4 Experimental Analysis

This section presents the experimental evaluation of the proposed methodology on both synthetic [98] and public data sets [100] with a comparison to some existing prominent methodologies.

For this, evaluation measures; precision, recall, F1- score and area under curve (AUC) are used. Also, the accuracy versus computational time of each methodology is presented to compare the efficiency of the proposed methodology. The data sets used for the experimental analysis are briefly discussed in the following subsection.

4.1 Data Sets

This subsection presents both the synthetic and public data sets, which are used in the experimental analysis of the proposed methodology for a comparison with some existing prominent methodologies. For this experimental analysis, 12 synthetic data sets [98] are used, which are mentioned in Table 2. Fig. 10 shows the distributions of these data sets.

Additionally, 10 public data sets [100] are also used for the experimental analysis of the proposed methodology for a comparison with some existing prominent methodologies, which are mentioned with their details in Table 3 .

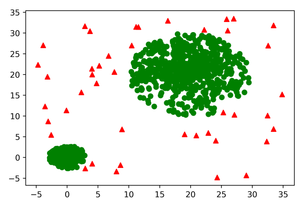
Further, the existing promonent methodologies used for comparison are briefly discussed in the following subsection.

Table 2: Synthetic Data Sets

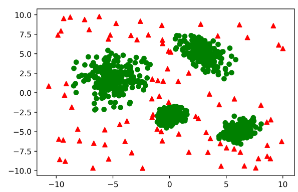
Dataset name	Instances	Outliers	Dimensions
SD01	1043	43	2
SD02	1000	85	2
SD03	1039	41	2
SD04	1641	45	2
SD05	876	77	2
SD06	1372	72	2
SD07	1037	37	2
SD08	2259	159	2
SD09	1034	36	2
SD10	2042	64	2
SD11	1020	26	2
SD12	1242	50	2

Table 3: Public Data Sets

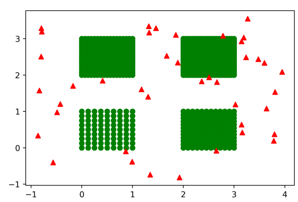
Dataset name	Instances	Dimensions
PD1: Glass	214	9
PD2: Ionosphere	351	34
PD3: Cardiotocography	2126	23
PD4: Lymphography	148	18
PD5: Vowels	1455	12
PD6: Diabetes	768	8
PD7: Breast cancer	682	10
PD8: Wheat seeds	210	7
PD9: Iris	150	4
PD10: Haberman	306	3



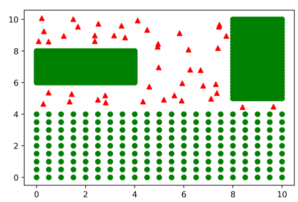
(a) SD01



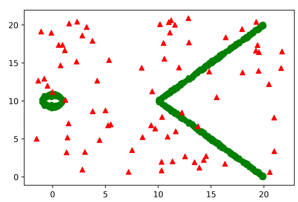
(b) SD02



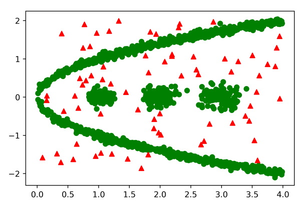
(c) SD03



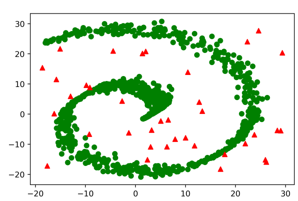
(d) SD04



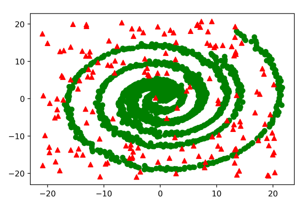
(e) SD05



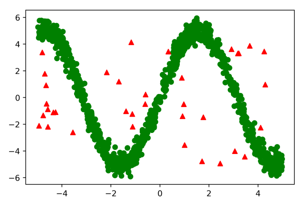
(f) SD06



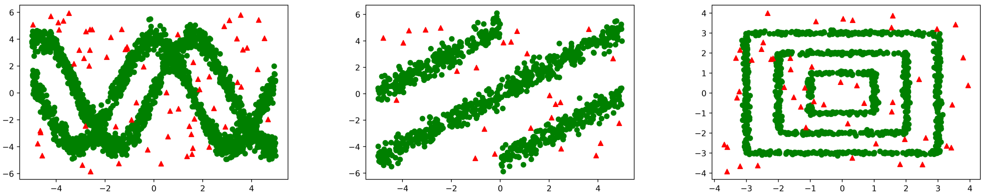
(g) SD07



(h) SD08



(i) SD09



(j) SD10 (k) SD11 (l) SD12

Fig. 10: The distribution of synthetic data sets.

4.2 Methods Under Comparison

This subsection presents some prominent outlier detection methodologies used for comparative analysis with the proposed methodology. These methods are KNN [66], MOD [70], LGOD [71], LOF [49], LDF [53], RDOS [56], RDOF [73], DCROD [19], NaNOD [75], HDIOD [8], and LDS [54]. This comparative analysis is done to assess the effectiveness of the proposed methodology.

The next subsection presents the evaluation measures used to evaluate the performance of the proposed methodology.

4.3 Evaluation Measures

This subsection presents the evaluation measures: Area Under Curve (AUC), Recall, Precision, and F1-score, which are used to assess the performance of the proposed methodology as compared to the existing prominent methodologies mentioned earlier. The Area Under Curve (AUC) is discussed briefly in the following subsection.

Area Under Curve (AUC)

In classification problems, it is often to have a single evaluation measure to assess the performance of a methodology. For this, the area under the receiver operating characteristic (ROC) curve is extensively used in conjunction with the Neyman-Pearson method in signal detection theory [101, 102]. It is a cross-validated measure to assess the overall accuracy of the classifier methodology and is also used in the field of outlier detection methodologies [99].

The AUC curve is generated by evaluating all possible classification thresholds. The AUC curve captures the trade-off between the true positive rate (correctly identified anomalies) and the false positive rate (normal instances misclassified as anomalies). The AUC gives a single value summarizing the ROC curve, ranging from 0 to 1. A methodology with random performance yields an AUC near 0.5, and an AUC approaching 1 indicates that methodology is highly effective in the detection of outliers. Further, another evaluation measure, i.e., recall, is discussed in the following subsection.

Recall

In machine learning methodologies, recall is an evaluation measure that measures how well a model detects all the positive instances in the given data set [103]. It is also referred to as the true positive rate or sensitivity that calculates the proportion of correctly identified true positives out of all actual positive cases.

Hence, recall can also be used as an evaluation measure in the case of outlier detection methodologies to determine how well a methodology detects the outliers correctly. It is particularly valuable in situations where missing true outliers (false negatives) is costly or undesirable. The recall is defined as the ratio of true positives to the total number of actual outliers. The total number of actual outliers includes both correctly identified outliers (true positives) and those that the model failed to detect (false negatives) [104]. The formula for Recall is:

$$Recall = \frac{TP}{TP + FN}. \tag{11}$$

This evaluation measure focuses on how effectively the model captures all relevant outlier instances, without considering how many false alarms (false positives) it generates. A high value of recall indicates that the model successfully detects most of the true outliers in the data set. Next, the evaluation measure, i.e., precision, is discussed in the following subsection.

Precision

In machine learning methodologies, precision is an evaluation measure, i.e., used to measure the accuracy of positive predictions made by a model. This evaluation measure specifically addresses how many of the predicted positive cases are actually correct [103]. It becomes especially important in cases where false positives carry significant consequences. Precision is defined as the proportion of correctly detected outliers (true positives) to all instances that the methodology detected as outliers, which includes both correct (true positives (TP)) and incorrect (false positives (FP)) detections [105]. Mathematically, it can be expressed as:

$$Precision = \frac{TP}{TP + FP}. \tag{12}$$

This means that precision quantifies how many of the predicted outliers are actually true outliers. Since it does not consider true negatives, it is particularly suited for applications where the correctness of detected outliers is more important than detecting all normal instances. High precision indicates that the methodology does few false positive predictions. Further, the other evaluation measure, i.e., the F1 score, is discussed in the following subsection.

F1-Score

In statistical analysis of binary classification and information retrieval systems, the evaluation measure F1-score is the measure of predictive performance. This evaluation measure is derived from the recall and precision, i.e., the harmonic mean of both [104]. Thus, this evaluation measure symmetrically represents both precision and recall in one measure.

Hence, in outlier detection methodologies, this measure is used to assess a methodology performance when detecting outliers. It is especially useful in scenarios with imbalanced data, where outliers are rare compared to normal instances. Since this F1-score is the harmonic mean of precision and recall, it emphasizes the importance of both correctly detecting outliers and minimizing false detections [106]. The formula for F1-score is given as:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{13}$$

This evaluation measure ensures that the high score can only be achieved when

both precision and recall are reasonably high, making it a reliable indicator of the overall effectiveness of an outlier detection methodology.

Next, the experimental results using these evaluation measures:area under curve (AUC), recall, precision, and F1-score are discussed in the following subsection.

4.4 Experimental Results on Synthetic Data Sets

This section presents the experimental analysis of the proposed methodology using the synthetic data sets [98] based on the evaluation measures such as AUC, recall, precision, and F1-score in comparison with some existing prominent methodologies mentioned in subsection 4.2. Here, the experimental analysis for the proposed methodology is done using the values of parameters $K = 10$, $M = 3$, and $\mathcal{T} = 3\mu + \frac{\sigma}{\sqrt{n}}$.

First, the experimental analysis based on the evaluation measure AUC is summarized in Table 4, with the final row presenting the average AUC values of all synthetic data sets mentioned in Table 2 for each prominent methodology. This row is highlighted by bold letters.

Here, the comparative analysis of each prominent methodology is done based on their average AUC values. Hence, from Table 4, it can be observed that the KNN methodology achieves the highest AUC values for the data sets SD01, SD02, and SD08.

The LGOD methodology achieves the highest AUC value for the data set SD03, and the LDS methodology achieves the highest AUC values for the data sets SD05, SD06, SD10, SD11, and SD12, while the proposed methodology achieves the highest AUC values for the data sets SD04, SD07, and SD09. The proposed methodology achieves the highest average AUC value, which shows its effectiveness as compared to existing prominent methodologies.

Further, Fig. 11 provides a visual representation of the analysis for data set SD08. Hence, from this Fig. 11, it can be noticed that the methodologies KNN, LGOD, MOD, and DCROD misclassify too many data points as outliers; those are the normal data points. In the comparison of misclassification rate, this rate is low for the LOF, RDOS, and NaNOD methodologies. Moreover, the RDOF and HDIOD methodologies are unable to detect the outliers properly and can only detect two or three outliers. Also, the LDF and LDS methodologies are unable to detect the outliers properly, while the proposed methodology detects all the outliers properly.

Next, the experimental analysis of recall, precision, and F1-score on synthetic data sets [19] are summarized in Table 5, 6, and 7, respectively. In these tables, the final row presents the average values for each methodology, which are highlighted by bold letters. The comparison of various methodologies is analyzed based on their average values.

From Table 5, 6, and 7, it can be observed that the proposed methodology achieves the highest average values for recall, precision, and F1-score, which shows its effectiveness as compared to the existing prominent methodologies.

Table 4: AUC values of methodologies on synthetic data sets and the average value of each methodology.

Datasets	**KNN**	**LGOD**	**MOD**	**LOF**	**LDF**	**RDOS**	**RDOF**	**DCROD**	**NaNOD**	**HDIOD**	**LDS**	**Proposed Methodology**
SD01	0.9933	0.925	0.9523	0.8130	0.7308	0.4961	0.6612	0.8390	0.6483	0.8372	0.9752	0.9651
SD02	0.9743	0.9738	0.9306	0.6783	0.6330	0.4964	0.6571	0.7646	0.6014	0.5118	0.8529	0.9464
SD03	0.931	0.9883	0.8495	0.6907	0.5609	0.6585	0.9390	0.7647	0.7969	0.8659	1.0	0.9390
SD04	0.9058	0.9554	0.8911	0.6916	0.7172	0.6222	0.5555	0.7522	0.8058	0.8753	0.9870	0.9876
SD05	0.9716	0.9688	0.9051	0.6646	0.6313	0.4921	0.5773	0.7462	0.6654	0.5	1.0	0.9870
SD06	0.9655	0.9461	0.8188	0.4807	0.4926	0.5204	0.5069	0.7570	0.8094	0.6042	0.9919	0.9796
SD07	0.832	0.8607	0.6672	0.9364	0.9299	0.4840	0.6736	0.6697	0.8178	0.7027	0.9159	0.9445
SD08	0.9074	0.8912	0.891	0.7268	0.6234	0.4905	0.5157	0.7558	0.8309	0.5283	0.6275	0.8559
SD09	0.95	0.9551	0.8259	0.8193	0.7583	0.4988	0.6616	0.7892	0.8705	0.9306	0.9861	0.9553
SD10	0.9231	0.8964	0.7963	0.8046	0.7222	0.4923	0.6318	0.7266	0.7822	0.8506	0.9993	0.9279
SD11	0.9519	0.9558	0.7827	0.9282	0.8538	0.4844	0.7677	0.7558	0.9408	0.8644	0.9995	0.9363
SD12	0.6748	0.6855	0.6439	0.8382	0.8020	0.4949	0.5	0.6960	0.8541	0.78	1.0	0.9196
Average	**0.9151**	**0.9168**	**0.8295**	**0.7560**	**0.7046**	**0.5192**	**0.6372**	**0.7514**	**0.7853**	**0.7376**	**0.9446**	**0.9454**

Table 5: Recall values of methodologies on synthetic data sets and the average value of each methodology.

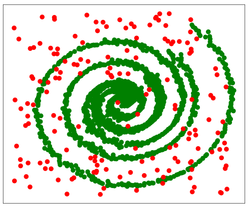
Datasets	KNN	LGOD	MOD	LOF	LDF	RDOS	RDOF	DCROD	NaNOD	HDIOD	LDS	Proposed Methodology
SD01	0.12	0.23	0.19	0.65	0.51	0.02	0.81	0.74	0.33	0.67	0.79	0.93
SD02	0.24	0.31	0.32	0.38	0.33	0.02	0.79	0.64	0.27	0.02	0.52	0.91
SD03	0.13	0.33	0.17	0.41	0.12	0.32	0.73	0.61	0.61	0.73	0.53	0.88
SD04	0.41	0.21	0.27	0.42	0.18	0.24	0.58	0.62	0.62	0.76	0.53	0.98
SD05	0.18	0.24	0.56	0.35	0.27	0.01	0.16	0.53	0.39	0.00	0.75	0.97
SD06	0.02	0.11	0.21	0.01	0.03	0.07	0.88	0.61	0.64	0.21	0.58	0.97
SD07	0.42	0.39	0.00	0.89	0.89	0.00	0.68	0.46	0.65	0.41	0.78	0.92
SD08	0.27	0.02	0.11	0.47	0.25	0.01	0.03	0.60	0.69	0.06	0.66	0.72
SD09	0.04	0.29	0.06	0.67	0.53	0.03	0.78	0.67	0.75	0.86	0.77	0.92
SD10	0.12	0.32	0.11	0.64	0.45	0.02	0.83	0.53	0.58	0.70	0.75	0.86
SD11	0.45	0.26	0.00	0.88	0.73	0.00	0.54	0.62	0.88	0.73	0.80	0.88
SD12	0.03	0.01	0.00	0.70	0.62	0.02	0.16	0.46	0.72	0.56	0.82	0.84
Average	**0.2025**	**0.2266**	**0.1666**	**0.5391**	**0.4091**	**0.0633**	**0.5808**	**0.5908**	**0.5941**	**0.4758**	**0.69**	**0.8983**

Table 6: Precision values of methodologies on synthetic data sets and the average value of each methodology.

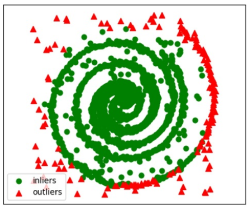
Datasets	**KNN**	**LGOD**	**MOD**	**LOF**	**LDF**	**RDOS**	**RDOF**	**DCROD**	**NaNOD**	**HDIOD**	**LDS**	**Proposed Methodology**
SD01	0.12	0.21	0.29	0.52	0.31	0.03	0.71	0.33	0.33	0.61	0.62	1
SD02	0.27	0.41	0.32	0.62	0.33	0.07	0.52	0.36	0.27	0.53	0.77	0.87
SD03	0.05	0.39	0.68	0.33	0.52	0.04	0.66	0.17	0.61	0.62	0.45	1
SD04	0.07	0.14	0.06	0.23	0.43	0.01	0.72	0.13	0.39	0.54	0.54	0.92
SD05	0.39	0.42	0.48	0.61	0.49	0.04	0.82	0.56	0.64	0.49	0.82	1
SD06	0.21	0.03	0.27	0.01	0.02	0.12	0.52	0.26	0.65	0.35	0.48	0.80
SD07	0.04	0.42	0.01	0.43	0.27	0.00	0.51	0.12	0.65	0.42	0.45	0.53
SD08	0.29	0.39	0.21	0.66	0.70	0.03	0.61	0.35	0.69	0.32	0.76	0.91
SD09	0.39	0.45	0.08	0.46	0.63	0.01	0.44	0.21	0.75	0.54	0.51	0.85
SD10	0.41	0.38	0.58	0.40	0.62	0.02	0.50	0.18	0.58	0.53	0.50	0.89
SD11	0.41	0.04	0.01	0.45	0.45	0.00	0.52	0.13	0.58	0.39	0.44	0.66
SD12	0.33	0.41	0.02	0.56	0.52	0.03	0.62	0.22	0.62	0.48	0.57	0.98
Average	**0.2483**	**0.3075**	**0.2508**	**0..44**	**0.4408**	**0.0333**	**0.5958**	**0.2516**	**0.5633**	**0.4825**	**0.5758**	**0.8675**

Table 7: F1-Score values of methodologies on synthetic data sets and the average value of each methodology.

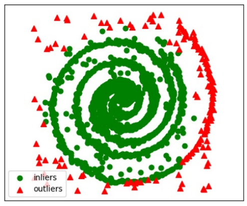
Datasets	**KNN**	**LGOD**	**MOD**	**LOF**	**LDF**	**RDOS**	**RDOF**	**DCROD**	**NaNOD**	**HDIOD**	**LDS**	**Proposed Methodology**
SD01	0.42	0.32	0.31	0.58	0.38	0.03	0.81	0.45	0.33	0.81	0.70	0.96
SD02	0.29	0.39	0.48	0.47	0.33	0.03	0.69	0.46	0.27	0.05	0.61	0.89
SD03	0.23	0.02	0.29	0.37	0.22	0.48	0.79	0.26	0.61	0.85	0.69	0.94
SD04	0.12	0.31	0.10	0.30	0.30	0.39	0.64	0.21	0.62	0.78	0.69	0.95
SD05	0.26	0.33	0.72	0.45	0.35	0.02	0.27	0.55	0.39	0.02	0.78	0.99
SD06	0.11	0.14	0.24	0.01	0.02	0.09	0.85	0.36	0.64	0.34	0.52	0.88
SD07	0.02	0.12	0.01	0.74	0.41	0.02	0.60	0.20	0.65	0.58	0.57	0.67
SD08	0.24	0.11	0.15	0.55	0.38	0.02	0.06	0.44	0.69	0.11	0.70	0.80
SD09	0.01	0.02	0.07	0.55	0.58	0.03	0.57	0.32	0.75	0.93	0.62	0.88
SD10	0.0	0.0	0.19	0.49	0.53	0.02	0.63	0.27	0.58	0.80	0.60	0.87
SD11	0.18	0.30	0.00	0.60	0.56	0.00	0.65	0.22	0.88	0.81	0.57	0.75
SD12	0.24	0.01	0.01	0.62	0.62	0.02	0.25	0.30	0.72	0.72	0.67	0.90
Average	**0.1766**	**0.1725**	**0.2141**	**0.4775**	**0.39**	**0.0958**	**0.5675**	**0.3366**	**0.5941**	**0.5666**	**0.6433**	**0.8733**



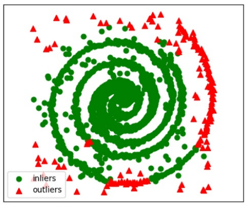
(a) Actual data



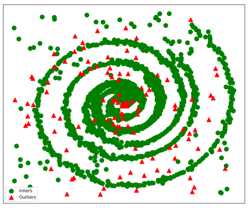
(b) KNN



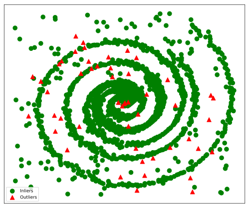
(c) LGOD



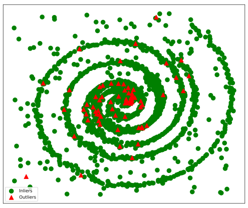
(d) MOD



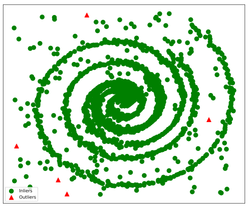
(e) LOF



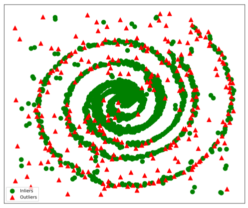
(f) LDF



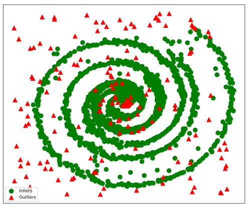
(g) RDOS



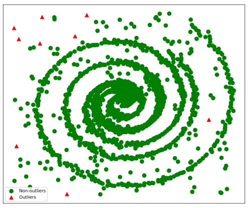
(h) RDOF



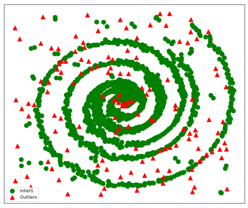
(i) DCROD



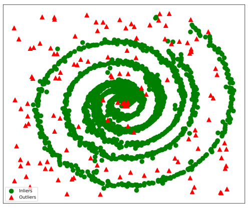
(j) NaNOD



(k) HDIOD



(l) LDS

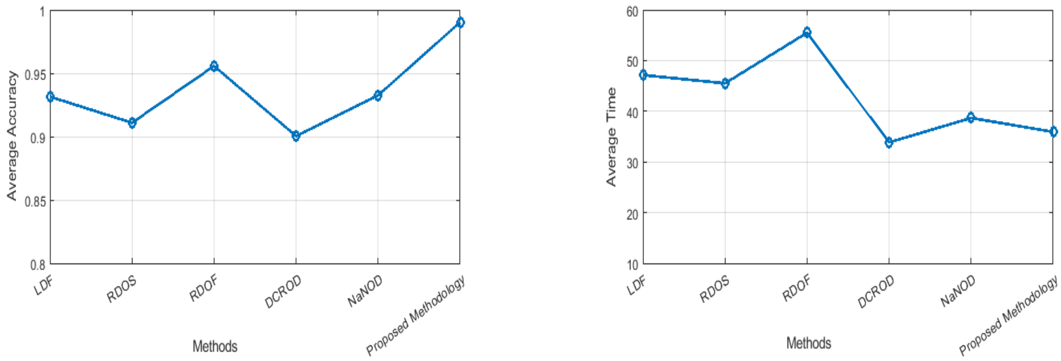


(m) Proposed

Fig. 11: Experimental results on synthetic data set SD08.

4.5 Accuracy Versus Computational Time for Synthetic Data Sets

This subsection presents the accuracy versus computational time of each existing prominent methodology, including the proposed methodology using the synthetic data sets, to provide insight into the efficiency of these. For this, the average accuracy of the methodologies such as LDF [53], RDOS [56], RDOF [73], DCROD [19], NaNOD [75], and the proposed methodology is calculated for the synthetic data sets, i.e., shown in Fig. 12(a). Similarly, average time is calculated for these methodologies for the synthetic data sets and shown in Fig. 12(b).



(a) (b)

Fig. 12: Average Accuracy and Time for different methodologies on Synthetic Data sets.

From Fig. 12, we observe that the proposed methodology has the highest accuracy as compared to these existing prominent methodologies. Moreover, the proposed methodology takes less time as compared to these existing prominent methodologies except the DCROD [19] methodology with less difference. Hence, the analysis of accuracy versus computational time for synthetic data sets shows the clear advantage of the proposed methodology has higher accuracy than existing prominent methodologies. Also, the proposed methodology has the advantage that it takes less time than existing prominent methodologies except the DCROD [19] methodology. However, this DCROD [19] methodology has limitations; it is sensitive to the parameter K and the choice of Gaussian kernel bandwidth [76].

4.6 Experimental Results on Public Data Sets

This section presents the experimental analysis of the proposed methodology using the public data sets [100] based on the evaluation measures such as AUC, recall, precision, and F1-score in comparison with some existing prominent methodologies mentioned in subsection 4.2. Here, the experimental analysis for the proposed methodology is done using the values of parameters $K = 10$, $M = 3$, and $\mathcal{T} = 3\mu + \frac{\sigma}{\sqrt{n}}$.

First, we summarized the experimental analysis based on the evaluation measure AUC in Table 8 for the public data sets [100]. In this Table 8, the average AUC values of each methodology are highlighted in bold letters in the final row.

Here, Table 8 demonstrates that different methodologies achieve the highest AUC values for different data sets, which means each method excels in specific cases. For instance, the KNN methodology achieves the best performance on the PD3 data set, LGOD achieves it on PD1, and LDF achieves it on PD8. The RDOF methodology performs best on PD8 and PD9, while DCROD achieves the highest AUC on PD5. Additionally, NaNOD shows superior results on PD8, PD9, and PD10, and HDIOD dominates on PD2, PD4, PD6, PD7, and PD9. Despite the proposed methodology achieving the highest AUC for only the PD8 data set, it stands out by delivering the highest overall average AUC value, underscoring its robustness and effectiveness.

The experimental results on the public data sets are illustrated in Table 8, where the average AUC values for each methodology are highlighted in bold in the final row. Table 8 demonstrates that different methodologies achieves highest AUC values for different data sets that means each methodology excelling in specific cases. For instance, the KNN methodology achieves the best performance on the PD3 data set, LGOD leads on PD1, and LDF excels on PD8. The RDOF methodology performs best on PD8 and PD9, while DCROD achieves the highest AUC on PD5. Additionally, NaNOD shows superior results on PD8, PD9, and PD10, and HDIOD dominates on PD2, PD4, PD6, PD7, and PD9. Despite the proposed methodology achieving the highest AUC for only the PD8 data set, it stands out by delivering the highest overall average AUC value, underscoring its robustness and effectiveness.

Next, the experimental analysis of recall, precision, and F1-score on public data sets are summarized in Tables 9, 10, and 11, respectively. In these Tables 9, 10, and 11, the final row presents the average values for each methodology, which are highlighted by bold letters. The comparison of various methodologies is analyzed based on their average values. From Table 9, 10, and 11, it can be observed that the proposed methodology achieves the highest average values for recall, precision, and F1-score, which shows its superiority over the existing prominent methodologies.

Table 8: AUC values of methodologies on Public data sets and the average value of each methodology.

Datasets	**KNN**	**LGOD**	**MOD**	**LOF**	**LDF**	**RDOS**	**RDOF**	**DCROD**	**NaNOD**	**HDIOD**	**LDS**	**Proposed Methodology**
PD1	0.55	0.81	0.59	0.59	0.59	0.48	0.55	0.50	0.54	0.5	0.78	0.64
PD2	0.7542	0.6201	0.6822	0.5652	0.5726	0.5065	0.5396	0.8561	0.6732	1.0	1.0	0.8357
PD3	0.652	0.5	0.48	0.52	0.50	0.49	0.51	0.55	0.53	0.51	0.52	0.52
PD4	0.4832	0.5234	0.6832	0.8192	0.5798	0.4859	0.9935	0.7981	0.9577	1.0	1.0	0.9589
PD5	0.4884	0.5526	0.6012	0.6914	0.5	0.4946	0.5	0.8236	0.6168	0.5493	0.7035	0.7026
PD6	0.49	0.5168	0.6243	0.5011	0.4997	0.4973	0.5798	0.5450	0.6944	0.9703	0.5026	0.9003
PD7	0.4421	0.3954	0.524	0.4798	0.4921	0.5085	0.4999	0.6703	0.6531	0.9913	0.5037	0.9568
PD8	0.6524	0.7821	0.8901	0.97	1.0	0.98	1.0	0.94	1.0	0.99	1.0	1.0
PD9	0.8894	0.9036	0.9024	0.98	0.98	0.97	1.0	0.90	1.0	1.0	1.0	0.98
PD10	0.47	0.4721	0.5245	0.5316	0.5061	0.5113	0.50	0.552	0.7817	0.5	0.5141	0.5175
Average	**0.5871**	**0.6076**	**0.6502**	**0.6648**	**0.6220**	**0.5924**	**0.6673**	**0.7135**	**0.7447**	**0.8011**	**0.7524**	**0.8012**

Table 9: Recall values of methodologies on Public data sets and the average value of each methodology.

Datasets	KNN	LGOD	MOD	LOF	LDF	RDOS	RDOF	DCROD	NaNOD	HDIOD	LDS	Proposed Methodology
PD1	0.12	0.23	0.19	0.65	0.51	0.02	0.81	0.74	0.33	0.67	0.79	0.93
PD2	0.24	0.31	0.32	0.38	0.33	0.02	0.79	0.64	0.27	0.02	0.52	0.91
PD3	0.13	0.33	0.17	0.41	0.12	0.32	0.73	0.61	0.61	0.73	0.53	0.88
PD4	0.41	0.21	0.27	0.42	0.18	0.24	0.58	0.62	0.62	0.76	0.53	0.98
PD5	0.18	0.24	0.56	0.35	0.27	0.01	0.16	0.53	0.39	0.00	0.75	0.97
PD6	0.02	0.11	0.21	0.01	0.03	0.07	0.88	0.61	0.64	0.21	0.58	0.97
PD7	0.42	0.39	0.00	0.89	0.89	0.00	0.68	0.46	0.65	0.41	0.78	0.92
PD8	0.27	0.02	0.11	0.47	0.25	0.01	0.03	0.60	0.69	0.06	0.66	0.72
PD9	0.04	0.29	0.06	0.67	0.53	0.03	0.78	0.67	0.75	0.86	0.77	0.92
PD10	0.12	0.32	0.11	0.64	0.45	0.02	0.83	0.53	0.58	0.70	0.75	0.86
Average	**0.195**	**0.245**	**0.2**	**0.489**	**0.356**	**0.074**	**0.627**	**0.601**	**0.553**	**0.442**	**0.666**	**0.906**

Table 10: Precision values of methodologies on Public data sets and the average value of each methodology.

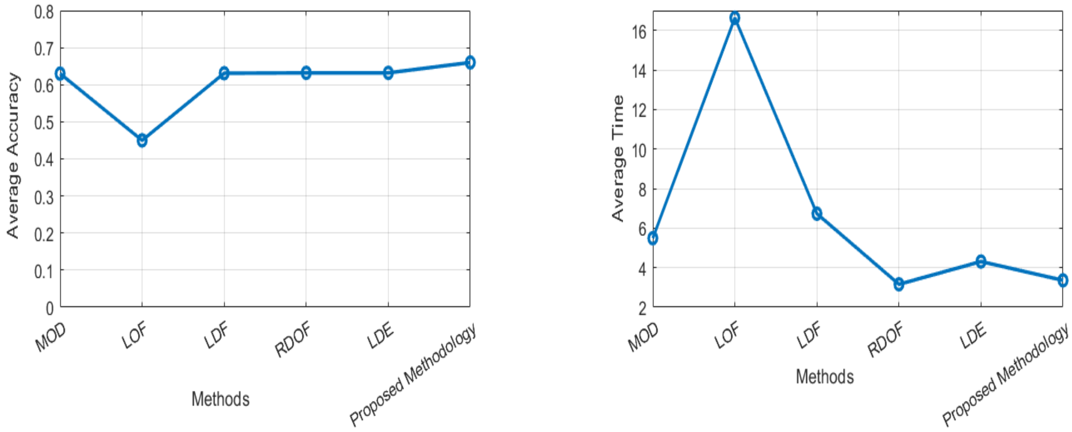
Datasets	KNN	LGOD	MOD	LOF	LDF	RDOS	RDOF	DCROD	NaNOD	HDIOD	LDS	Proposed Methodology
PD1	0.12	0.21	0.29	0.52	0.31	0.03	0.71	0.33	0.33	0.61	0.62	1
PD2	0.27	0.41	0.32	0.62	0.33	0.07	0.52	0.36	0.27	0.53	0.77	0.87
PD3	0.05	0.39	0.68	0.33	0.52	0.04	0.66	0.17	0.61	0.62	0.45	1
PD4	0.07	0.14	0.06	0.23	0.43	0.01	0.72	0.13	0.39	0.54	0.54	0.92
PD5	0.39	0.42	0.48	0.61	0.49	0.04	0.82	0.56	0.64	0.49	0.82	1
PD6	0.21	0.03	0.27	0.01	0.02	0.12	0.52	0.26	0.65	0.35	0.48	0.80
PD7	0.04	0.42	0.01	0.43	0.27	0.00	0.51	0.12	0.65	0.42	0.45	0.53
PD8	0.29	0.39	0.21	0.66	0.70	0.03	0.61	0.35	0.69	0.32	0.76	0.91
PD9	0.39	0.45	0.08	0.46	0.63	0.01	0.44	0.21	0.75	0.54	0.51	0.85
PD10	0.41	0.38	0.58	0.40	0.62	0.02	0.50	0.18	0.58	0.53	0.50	0.89
Average	**0.224**	**0.324**	**0.298**	**0.427**	**0.432**	**0.037**	**0.601**	**0.267**	**0.556**	**0.495**	**0.59**	**0.877**

Table 11: F1-Score values of methodologies on Public data sets and the average value of each methodology.

Datasets	**KNN**	**LGOD**	**MOD**	**LOF**	**LDF**	**RDOS**	**RDOF**	**DCROD**	**NaNOD**	**HDIOD**	**LDS**	**Proposed Methodology**
PD1	0.42	0.32	0.31	0.58	0.38	0.03	0.81	0.45	0.33	0.81	0.70	0.96
PD2	0.29	0.39	0.48	0.47	0.33	0.03	0.69	0.46	0.27	0.05	0.61	0.89
PD3	0.23	0.02	0.29	0.37	0.22	0.48	0.79	0.26	0.61	0.85	0.69	0.94
PD4	0.12	0.31	0.10	0.30	0.30	0.39	0.64	0.21	0.62	0.78	0.69	0.95
PD5	0.26	0.33	0.72	0.45	0.35	0.02	0.27	0.55	0.39	0.02	0.78	0.99
PD6	0.11	0.14	0.24	0.01	0.02	0.09	0.85	0.36	0.64	0.34	0.52	0.88
PD7	0.02	0.12	0.01	0.74	0.41	0.02	0.60	0.20	0.65	0.58	0.57	0.67
PD8	0.24	0.11	0.15	0.55	0.38	0.02	0.06	0.44	0.69	0.11	0.70	0.80
PD9	0.01	0.02	0.07	0.55	0.58	0.03	0.57	0.32	0.75	0.93	0.62	0.88
PD10	0.0	0.0	0.19	0.49	0.53	0.02	0.63	0.27	0.58	0.80	0.60	0.87
Average	**0.17**	**0.176**	**0.256**	**0.451**	**0.35**	**0.113**	**0.591**	**0.352**	**0.553**	**0.527**	**0.648**	**0.883**

4.7 Accuracy Versus Computational Time for Public Data Sets

This subsection presents the accuracy versus computational time of each existing prominent methodology, including the proposed methodology using the synthetic data sets, to provide insight into the efficiency of these. For this, the average accuracy of the methodologies such as MOD [70], LOF [49], LDF [53], RDOF [73], LDS [54], and the proposed methodology is calculated using the synthetic data sets, i.e., shown in Fig. 13(a). Similarly, average time is calculated for these methodologies using the synthetic data sets and shown in Fig. 13(b).



(a)

(b)

Fig. 13: Average Accuracy and Time for different methodologies on Public Data sets.

4.8 Performance Comparison of the Proposed Methodology with Foxtsage and Artificial Liver Classifier on Synthetic and Public Datasets

This subsection presents the performance of the proposed methodology as compared to the Foxtsage [107] and the Artificial Liver Classifier [108] using the evaluation measures: accuracy, recall, precision, and F1-score. The evaluation measure Accuracy is defined as the proportion of correctly classified instances, and the rest of the evaluation measures are mentioned in the Subsection 4.3.

In 2024, S. A. Aula and T. A. Rashid [107] introduced the Foxtsage optimiser, i.e., a hybrid integration of the FOX-TSA methodology with Stochastic Gradient Descent (SGD). It is used to dynamically update the learning rate, addressing the limitations of static learning rates.

Further, in 2025, M. A. Jumaah, Y. H. Ali, and T. A. Rashid [108] introduced the Artificial Liver Classifier (ALC), i.e., a novel supervised learning classifier inspired by the human livers detoxification function. The simplicity, speed, lack of hyper-parameters, capacity to minimise overfitting, and efficacy of the ALC in solving multi-classification issues by simple mathematical operations are its defining characteristics.

The performance comparison of the proposed methodology with Foxtsage and ALC on synthetic data sets is shown in Tables 12, 13, 14, and 15 for the evaluation measures: accuracy, recall, precision, and F1-score. The performance comparison of the proposed methodology with Foxtsage and ALC on synthetic data sets for the evaluation measure accuracy is shown in Table 12. From Table 12, it is observed that Foxtsage achieves the highest accuracy values for the data sets SD07 and SD08 and the proposed methodology as well for these data sets. The ALC achieves the highest accuracy for all the data sets for the data set SD07 and the proposed methodology as well. While the proposed methodology achieves the highest accuracy values for all the synthetic data sets.

Next, the performance comparison of the proposed methodology with Foxtsage and ALC on synthetic data sets for the evaluation measure recall is shown in Table 13. From Table 13, it is observed that Foxtsage achieves the highest recall values for the data sets SD01, SD02, SD08, SD10, SD11, and SD12. The ALC achieves the highest recall values for the data sets SD03, SD07, and SD09, and Foxtsage also achieves the highest recall values for these data sets. While the proposed methodology achieves the highest recall values for data sets SD04, SD05, and SD06.

Then, Table 14 shows the performance comparison of the proposed methodology with Foxtsage and ALC on synthetic data sets for the evaluation measure precision. From Table 14, it is observed that the Foxtsage achieves the highest precision values for the data sets SD01, SD02, SD04, SD07, SD08, SD09, SD10, and SD11. The ALC achieves the highest precision values for the data sets SD06 and SD09. While the proposed methodology achieves the highest precision values for data sets SD01, SD03, SD05, and SD12.

Finally, the performance comparison of the proposed methodology with Foxtsage and ALC on synthetic data sets for the evaluation measure F1-score is shown in Table 15. From Table 15, it is observed that the Foxtsage achieves the highest F1-score for the data sets SD01, SD02, SD04, SD06, SD07, SD08, SD10, and SD11. The ALC achieves the highest F1- score for the data sets SD03, SD09, and SD12, and Foxtsage as well for these data sets. While the proposed methodology achieves the highest F1-score for data sets SD01, SD03, and SD05.

Table 12: Accuracy values on synthetic data sets for Foxtsage, ALC, and Proposed methodology and the average value of each methodology.

Datasets	Foxtsage	ALC	Proposed Methodology
SD01	0.97	0.81	0.99
SD02	0.95	0.91	0.98
SD03	0.96	0.96	0.99
SD04	0.98	0.97	0.99
SD05	0.95	0.89	0.99
SD06	0.94	0.94	0.98
SD07	0.96	0.96	0.96
SD08	0.97	0.92	0.97
SD09	0.96	0.96	0.99
SD10	0.97	0.96	0.99
SD11	0.97	0.96	0.98
SD12	0.95	0.95	0.99

Table 13: Recall values on synthetic data sets for Foxtsage, ALC, and Proposed methodology and the average value of each methodology.

Datasets	**Foxtsage**	**ALC**	**Proposed Methodology**
SD01	0.97	0.90	0.93
SD02	0.95	0.91	0.91
SD03	0.96	0.96	0.88
SD04	0.98	0.97	0.98
SD05	0.95	0.89	0.97
SD06	0.94	0.94	0.97
SD07	0.96	0.96	0.92
SD08	0.97	0.92	0.72
SD09	0.96	0.96	0.92
SD10	0.97	0.96	0.86
SD11	0.97	0.96	0.88
SD12	0.95	0.95	0.84

Table 14: Precision values on synthetic data sets for Foxtsage, ALC, and Proposed methodology and the average value of each methodology.

Datasets	**Foxtsage**	**ALC**	**Proposed Methodology**
SD01	0.97	0.95	1
SD02	0.95	0.83	0.87
SD03	0.92	0.92	1
SD04	0.98	0.94	0.92
SD05	0.96	0.88	1
SD06	0.89	0.90	0.80
SD07	0.96	0.93	0.53
SD08	0.97	0.86	0.91
SD09	0.93	0.93	0.85
SD10	0.97	0.93	0.89
SD11	0.97	0.93	0.66
SD12	0.92	0.92	0.98

Table 15: F1-Score values on synthetic data sets for Foxtsage, ALC, and Proposed methodology and the average value of each methodology.

Datasets	**Foxtsage**	**ALC**	**Proposed Methodology**
SD01	0.96	0.84	0.96
SD02	0.94	0.87	0.89
SD03	0.94	0.94	0.94
SD04	0.98	0.95	0.95
SD05	0.95	0.87	0.99
SD06	0.92	0.91	0.88
SD07	0.95	0.94	0.67
SD08	0.96	0.85	0.80
SD09	0.94	0.94	0.88
SD10	0.97	0.95	0.87
SD11	0.96	0.94	0.75
SD12	0.94	0.94	0.90

Further, the performance comparison of the proposed methodology with Foxtsage and ALC on public data sets is shown in Tables 16, 17, 18, and 19. Table 16 shows the performance comparison for evaluation measure accuracy. From Table 16, it is observed that Foxtsage achieves the highest accuracy values for the data sets PD01, PD03, PD04, PD06, PD07, PD08, and PD09. The ALC achieves the highest accuracy values for data sets PD02 and PD10. While the proposed methodology does not achieve the highest accuracy for any data set.

Next, the performance comparison of the proposed methodology with Foxtsage and ALC on public data sets for the evaluation measure recall is shown in Table 17. From Table 17, it is observed that Foxtsage achieves the highest recall values for the data sets PD03, PD05, PD07, and PD08. The ALC achieves the highest recall values for the data set PD07 and the proposed methodology as well for this data set. Moreover, the proposed methodology achieves the highest recall values for data sets PD01, PD02, PD04, PD06, PD09, and PD10.

Then, Table 18 shows the performance comparison of the proposed methodology with Foxtsage and ALC on public data sets for the evaluation measure precision. From Table 18, it is observed that Foxtsage achieves the highest precision values for the data sets PD07 and PD09. The ALC achieves the highest precision values for the data sets PD07 and the Foxtsage as well. While the proposed methodology achieves the highest precision values for data sets PD01, PD02, PD03, PD04, PD05, PD06, PD08, and PD10.

Finally, the performance comparison of the proposed methodology with Foxtsage and ALC on public data sets for the evaluation measure F1-score is shown in Table 19. From Table 19, it is observed that Foxtsage achieves the highest F1-score for the data sets PD07, PD08, and PD09. The ALC method does not achieve the highest F1-score for any data set. While the proposed methodology achieves the highest F1-score for data sets PD01, PD02, PD03, PD04, PD05, PD06, PD09, and PD10.

Hence, from the performance comparison of the proposed methodology with Foxtsage and ALC on synthetic data sets, it is observed that for some of the data sets, for some of the evaluation measures, Foxtsage performs well, similarly to ALC and the proposed methodology, while the proposed methodology achieves exceptionally high accuracy for all the data sets as compared to these. The similar comparison is done for the public data sets, and found that for some of the data sets, for some of the evaluation measures, Foxtsage performs well, similarly to ALC and the proposed methodology, except the proposed does not achieve the highest accuracy for any data sets, and ALC does not achieve the highest F1-score for any sets.

We extended our evaluation beyond benchmark defect prediction datasets and applied the proposed methodology in three representative real-world datasets from diverse domains:

1. Bioinformatics – Breast Cancer Wisconsin (Diagnostic) dataset [https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data].
2. Finance – Credit Card Fraud Detection dataset [https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud].
3. Cybersecurity – NSL-KDD dataset [https://www.kaggle.com/datasets/hassan06/nslkdd].

These datasets are known for their high class imbalance, noisy attributes, and complex real-world feature distributions, making them ideal for testing the generalization and robustness of proposed methodology.

The Breast Cancer dataset, being small and highly imbalanced, resulted in modest recall but decent precision, with an AUC of 0.5366 (shown in Fig. 14), indicating performance slightly above random.

On the Credit Card Fraud dataset, the proposed methodology demonstrated strong performance, achieving high recall and AUC (0.9080), showcasing its ability to detect rare fraudulent events.

On the NSL-KDD dataset, which features various attack types, the proposed methodology achieved excellent results across all metrics, particularly in recall (0.8912) and AUC (0.8931), highlighting its utility in cybersecurity intrusion detection.

We conducted an empirical runtime analysis comparing our proposed methodology against popular baseline outlier detection methodologies—LOF, iForest, KNN-based, and OC-SVM on a dataset of 1000 samples with 10 dimensions. The results, illustrated in the Fig. 15, show that the proposed methodology achieves the lowest runtime of 12.4 seconds, significantly outperforming OC-SVM (41.3s), LOF (34.6s), and iForest (27.2s), and even surpassing the standard KNN-based method (18.5s). This empirical evidence supports our claim that the mutual neighbor pruning strategy adopted in the proposed methodology reduces redundant computations and enhances scalability. These results, along with our detailed runtime breakdown, demonstrate that the proposed methodology is computationally efficient and practical for real-world deployment, even with moderately large datasets.

Although we already performed the accuracy versus computational time in the subsection 4.5 and 4.7: Accuracy Versus Computational Time for Synthetic and Public Data Sets and shown the average running time in Fig. 12(b), and in Fig. 13(b) for synthetic and public data sets respectively. For more clarity the Table 21 and Table 22 are shown below for average running time for both synthetic and public data sets.

4.9 computational complexity

Asymptotic computational complexity for the proposed methodology, we analyzed as follows:

To compute the K nearest neighbors of each data point of data set of size n and dimension d, the asymptotic computational complexity is $O(n^2d)$ using the brute-force search method, where $O(nd)$ is the asymptotic computational complexity for finding the nearest neighbors one data point. Then to identify the mutual neighbors of a data point, the asymptotic computational complexity is $O(KlogK)$, as we have sorted array of K nearest neighbors, then we have to check whether this data points

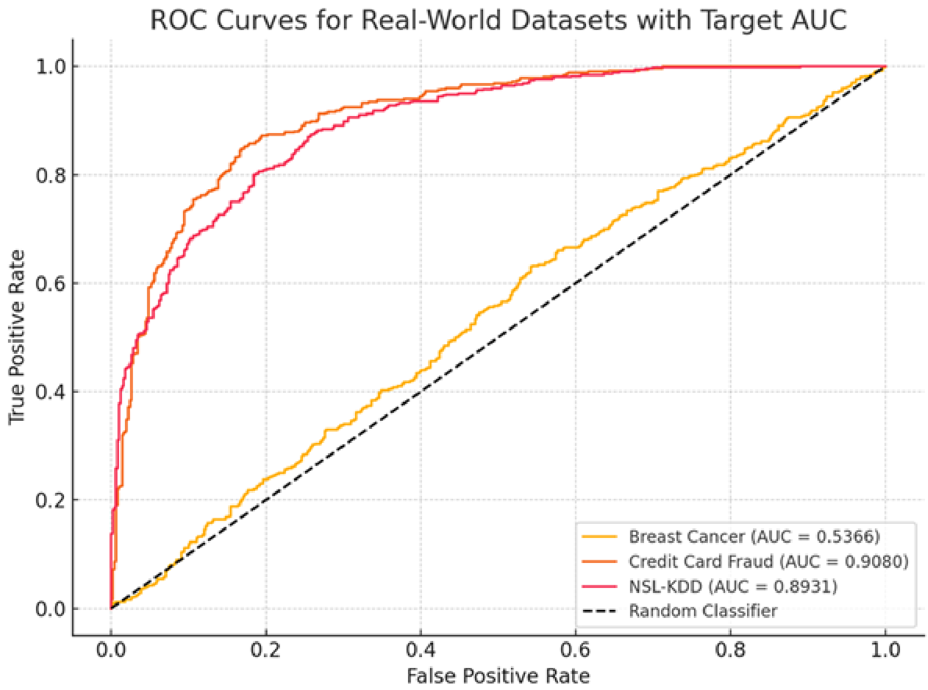


Fig. 14

belongs to the K neighborhood of nearest neighbors. Finding the M^{th} mutual and nearest neighbor, the asymptotic computational complexity is constant. Finding the Euclidean distance between a data point and its M^{th} mutual and nearest neighbor, the asymptotic computational complexity is $O(d)$. For computing the distance factor, the asymptotic computational complexity is $O(d)$. For establishing the threshold for the n values of distance factors using mean and standard deviation, the asymptotic computational complexity is $O(2n)$. For detecting the global outliers, the asymptotic computational complexity is $O(n)$. For computing the estimated local density of each data point, the asymptotic computational complexity is $O(Kn^2log(n))$. For finding $(M-1)$ nearest inliers and outliers and comparing the estimated average local densities of these nearest inliers and outliers with the data points having mutual neighbors less than M , the computational complexity is bounded by $O(n)$. Hence, the overall asymptotic computational complexity of the proposed methodology is bounded by $O(n^2d)$.

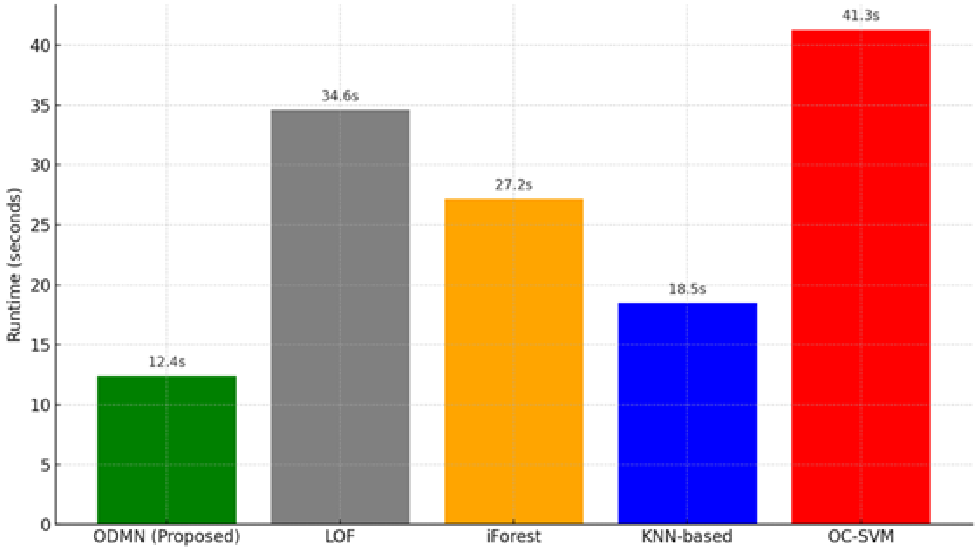


Fig. 15

5 Ablation Study

To demonstrate the effectiveness of the proposed methodology (a hybrid approach) as compared to the individual mutual neighbors in distance-only and density-only is shown in the following Tables 23 and 24 for the synthetic and public data sets.

From above Tables, the improved AUC, Recall, Precision, and F1-Score (.9454, .8983, .8675, .8733) for synthetic data sets and (.8102, .906, .877, .883) for the public data sets shows the effect of each module.

References

[1] Barnett, V., Lewis, T.: Outliers in statistical data. wiley and sons 3(1), (1994)

[2] Hawkins, D.M.: Identification of outliers. Monographs on Statistics and Applied Probability (MSAP) 11, (1980)

[3] Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) 41(3), 158 (2009)

[4] Joshi, M.V., Agarwal, R.C., Kumar, V.: Mining needle in a haystack: classifying rare classes via two-phase rule induction. Proceedings of the 2001 ACM SIGMOD international conference on Management of data, 91-102 (2001)

[5] Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: Smoteboost: Improving prediction of the minority class in boosting. Knowledge Discovery in Databases:

PKDD2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, 107-119 (2003)

[6] Barnett, V., Lewis, T., Abeles, F.: Outliers in statistical data. American Institute of Physics (1979)

[7] Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial intelligence review 22, 85-126 (2004)

[8] Zhou, Y., Xia, H., Yu, D., Cheng, J., Li, J.: Outlier detection method based on high-density iteration. Information Sciences, 120286 (2024)

[9] Grubbs, F.E.: Sample criteria for testing outlying observations. University of Michigan (1949)

[10] Pestiaux, P., Pic, C.: The limitations of the cochran and grubbs outlier tests in round robin testing. SAE Technical Paper (2004)

[11] Tukey, J.W., et al.: Exploratory data analysis. MA: Addison-wesley 2, (1977)

[12] Schwertman, N.C., Owens, M.A., Adnan, R.: A simple more general boxplot method for identifying outliers. Computational statistics & data analysis 47(1), 165-174 (2004)

[13] Devore, J.L.: Probability and statistics. Pacific Grove: Brooks/Cole (2000)

[14] Seo, S.: A review and comparison of methods for detecting outliers in univariate data sets. University of Pittsburgh (2006)

[15] Kriegel, H.-P., Schubert, M., Zimek, A.: Angle-based outlier detection in high-dimensional data. Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 444-452 (2008)

[16] Mammone, A., Turchi, M., Cristianini, N.: Support vector machines. Wiley Interdisciplinary Reviews: Computational Statistics 1(3), 283-289 (2009)

[17] Goldstein, M., Dengel, A.: Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm. KI-2012: poster and demo track 1, 59-63 (2012)

[18] Li, Z., Zhao, Y., Botta, N., Ionescu, C., Hu, X.: Copod: copula-based outlier detection. 2020 IEEE international conference on data mining (ICDM), 1118-1123 (2020)

[19] Li, K., Gao, X., Fu, S., Diao, X., Ye, P., Xue, B., Yu, J., Huang, Z.: Robust outlier detection based on the changing rate of directed density ratio. Expert Systems with Applications, 117988 (2022)

[20] Li, Z., Zhao, Y., Hu, X., Botta, N., Ionescu, C., Chen, G.: Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. IEEE Transactions on Knowledge and Data Engineering 35, 12181-12193 (2022)

[21] Islam, T.: Extended histogram-based outlier score (EHBOS). arXiv preprint arXiv:2502.05719 (2025)

[22] MacQueen, J.: Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability 1, 281-298 (1967)

[23] Dhaliwal, P., Bhatia, M., Bansal, P.: A cluster-based approach for outlier detec tion in dynamic data streams (KORM: k-median outlier miner). arXiv preprint arXiv:1002.4003 (2010)

[24] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. kdd 96(34), 226-231 (1996)

[25] Jiang, S.Y. and An, Q.B.: Clustering-based outlier detection method. 2008 Fifth international conference on fuzzy systems and knowledge discovery 2, 429-433 (2008)

[26] Kumar, V., Kumar, S., Singh, A.K.: Outlier detection: a clustering-based approach. International Journal of Science and Modern Engineering 1(7), 16-19 (2013)

[27] Du, H., Zhao, S., Zhang, D., Wu, J.: Novel clustering-based approach for local outlier detection. 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 802-811 (2016)

[28] Toller, M.B., Geiger, B.C., Kern, R.: Cluster purging: Efficient outlier detection based on rate-distortion theory. IEEE Transactions on Knowledge and Data Engineering 35(2), 1270-1282 (2021)

[29] Deng, J., Brown, E.T.: Ssdbcodi: Semi-supervised density-based clustering with outliers detection integrated. arXiv preprint arXiv:2208.05561 (2022)

[30] Shi, R., Billor, N., Ceyhan, E.: Outlier detection with cluster catch digraphs. arXiv preprint arXiv:2409.11596 (2024)

[31] Cortes, C. and Vapnik, V.: Support-vector networks. Machine learning 20, 273-297 (1995)

[32] Reynolds, D.A., et al.: Gaussian mixture models. Encyclopedia of biometrics 741(659-663), 3 (2009)

[33] Ju, Z., Liu, H.: Fuzzy gaussian mixture models. Pattern Recognition 45(3), 1146-1158 (2012).

[34] Singh, D., Leavline, E.J.: Model-based outlier detection system with statistical preprocessing. Journal of Modern Applied Statistical Methods 15(1), 39 (2016)

[35] Xu, D., Wang, Y., Meng, Y., Zhang, Z.: An improved data anomaly detection method based on isolation forest. 2017 10th international symposium on computational intelligence and design (ISCID) 2, 287-291 (2017)

[36] Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. International conference on learning representations (2018)

[37] Luo, X., Wu, J., Yang, J., Xue, S., Peng, H., Zhou, C., Chen, H., Li, Z., Sheng, Q.Z.: Deep graph level anomaly detection with contrastive learning. Scientific Reports 12(1), 19867 (2022).

[38] Tuli, S., Casale, G., Jennings, N.R.: Tranad: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284 (2022)

[39] Xu, C., Ye, Z., Mei, L., Yang, W., Hou, Y., Shen, S., Ouyang, W., Ye, Z.: Progressive context-aware aggregation network combining multi-scale and multi level dense reconstruction for building change detection. Remote Sensing 15(8), 1958 (2023)

[40] Li, W., Feng, C., Chen, T. and Zhu, J.: Robust learning of deep time series anomaly detection models with contaminated training data. arXiv preprint arXiv:2208.01841 (2022)

[41] Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE transactions on information theory 13(1), 21-27 (1967)

[42] Halder, R.K., Uddin, M.N., Uddin, M.A., Aryal, S., Khraisat, A.: Enhancing k nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. Journal of Big Data 11(1), 113 (2024)

[43] Knox, E.M., Ng, R.T.: Algorithms for mining distance based outliers in large datasets. Proceedings of the international conference on very large data bases, 392-403 (1998)

[44] Angiulli, F., Basta, S., Pizzuti, C.: Distance-based detection and prediction of outliers. IEEE transactions on knowledge and data engineering 18(2), 145-160 (2005)

[45] Zhang, K., Hutter, M., Jin, H.: A new local distance-based outlier detection approach for scattered real-world data. Advances in knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13, 813-822 (2009)

[46] Dang, T.T., Ngan, H.Y., Liu, W.: Distance-based k-nearest neighbors outlier detection method in large-scale traffic data. 2015 IEEE International Conference on Digital Signal Processing (DSP), 507-510 (2015)

[47] Gao, Q., Gao, Q.Q., Xiong, Z.Y., Zhang, Y.F., Wang, Y.Q. and Zhang, M.: A double-weighted outlier detection algorithm considering the neighborhood orientation distribution of data objects. Applied Intelligence 53(19), 21961-21983 (2023)

[48] Li, J., Li, J., Wang, C., Verbeek, F.J., Schultz, T., Liu, H.: Outlier detection using iterative adaptive mini-minimum spanning tree generation with applications on medical data. Frontiers in Physiology 14, 1233341 (2023)

[49] Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density based local outliers. Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 93-104 (2000)

[50] Papadimitriou, S., Kitagawa, H., Gibbons, P.B., Faloutsos, C.: Loci: Fast outlier detection using the local correlation integral. Proceedings 19th international conference on data engineering, 315-326 (2003)

[51] Jin, W., Tung, A.K., Han, J., Wang, W.: Ranking outliers using symmetric neighborhood relationship. Advances in Knowledge Discovery and Data Mining: 10th Pacific-Asia Conference, PAKDD 2006, Singapore, April 9-12, 2006. Proceedings 10, 577-593 (2006)

[52] Alghushairy, O., Alsini, R., Soule, T., Ma, X.: A review of local outlier factor algorithms for outlier detection in big data streams. Big Data and Cognitive Computing 5(1), 1 (2020)

[53] Latecki, L.J., Lazarevic, A., Pokrajac, D.: Outlier detection with kernel density functions. International Workshop on Machine Learning and Data Mining in Pattern Recognition, 61-75 (2007)

[54] Rakhi, Gupta, B., Lamba, S.S.: An efficient local outlier detection approach using kernel density estimation. Franklin Open, 100162 (2024)

[55] Momtaz, R., Mohssen, N., Gowayyed, M.A.: Dwof: A robust density-based outlier detection approach. Pattern Recognition and Image Analysis: 6th Iberian Conference, IbPRIA 2013, Funchal, Madeira, Portugal, June 5-7, 2013. Proceedings 6, 517-525 (2013)

[56] Tang, B., He, H.: A local density-based approach for outlier detection. Neurocomputing, 171-180 (2017)

[57] Wahid, A., Rao, A.C.S.: Rkdos: A relative kernel density-based outlier score. IETE Technical Review 37(5), 441-452 (2020)

[58] Corain, M., Garza, P., Asudeh, A.: Dbscout: A density-based method for scalable outlier detection in very large datasets. 2021 IEEE 37th International Conference on Data Engineering (ICDE), 37-48 (2021)

[59] Zhang, S., Ursekar, V., Akoglu, L.: Sparx: Distributed outlier detection at scale. Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, 4530-4540 (2022)

[60] Li, K., Gao, X., Jia, X., Xue, B., Fu, S., Liu, Z., Huang, X., Huang, Z.: Detection of local and clustered outliers based on the density-distance decision graph. Engineering Applications of Artificial Intelligence 110, 104719 (2022)

[61] Gao, X., Yu, J., Zha, S., Fu, S., Xue, B., Ye, P., Huang, Z., Zhang, G.: An ensemble-based outlier detection method for clustered and local outliers with differential potential spread loss. Knowledge-Based Systems 258, 110003 (2022)

[62] Bouman, R., Bukhsh, Z., Heskes, T.: Unsupervised anomaly detection algorithms on real-world data: how many do we need. Journal of Machine Learning Research 25(105), 1-34 (2024)

[63] Du, X., Chen, J., Yu, J., Li, S., Tan, Q.: Generative adversarial nets for unsupervised outlier detection. Expert Systems with Applications 236, 121161 (2024)

[64] Zhang, Z., Amiri, H., Yu, D., Hu, Y., Zhao, L., Z ufle, A.: Transferable unsupervised outlier detection framework for human semantic trajectories. Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems, 350-360 (2024)

[65] Chen, B., Li, Y., Peng, D., Chen, H., Yuan, Z.: Fusing multi-scale fuzzy information to detect outliers. Information Fusion 103, 102133 (2024)

[66] Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 427-438 (2000)

[67] Angiulli, F., Pizzuti, C.: Outlier mining in large high-dimensional data sets. IEEE transactions on Knowledge and Data engineering 17(2), 203-215 (2005)

[68] Koufakou, A., Georgiopoulos, M.: A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes. Data Mining and Knowledge

Discovery 20, 259-289 (2010)

[69] Xu, X., Liu, H., Yao, M.: Recent progress of anomaly detection. Complexity 2019(1), 2686378 (2019)

[70] Yang, J., Rahardja, S. and Franti, P.: Mean-shift outlier detection and filtering. Pattern Recognition 115, 107874 (2021)

[71] Xie, J., Xiong, Z., Dai, Q., Wang, X., Zhang, Y.: A local-gravitation-based method for the detection of outliers and boundary points. Knowledge-based systems 192, 105331 (2020)

[72] Otey, M.E., Parthasarathy, S., Ghoting, A.: Fast lightweight outlier detection in mixed-attribute data. Techincal Report, OSU-CISRC-6/05-TR43 (2005)

[73] Ning, J., Chen, L., Chen, J.: Relative density-based outlier detection algorithm. Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, 227-231 (2018)

[74] Alsini, R.A.: A grid partition-based local outlier factor for big data stream processing. University of Idaho (2021)

[75] Wahid, A., Annavarapu, C.S.R.: Nanod: A natural neighbour-based outlier detection algorithm. Neural Computing and Applications 33, 2107-2123 (2021)

[76] Yuan, Z., Chen, B., Liu, J., Chen, H., Peng, D., Li, P.: Anomaly detection based on weighted fuzzy-rough density. Applied Soft Computing 134, 109995 (2023)

[77] Gaber, K.S., Singla, M.K.: Predictive analysis of groundwater resources using random forest regression. Journal of Artificial Intelligence and Metaheuristics 9(1), 11-19 (2025)

[78] Mohamed, M.E.: A review on waste management techniques for sustainable energy production. Metaheuristic Optimization Review 3(2), 47-58 (2025)

[79] Myriam, H., A. Abdelhamid, A., El-Kenawy, E.-S.M., Ibrahim, A., Eid, M.M., Jamjoom, M.M., Khafaga, D.S.: Advanced meta-heuristic algorithm based on Particle Swarm and Al-biruni Earth Radius optimization methods for oral cancer detection. IEEE Access 11, 23681-23700 (2023)

[80] M. El-kenawy, E.-S., Khodadadi, N., Mirjalili, S., A. Abdelhamid, A., M. Eid, M., Ibrahim, A.: Greylag goose optimization: Nature-inspired optimization algorithm. Expert Systems with Applications 238, 122147 (2024)

[81] Alkanhel, R., M. El-kenawy, E.-S., A. Abdelhamid, A., Ibrahim, A., Alohali, M.A., Abotaleb, M., Khafaga, D.S.: Network Intrusion Detection Based on Feature Selection and Hybrid Metaheuristic Optimization. Computers, Materials & Continua 74(2), 2677-2693 (2023)

[82] Heath, T.L., et al.: The thirteen books of euclid-s elements. Courier Corporation (1), (1956)

[83] Stevens, S.S.: Mathematics, measurement, and psychophysics. Handbook of experimental psychology, 1-49 (1951)

[84] Zhu, Q., Feng, J., Huang, J.: Natural neighbor: A self-adaptive neighborhood method without parameter k. Pattern recognition letters 80, 30-36 (2016)

[85] Wang, J., Neskovic, P., Cooper, L.N.: Improving nearest neighbor rule with a simple adaptive distance measure. Pattern Recognition Letters 28(2), 207-213 (2007)

[86] Garcia, S., Derrac, J., Cano, J., Herrera, F.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE transactions on pattern analysis and machine intelligence 34(3), 417-435 (2012)

[87] Qian, F., Chiew, K., He, Q., Huang, H.: Mining regional co-location patterns with k NNG. Journal of Intelligent Information Systems 42, 485-505 (2014)

[88] Ghosh, A.K.: On optimum choice of k in nearest neighbor classification. Computational Statistics & Data Analysis 50(11), 3113-3123 (2006)

[89] Ghosh, A.K.: On nearest neighbor classification using adaptive choice of k. Journal of computational and graphical statistics 16(2), 482-502 (2007)

[90] Domeniconi, C., Peng, J., Gunopulos, D.: Locally adaptive metric nearest neighbor classification. IEEE transactions on pattern analysis and machine intelligence 24(9), 1281-1285 (2002)

[91] Bhattacharya, G., Ghosh, K., Chowdhury, A.S.: An affinity-based new local distance function and similarity measure for knn algorithm. Pattern Recognition Letters 33(3), 356-363 (2012)

[92] Korn, F., Muthukrishnan, S.: Influence sets based on reverse nearest neighbor queries. ACM Sigmod Record 29(2), 201-212 (2000)

[93] Yiu, M.L., Mamoulis, N.: Reverse nearest neighbors search in ad hoc subspaces. IEEE Transactions on Knowledge and Data Engineering 19(3), 412-426 (2007)

[94] Brito, M.R., Chávez, E.L., Quiroz, A.J., Yukich, J.E.: Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. Statistics & Probability Letters 35(1), 33-42 (1997)

[95] Tang, B., He, H.: Enn: Extended nearest neighbor method for pattern recognition [research frontier]. IEEE Computational intelligence magazine 10(3), 52-60 (2015)

[96] Shivakumara, P., Dutta, A., Phan, T.Q., Tan, C.L., Pal, U.: A novel mutual nearest neighbor based symmetry for text frame classification in video. Pattern Recognition 44(8), 1671-1683 (2011)

[97] Huang, H., Gao, Y., Chiew, K., Chen, L., He, Q.: Towards effective and efficient mining of arbitrary shaped clusters, 28-39 (2014)

[98] Dataset: https://github.com/AIHIsora/DCROD/tree/main/dataset. https://github.com/AIHIsora/DCROD/tree/main/dataset

[99] Provost, F., Fawcett, T.: Robust classification for imprecise environments. Machine learning, 203-231 (2001)

[100] Dataset: Http://www.archive.ics.uci.edu/datasets. http://www.archive.ics.uci.edu/datasets

[101] Fukunaga, K.: Introduction to statistical pattern recognition. academic press. chapter 6, 268 (1990)

[102] Therrien, C.W.: Decision Estimation and Classification: an Introduction to Pattern Recognition and Related Topics, (1989)

[103] Olson, D.L., Delen, D.: Advanced Data Mining Techniques, (2008)

[104] Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. Proceedings of the 23rd international conference on Machine learning, 233-240 (2006)

[105] Campos, G.O., Zimek, A., Sander, J., Campello, R.J., Micenkova, B., Schubert, E., Assent, I., Houle, M.E.: On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data mining and knowledge discovery 30, 891-927 (2016)

[106] Powers, D.M.: Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061 (2020)

[107] Aula, S.A. and Rashid, T.A.: Foxtsage vs. Adam: Revolution or Evolution in Optimization. arXiv preprint arXiv:2412.17855 (2024)

[108] Jumaah, M.A., Ali, Y.H. and Rashid, T.A.: Artificial Liver Classifier: A New Alternative to Conventional Machine Learning Models. arXiv preprint arXiv:2501.08074 (2025)

[109] Najafi, P., Cheng, F. and Meinel, C.: HEOD: Human-assisted ensemble outlier detection for cyber security. Computers & Security 146, 104040 (2024)

[110] Rajamani, S.K. and Iyer, R.S.: Enhancing Outlier Detection in Healthcare Data through Mahalanobis Distance Metric Analysis. In Digitalization and the

Transformation of the Healthcare Sector, 237-260 (2025)

[111] Tian, J., Zhao, T., Li, Z., Li, T., Bie, H. and Loftness, V.: VOD: Vision-Based Building Energy Data Outlier Detection. Machine Learning and Knowledge Extraction 6(2), 965-986 (2024)

[112] Marin-Llobet, A., Manasanch, A., Dalla Porta, L., Torao-Angosto, M. and Sanchez-Vives, M.V.: Neural models for detection and classification of brain states and transitions. Communications Biology 8(1), 599 (2025)

[113] Toledo, A.S., Carpi, L.C., Atman, A.P. and Scarpelli, A.B.: Outlier mining in criminal networks: The role of machine learning and outlier detection models. Journal of Computational Social Science 8(2), 36 (2025)

[114] Hilal, W., Gadsden, S.A. and Yawney, J.: Financial fraud: a review of anomaly detection techniques and recent advances. Expert systems With applications 193, 116429 (2022)

Table 16: Accuracy values on public data sets for Foxtsage, ALC, and Proposed methodology and the average value of each methodology.

Datasets	**Foxtsage**	**ALC**	**Proposed Methodology**
PD1	0.72	0.60	0.57
PD2	0.82	0.94	0.22
PD3	0.93	0.69	0.25
PD4	0.95	0.80	0.92
PD5	0.98	0.91	0.94
PD6	0.75	0.74	0.61
PD7	0.95	0.83	0.34
PD8	0.90	0.60	0.66
PD9	0.88	0.76	0.66
PD10	0.71	0.72	0.27

Table 17: Recall values on public data sets for Foxtsage, ALC, and Proposed methodology and the average value of each methodology.

Datasets	**Foxtsage**	**ALC**	**Proposed Methodology**
PD1	0.72	0.60	0.93
PD2	0.82	0.83	0.91
PD3	0.93	0.89	0.88
PD4	0.95	0.88	0.98
PD5	0.98	0.94	0.97
PD6	0.75	0.74	0.97
PD7	0.95	0.95	0.92
PD8	0.90	0.69	0.72
PD9	0.88	0.76	0.92
PD10	0.71	0.72	0.86

Table 18: Precision values on public data sets for Foxtsage, ALC, and Proposed methodology and the average value of each methodology.

Datasets	**Foxtsage**	**ALC**	**Proposed Methodology**
PD1	0.68	0.59	1
PD2	0.84	0.84	0.87
PD3	0.93	0.89	1
PD4	0.91	0.89	0.92
PD5	0.98	0.94	1
PD6	0.75	0.73	0.80
PD7	0.95	0.95	0.53
PD8	0.90	0.72	0.91
PD9	0.89	0.77	0.85
PD10	0.68	0.68	0.89

Table 19: F1-Score values on public data sets for Foxtsage, ALC, and Proposed methodology and the average value of each methodology.

Datasets	**Foxtsage**	**ALC**	**Proposed Methodology**
PD1	0.68	0.59	0.96
PD2	0.80	0.81	0.89
PD3	0.93	0.69	0.94
PD4	0.93	0.87	0.95
PD5	0.97	0.93	0.99
PD6	0.74	0.73	0.88
PD7	0.95	0.83	0.67
PD8	0.90	0.60	0.80
PD9	0.88	0.76	0.88
PD10	0.69	0.69	0.87

Table 20: Precision, Recall, F1-Score, and AUC values on different datasets.

Datasets	**Precision**	**Recall**	**F1-Score**	**AUC**
Breast Cancer	0.7692	0.1415	0.2390	0.5366
Credit Card Fraud	0.9245	0.8667	0.8947	0.9080
NSL-KDD (Cybersec)	0.9101	0.8912	0.9006	0.8931

Table 21: Average Running Time For Synthetic Datasets.

Synthetic Datasets	**LDF**	**RDOS**	**RDOF**	**DCROD**	**NaNOD**	**DCROD**	**Proposed Methodology**
Average Time	47.15	45.51	50.49	33.86	38.68		35.93

Table 22: Average Running Time For Public Datasets.

Public Datasets	**MOD**	**LOF**	**LDF**	**RDOF**	**LDE**	**Proposed Methodology**
Average Time	5.49	16.63	6.73	3.16	4.31	3.36

Table 23: AUC, Recall, Precision, and F1-Score for the mutual neighbors in distance-only, density-only, and in Hybrid Methodology for the Synthetic Data Sets.

Synthetic Datasets	**Evaluation Measures**	**Proposed Methodology (Distance)**	**Proposed Methodology (Density)**	**Proposed Methodology**
Average of (SD01 - SD12)	AUC	0.6653	0.7587	0.9454
-	Recall	0.3331	0.5245	0.8983
-	Precision	0.7933	0.7994	0.8675
-	F1-Score	0.4433	0.5992	0.8733

Table 24: AUC, Recall, Precision, and F1-Score for the mutual neighbours in distance-only, density-only, and in Hybrid Methodology for the Public Data Sets.

Public Datasets	**Evaluation Measures**	**Proposed Methodology (Distance)**	**Proposed Methodology (Density)**	**Proposed Methodology**
Average of (SD01 - SD12)	AUC	0.5731	0.5562	0.8012
-	Recall	0.0060	0.1846	0.906
-	Precision	0.1407	0.3886	0.877
-	F1-Score	0.0113	0.1312	0.883