + SMOTE was most effective alongside Logistic Regression, yielding MCC = 0.73 and G-Mean = 0.81. While competitive, this figure should be read with the understanding that Hamming distance operates on a mismatch-based representation. Consequently, Hamming + SMOTE is best suited to cases where features can be cast as categorical or discretized indicators.

Table 6. Comparative Summary of Chebyshev + SMOTE and Hamming + SMOTE

Classifier	Chebyshev MCC	Chebyshev G-Mean	Hamming MCC	Hamming G-Mean	Dominant Result
Naive Bayes	0.63	0.78	0.52	0.77	Chebyshev
Decision Tree	0.74	0.84	0.65	0.79	Chebyshev
Logistic Regression	0.71	0.78	0.73	0.81	Hamming
KNN	0.68	0.73	0.58	0.64	Chebyshev
Average across classifiers	0.69	0.78	0.62	0.75	Chebyshev

Source: Research Result (2025)

Table 6 sets Chebyshev + SMOTE against Hamming + SMOTE directly, as these were the two setups with the strongest average scores in Tables 4 and 5. Chebyshev + SMOTE comes out ahead for Naive Bayes, Decision Tree, and KNN, whereas Hamming + SMOTE leads only for Logistic Regression. The averaged figures further reveal that Chebyshev + SMOTE is the more consistent option across classifiers, posting a higher mean MCC (0.69) and G-Mean (0.78) than Hamming + SMOTE, whose averages stood at MCC = 0.62 and G-Mean = 0.75.

Viewed as a whole, the experiments show that the distance measure adopted within SMOTE reshapes the layout of the synthetic minority points and thereby influences classifier behavior. The most balanced result came from Chebyshev + SMOTE paired with the Decision Tree, with Hamming + SMOTE combined with Logistic Regression following behind. Yet, because Hamming distance depends on categorical or discretized representations, Chebyshev distance appears the more dependable choice for numerical binary imbalanced datasets in this experimental context. These results lend further weight to the view that SMOTE ought not to be applied as a rigid, one-size-fits-all preprocessing step; its distance measure should instead be chosen in light of the data's nature and the classifier's tendencies.

CONCLUSION

The study concludes that the distance measure employed within SMOTE exerts a measurable effect on binary imbalanced classification. Its principal contribution lies in a structured comparison of the Euclidean, Manhattan, Chebyshev, and Hamming measures inside SMOTE, conducted over ten publicly available binary imbalanced datasets and four learning algorithms. The strongest configuration overall was the Decision Tree paired with Chebyshev + SMOTE, which reached MCC = 0.74 and G-Mean = 0.84. Hamming + SMOTE together with Logistic Regression was likewise competitive (MCC = 0.73; G-Mean = 0.81), though its usefulness hinges on a suitable categorical or discretized representation of the features. The evidence therefore indicates that Chebyshev distance yields a more stable SMOTE neighborhood structure for the numerical binary datasets studied. MCC and G-Mean are best read as evaluation indicators rather than accuracy scores, since each captures a distinct facet of model behavior under imbalanced class distributions. For future work, it would be valuable to add explicit baseline runs without SMOTE, to compare further SMOTE variants such as Borderline-SMOTE, SMOTE-ENN, and ADASYN, to adopt repeated stratified cross-validation, and to test statistical significance in order to confirm whether the differences observed hold across a wider range of datasets and domains.

REFERENCES

Asniar, Maulidevi, N. U., & Surendro, K. (2022). SMOTE-LOF for noise identification in imbalanced data classification. Journal of King Saud University - Computer and Information Sciences, 34(6), 3413-3423. https://doi.org/10.1016/j.jksuci.2021.01.014

Carvalho, M., Pinho, A. J., & Bras, S. (2025). Resampling approaches to handle class imbalance: A review from a data perspective. Journal of Big Data, 12, Article 71. https://doi.org/10.1186/s40537-025-01119-4

Chandra, W., Suprihatin, B., & Resti, Y. (2023). Median-KNN Regressor-SMOTE-Tomek Links for handling missing and imbalanced data in air quality prediction. Symmetry, 15(4), Article 887. https://doi.org/10.3390/sym15040887

Chen, W., Yang, K., Yu, Z., Shi, Y., & Chen, C. L. P. (2024). A survey on imbalanced learning: Latest research, applications and future directions. Artificial Intelligence Review, 57, Article 137. https://doi.org/10.1007/s10462-024-10759-6

Dai, Q., Liu, J., & Zhao, J. L. (2023). Distance-based arranging oversampling technique for imbalanced data. Neural Computing and Applications, 35(2), 1323-1342. https://doi.org/10.1007/s00521-022-07828-8

Cruz Huayanay, A., Bazan, J. L., & Russo, C. M. (2025). Performance of evaluation metrics for classification in imbalanced data. Computational Statistics, 40(3), 1447-1473. https://doi.org/10.1007/s00180-024-01539-5

Elreedy, D., Atiya, A. F., & Kamalov, F. (2024). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. Machine Learning, 113(7), 4903-4923. https://doi.org/10.1007/s10994-022-06296-4

Feng, S., Keung, J., Zhang, P., Xiao, Y., & Zhang, M. (2022). The impact of the distance metric and measure on SMOTE-based techniques in software defect prediction. Information and Software Technology, 142, Article 106742. https://doi.org/10.1016/j.infsof.2021.106742

Hairani, H., Widiyaningtyas, T., & Prasetya, D. D. (2024). Addressing class imbalance of health data: A systematic literature review on modified Synthetic Minority Oversampling Technique (SMOTE) strategies. International Journal on Informatics Visualization, 8(3), 1310-1318. https://doi.org/10.62527/joiv.8.3.2283

Hemmatian, J., Hajizadeh, R., & Nazari, F. (2025). Addressing imbalanced data classification with Cluster-Based Reduced Noise SMOTE. PLoS ONE, 20(2), Article e0317396. https://doi.org/10.1371/journal.pone.0317396

Husain, G., Nasef, D., Jose, R., Mayer, J., Bekbolatova, M., Devine, T., & Toma, M. (2025). SMOTE vs. SMOTEENN: A study on the performance of resampling algorithms for addressing class imbalance in regression models. Algorithms, 18(1), Article 37. https://doi.org/10.3390/a18010037

Imani, M., Joudaki, M., Bagheri, A., & Arabnia, H. R. (2026). Why ROC-AUC is misleading for highly imbalanced data: In-depth evaluation of MCC, F2-score, H-measure, and AUC-based metrics across diverse classifiers. Technologies, 14(1), Article 54. https://doi.org/10.3390/technologies14010054

Jain, A., Dubey, A. K., Khan, S., Panwar, A., Alkhatib, M., & Alshahrani, A. M. (2025). A PSO weighted ensemble framework with SMOTE balancing for student dropout prediction in smart education systems. Scientific Reports, 15, Article 17463. https://doi.org/10.1038/s41598-025-97506-1

Li, Y., Yang, Y., Song, P., Duan, L., & Ren, R. (2025). An improved SMOTE algorithm for enhanced imbalanced data classification by expanding sample generation space. Scientific Reports, 15, Article 23521. https://doi.org/10.1038/s41598-025-09506-w

Lubis, A., Irawan, Y., Junadhi, J., & Defit, S. (2024). Leveraging K-Nearest Neighbors with SMOTE and boosting techniques for data imbalance and accuracy improvement. Journal of Applied Data Sciences, 5(4), 1625-1638. https://doi.org/10.47738/jads.v5i4.343

Lyu, J., Yang, J., Su, Z., & Zhu, Z. (2025). LD-SMOTE: A novel local density estimation-based oversampling method for imbalanced datasets. Symmetry, 17(2), Article 160. https://doi.org/10.3390/sym17020160

Maldonado, S., Vairetti, C., Fernandez, A., & Herrera, F. (2022). FW-SMOTE: A feature-weighted oversampling approach for imbalanced classification. Pattern Recognition, 124, Article 108511. https://doi.org/10.1016/j.patcog.2021.108511

Matharaarachchi, S., Domaratzki, M., & Muthukumarana, S. (2024). Enhancing SMOTE for imbalanced data with abnormal minority instances. Machine Learning with Applications, 18, Article 100597. https://doi.org/10.1016/j.mlwa.2024.100597

Misdram, M., Noersasongko, E., Purwanto, P., Muljono, M., & Pamuji, F. Y. (2023). Gaussian Based-SMOTE method for handling imbalanced small datasets. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, 9(4), 973-982. https://doi.org/10.26555/jiteki.v9i4.26881

Mukherjee, A., Qazani, M. R. C., Rana, B. M. J., Akter, S., Mohajerzadeh, A., Sathi, N. J., Ali, L. E., Khan, M. S., & Asadi, H. (2025). SMOTE-ENN resampling technique with Bayesian optimization for multi-class classification of dry bean varieties. Applied Soft Computing, 181, Article 113467. https://doi.org/10.1016/j.asoc.2025.113467

Nasaruddin, N., Masseran, N., Idris, W. M. R., & Ul-Saufie, A. Z. (2025). A SMOTE PCA HDBSCAN approach for enhancing water quality classification in imbalanced datasets. Scientific Reports, 15, Article 13059. https://doi.org/10.1038/s41598-025-97248-0

Pei, W., Xue, B., Zhang, M., & Shang, L. (2024). A survey on unbalanced classification: How can evolutionary computation help? IEEE Transactions on Evolutionary Computation, 28(2), 353-373. https://doi.org/10.1109/TEVC.2023.3257230

Rezvani, S., & Wang, X. (2023). A broad review on class imbalance learning techniques. Applied Soft Computing, 143, Article 110415. https://doi.org/10.1016/j.asoc.2023.110415

Swana, E. F., Doorsamy, W., & Bokoro, P. (2022). Tomek Link and SMOTE approaches for machine fault classification with an imbalanced dataset. Sensors, 22(9), Article 3246. https://doi.org/10.3390/s22093246

Taskiran, S. F., Turkoglu, B., Kaya, E., & Asuroglu, T. (2025). A comprehensive evaluation of oversampling techniques for enhancing text classification performance. Scientific Reports, 15, Article 21631. https://doi.org/10.1038/s41598-025-05791-7

Wainer, J. (2024). An empirical evaluation of imbalanced data strategies from a practitioner's point of view. Expert Systems with Applications, 256, Article 124863. https://doi.org/10.1016/j.eswa.2024.124863

Widiyaningtyas, T., Hairani, H., Prasetya, D. D., Pujianto, U., & Caesarendra, W. (2025). A modified SMOTE with noise filtering and Manhattan distance metric approach to address imbalanced health datasets. Engineering, Technology and Applied Science Research, 15(4), 25452-25459. https://doi.org/10.48084/etasr.11925

Wong, T. T., & Chung, P. C. (2025). A consistency analysis on four evaluation metrics for classifying imbalanced data. Knowledge and Information Systems, 67, 10639-10656. https://doi.org/10.1007/s10115-025-02544-w

Yulian Pamuji, F., Muslikh, A. R., Arief, R. M., & Muti, D. (2024). Komparasi metode Mean dan KNN Imputation dalam mengatasi missing value pada dataset kecil. Jurnal Informatika Polinema, 10(2), 257-264. https://doi.org/10.33795/jip.v10i2.5031

Zhang, Y., Deng, L., & Wei, B. (2024). Imbalanced data classification based on improved Random-SMOTE and feature standard deviation. Mathematics, 12(11), Article 1709. https://doi.org/10.3390/math12111709