An Outlier Detection Algorithm based on KNN-kernel Density Estimation

Abdul Wahid
Department of Computer Science and Engineering
Indian Institute of Technology (ISM) Dhanbad
Jharkhand, India
abdul.cspg14@nitp.ac.in

Annavarapu Chandra Sekhara Rao
Department of Computer Science and Engineering
Indian Institute of Technology (ISM) Dhanbad
Jharkhand, India
acsrao@iitism.ac.in

***Abstract*—The importance of outlier detection is growing significantly in a various fields, such as military surveillance,tax fraud detection, telecommunications, terrorist activities, medical and commercial sectors. Focusing on this has resulted in the growth of several outlier detection algorithms, mostly based on distance or density strategies. But for each approach, there are inherent weaknesses. The distance-based techniques have a local density issue, while the density-based method has a low-density pattern issue. In this article, we present an unsupervised density-based outlier detection algorithm to address these shortcomings. In the proposed approach, each object is assigned a local outlying degree, which indicates how much one point in its locality deviates from the other. The local outlying degree focuses explicitly on the concept of local density, which is defined as a relative measure of the local density of the object to the local density of its neighbour. The proposed approach uses a measure of k nearest neighbour kernel density (NKD) to estimate the density. Besides, our proposed algorithm used three different categories of nearest neighbours, k nearest neighbour (kNN), reverse nearest neighbour (RNN), and shared nearest neighbour (SNN) to make our systems more flexible in modeling different local data patterns. Formal analysis and extensive experiments on artificial and UCI machine learning repository datasets show that this technique can achieve better outlier detection performance.**

***Index Terms*—local outlier detection, density-based method, unsupervised outlier detection, nearest neighbors, kernel density estimation.**

I. Introduction

The outlier is one that appears to differ significantly from the other study participants. Hawkins [1] proposes a well-known definition of the outlier as "An outlier is an observation that deviates so much from other observations as to arouse suspicion that a different mechanism generated it". Data quality and analytical results from data mining are significantly affected by the existence of outliers in the dataset. The importance of outlier identification is due to the reality that outliers can suggest a new system pattern that generates data or detects illegitimate events in the dataset, and can also alter information in useful or critical ways. Outlier detection has received considerable attention in various real-world applications, including industrial wireless sensor network [2], fraud detection in health insurance [3], fraud detection in automobile insurance [4], intrusion detection [5], financial applications [6], manufacturing process [7] and so on, compared to other problems of knowledge discovery.

Outlier detection is a major data mining research issue designed to detect a unique or unusual data object that differs significantly from other data points. Outlier detection traditionally refers to pattern detection in a dataset that is not consistent with the established behavior [8]. Several outlier detection methods have been suggested in recent years, depending on the supervised and unsupervised method of learning. A supervised scenario involves information on normal or abnormal objects in a dataset, while there is no need for information on the distribution of classes in an unsupervised learning method. Most recent algorithms for outlier detection include an unsupervised scenario [9].

In this article, we present an unsupervised local outlier detection algorithm based on the kNN kernel density estimation [10]. A local outlying degree is assigned to each object in the proposed algorithm. In particular, the local outlying degree focuses on the notion of the local density that its neighbours give to the locality. In addition, our algorithm creates an extended neighbourhood in a new way by combining three neighbours: k Nearest Neighbours (kNN), Reverse Nearest Neighbours (RNN), and Shared Nearest Neighbours (SNN) of an object to model various local data patterns in our system. The proposed algorithm calculates the local outlying degree as the proportion of the average local neighbour density with the test point density. An object with a higher density compared to its neighbours will most likely be surrounded by dense regions, indicating that it would not be an outlier, and the lower density data points in comparison to its neighbours are a promising candidate for outliers. In short, our contributions are as follows:

- An unsupervised local outlier detection algorithm is proposed.
- A kNN kernel density (NKD) metric is used to estimate local density.
- To model various local data patterns, an object's kNN, RNN, and SNN are considered.
- Extensive experiments on artificial and real datasets and, in comparison with four existing algorithms, show the performance of our proposed algorithm.

The rest of the paper is arranged in this way. In Section II, we briefly discuss some of the existing algorithms for outlier

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

identification. Section III offers a new approach to outlier detection and explains in detail our new outlying degree (OD) measure. Section IV provides experimental results and analysis on different artificial and real datasets demonstrating the performance of the proposed approach. Finally, we conclude the paper with future work in Section V.

II. Related Work

This section gives a brief overview of the different outlier detection algorithms based on the unsupervised learning method. Most unsupervised outlier detection techniques are used to assess the outlying degree for each item in a dataset. All data points are then sorted by calculated scores, and those with a high outlying degree declare the outliers. The density-based method detects the outlier if the local density differs from that of the neighbourhood, where the density around the test point and its neighbours are assumed to be the same. For the calculation of local density of the test point, various density estimation schemes have been implemented. The most popular techniques for local outlier detection are: LOF [11], COF [12], INFLO [13], and KDEOS [14]. In Section IV, all these techniques are compared with the proposed one. Before briefly explaining these methods, we need to define the following terms.

Let X be a dataset, and $d_k(x)$ represents an Euclidean distance between point x and its k^{th} neighbour. We can use other distance metrics, such as Mahalanobis distance, Manhattan distance, etc. The measurement of distance usually depends on the variable types. Let $kNN(x)$ represents a set of k nearest neighbours of point x, defined as:

$$kNN(x) = y \in X - x : d(x,y) \leq d_k(x) \tag{1}$$

The local reachability density (lrd) of point x can be defined as:

$$lrd(x) = \frac{1}{\sum_{y \in kNN(x)} \frac{Rd_k(x,y)}{|kNN(x)|}} \tag{2}$$

where $Rd_k(x,y)$ is the reachability distance, defined as:

$$Rd_k(x,y) = max\{d_k(y), d(x,y)\} \tag{3}$$

Therefore, the final LOF score can be computed as:

$$LOF_k(x) = \frac{1}{|kNN(x)| \sum_{y \in kNN(x)} \frac{lrd_k(y)}{lrd_k(x)}} \tag{4}$$

Once the LOF score for each point $x \in X$ has been calculated, sort it by their LOF score in decreasing order. It is clear that the higher the reachability density of the nearest neighbours and the lower the reachable density of the point, the higher the LOF score of x, and the corresponding points are marked as outliers. It has been shown in [12] that one 'score, which has a higher LOF score rather than a threshold value, is more accurate to consider as outliers. Later, a number of LOF algorithm variants have been proposed.

Tang et al. [12] have developed a new strategy called COF for the underlying data patterns. The Set-based Nearest (SBN) route was chosen in the COF [12] to obtain some of the nearest neighbours. It was also used to calculate the relative density over the average chain distance of the test point. Similar to $LOF_k(x)$, the higher value of $COF_k(x)$ indicates that x is an outlier. LOF [11] and COF [12]-based approaches identify the outlier with the relative density distribution.

Jin et al. [13] projected a new density-based outlier detection algorithm called INFLO. In this strategy, the relative density is calculated by considering the influence space $IS_k(x)$, which is a combination of the kNNs and RNNs of the object. The value of $INFLO_k(x)$ is defined as:

$$INFLO_k(x) = \frac{\overline{den}(IS_k(x))}{den(x)} \tag{5}$$

where $\overline{den}(IS_k(x))=\frac{\sum_{i \in IS_k(x)} den(i)}{|IS_k(x)|}$, and $den(x) = \frac{1}{d_k(x)}$.

Several outlier detection approaches have been discussed in recent years using kernel density estimation (KDE) [14]–[18]. One of the KDE based approach is KDEOS [14], which includes KDE for outlier detection in the LOF framework. KDEOS standardized the KDE densities as a z-score compared to the KDE densities of the kNN at various neighbourhood size (k_{min} ... k_{max}), compared to neighbouring densities. To compare this with other outlier detection techniques, we set the parameter $k = k_{min} = k_{max}$.

$$KDEOS(x) =_{k_{min}..k_{max}}^{mean} \text{z-score}(KDE(x), \{KDE(y)_{y \in kNN(X)}\}) \tag{6}$$

In this section, we have discussed several popular density-based outlier detection techniques, where they have the problem of low density patterns. Therefore, compared to the methods as described above, we present an unsupervised method of outlier detection to solve this problem by proposing a new algorithm for local outlier detection based on k-NN kernel density estimation.

III. Methodology

The proposed method initially carried out a density estimation to measure the local outlier-ness score of an object. While several density estimation techniques were suggested in recent years. The most common measure of density is the *cut-off* density, which is defined by the the number of items in a $r-$ball centred on a particular object. However, the r parameter is highly sensitive. Due to a small variation of r, density estimation may vary drastically.

Another traditional measure for the density estimation is the kernel density estimator (KDE), which is defined as:

$$\rho(x) = \frac{1}{n} \sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{7}$$

where $K(\cdot)$ is a kernel function of width h that satisfies the following conditions:

$$\int K(x)dx = 1, \int xK(x)dx = 0, and \int x^2K(x)dx > 0. \tag{8}$$

KDE is continuous and less sensitive to parameter selection. However, this tends to give a bias in estimating data points in small clusters.

The density estimation we are proposing is based on k nearest neighbour kernel density (NKD)[10] and only takes into account neighboring data points to estimate the density of point x rather than the complete dataset. There are two reasons: firstly, the estimated density with whole dataset will lead to loss of local density and may detect local outliers with less accuracy. Secondly, it gives a high computational cost (i.e. $O(n^2)$) when considering the entire dataset for the calculation of the outlying degree. where n denotes the number of objects in a dataset.

We use kNN, RNN, and SNN of an object to assess more effectively the density distribution in an object's neighborhood. The RNN of the object x are those objects which consider x as one of their k nearest neighbors, i.e., y to be one of the $x's$ reverse nearest neighbors if $NN_r(y) = x$ for all $r \leq k$. The latest research has shown that RNNs can offer useful information on the local data distribution to identify outliers [13]. The SNN of the object x are those objects who share one or more nearest neighbors with x, in other words, y is one shared nearest neighbor of x if $NN_r(y) = NN_s(x)$ for any $r,s \leq k$. For an object, the nearest neighbors in kNN(x) should always be k, whereas RNN(x) and SNN(x) may have zero, one or more data points.

Given kNN(x), RNN(x), and SNN(x), we create an extended neighborhood space for an object x by merging in a new way, represented as:

$$S(x) = kNN(x) \cup RNN(x) \cup SNN(x). \tag{9}$$

Thus, the new local density measure is defined as:

$$\rho(x) = \alpha \sum_{y \in S(x)} exp(\frac{-\delta(x,y)}{\Delta}) \tag{10}$$

where

$$\delta(x,y) = \begin{cases} min_{y \in S(x)} d(x,y), & \text{if } \exists y s.t. \rho(x) < \rho(y) \\ max_{y \in S(x)} d(x,y), & \text{otherwise} \end{cases} \tag{11}$$

Δ is the average distance between point x and its k^{th} nearest neighbors, defined as:

$$\Delta = \frac{1}{n} \sum_{x \in D} d(x, NN_k(x)) \tag{12}$$

and α is a controlling parameter ranging from $(0,1)$.

A. Proposed Outlier Detection Algorithm

This section presents an algorithm for calculating the outlying degree for an object in its locality. Following the estimation of density at each object, we propose a new outlining approach to evaluating to what extent an object's density differs from its local neighbourhood.

$$OD(x) = \frac{\sum_{y \in S(x)} \rho(y)}{\rho(x) \cdot |S(x)|} \tag{13}$$

The proposed algorithm is the proportion of the average local neighbourhood density to the test point density. The data points with higher density compared to its neighbourhood is very likely to be surrounded by the dense neighbours, indicating that point would not be an outlier, and those with smaller density compared to its neighbours is expected to be an abnormal point. The steps involved in the proposed algorithm are presented in Algorithm 1.

Algorithm 1: Outlier Detection Algorithm

Input : X and k
Output: Outlier dataset OD_list

for *all* $x \in X$ **do**
  Compute the $kNN(x)$;▷ `get k nearest neighbors of x`
  Compute the $RNN(x)$; ▷ `get reverse nearest neighbors of x`
  $SNN(x)=\phi$; ▷ `initialize SNN(x)`
  for *each* $x_i \in kNN(x)$ **do**
    Compute the $RNN(x_i)$;
    $SNN(x) = SNN(x) \cup RNN(x_i)$;
  end
  Compute S(x) according to Eq. (9)
  Compute local density at point x using Eq. (10)
end
for *each* $x \in X$ **do**
  Compute OD for x using Eq. (13)
  $OD\_list \leftarrow$ Sort(OD, 'descending');
end

IV. Experimental Results and Analysis

The experimental analysis was performed to demonstrate the supremacy and efficacy of the proposed approach employing two 2-dimensional synthetic datasets. Also, the proposed algorithm was applied to five real-world datasets for further verification of the effectiveness of the proposed method. In the experiment, the results of our algorithm were compared with four existing approaches (LOF [11], COF [12], INFLO [13], and KDEOS [14]), since all these outlier detection algorithms focus on the unsupervised method of learning and share a standard parameter k. Nearly all of our algorithms required hyper-parameters. The default values are assigned as suggested in the literature to avoid complications to parameters in these outlier detection algorithms. All algorithms were implemented in R programming language, and run on a machine with an Intel(R) Core(TM) i7-4770 CPU at 3.40 GHz, 6 GB RAM and RAM frequency of 799.0 MHz.

A. Results Analysis

1) Synthetic Datasets: To demonstrate the effectiveness of the proposed approach, a comparative analysis was performed focused on two synthetic datasets. To assess the proposed approach in a harsh test environment, datasets were designed to take into account the different size of the cluster, cluster density degrees, and cluster models. Two important problems in outlier identification are: low density pattern [19] and local density pattern [11], both of which are included in the datasets. In prior studies [11], [20]–[23] these datasets were taken into

account. The outliers identified by our method in accordance with the previous results are coloured green in the following studies. These datasets are described in detail below.

Dataset 1, as shown in Fig.1a. It was designed to address the problem of local density, including three clusters with significantly different densities. Dataset 1, of which 45 are outliers, comprises a total of 1606 data points. The outputs detected by our proposed algorithm on this dataset are shown in Fig.1b. In our experimental settings, we set top$-n = 45$ (i.e. number of top outliers), where our proposed algorithm performs better than all methods of comparison. LOF and other methods of comparison misidentify normal samples in the red region as outliers and outliers in an area where the density of its neighbors differs significantly from normal points as inliers. Some actual outliers, too, cannot be identified by our algorithm. The experimental result of our proposed algorithm is good and achieved the AUC score of 0.9568 at $k = 70$, which is the highest among the comparison algorithms, as shown in Fig.2. From this, we can see that the detection performance of other methods of comparison is not as good as our proposed algorithm.

Dataset 2, as illustrated in Fig.3a includes low-density patterns and various degrees of different clusters. A large parabolic cluster comprising 1000 items is present in this dataset. There are three clusters within the parabola generated by Gaussian distribution, and every cluster has 100 items. There is a random distribution of one hundred outliers. The outliers detected by our algorithm are shown in Fig.3b. The AUC scores for each method with different values of k are shown in Fig.4. From this, it can be seen that INFLO's performance is the worst. INFLO incorrectly detects the points as outliers located in the red regions, that belongs to the different clusters. Further, one can note that the AUC scores of our algorithm are close to 0.89, which is the highest among these comparing algorithms.

2) Real datasets: To further investigate the effectiveness of our proposed outlier detection algorithm, We conducted a sequence of tests on five real-world datasets. All these datasets were obtained from the UCI machine learning repository[1] and were used to evaluate the performance of outlier detection algorithms in the literature. Table. I summarizes the dataset detail showing that the number of instances, dimensions and percentage of outliers, in a dataset differ significantly from others.

TABLE I: The characteristics of real dataset

Real data	***Instances***	***Dimensions***	***Outliers***
Hepatitis	80	19	13
Cardiotocography	2126	21	471
Annthyroid	7200	21	534
HeartDisease	270	13	120
Arrhythmia	450	259	206

[1]http://www.archive.ics.uci.edu/ml/

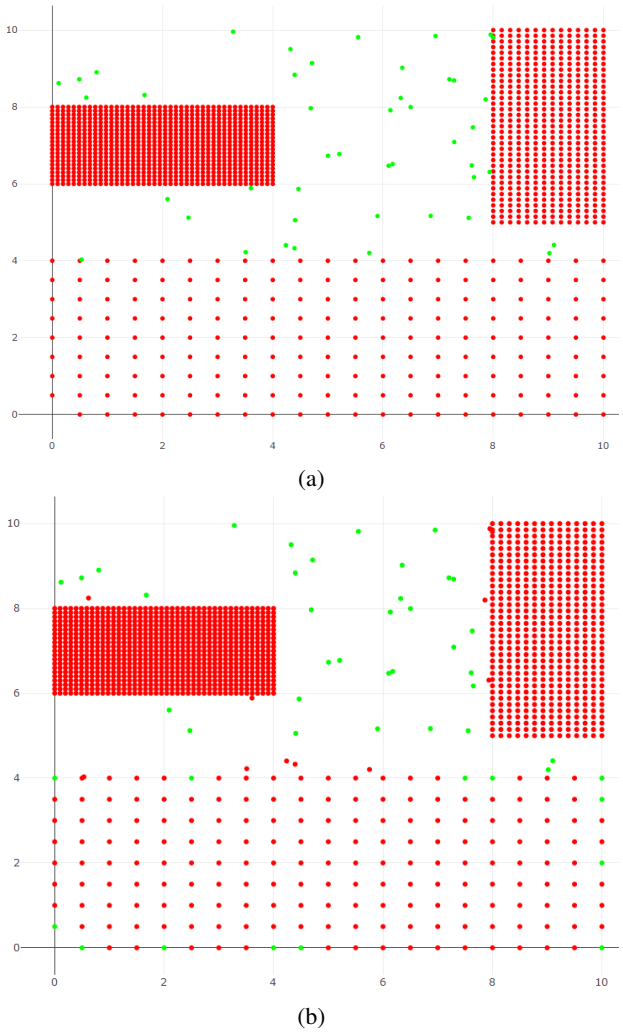


Fig. 1: (a) Original data 1, and (b) Outliers detected by our proposed algorithm

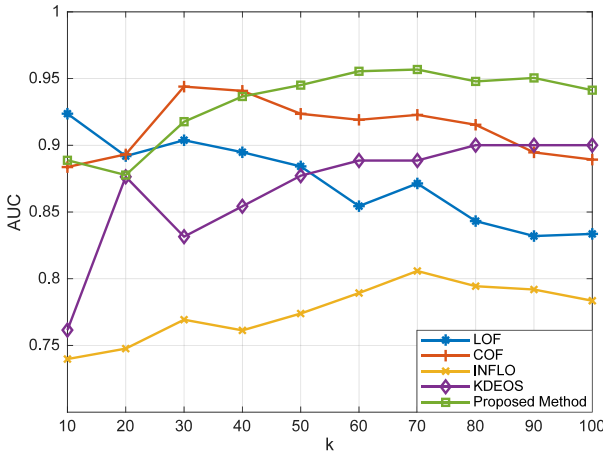
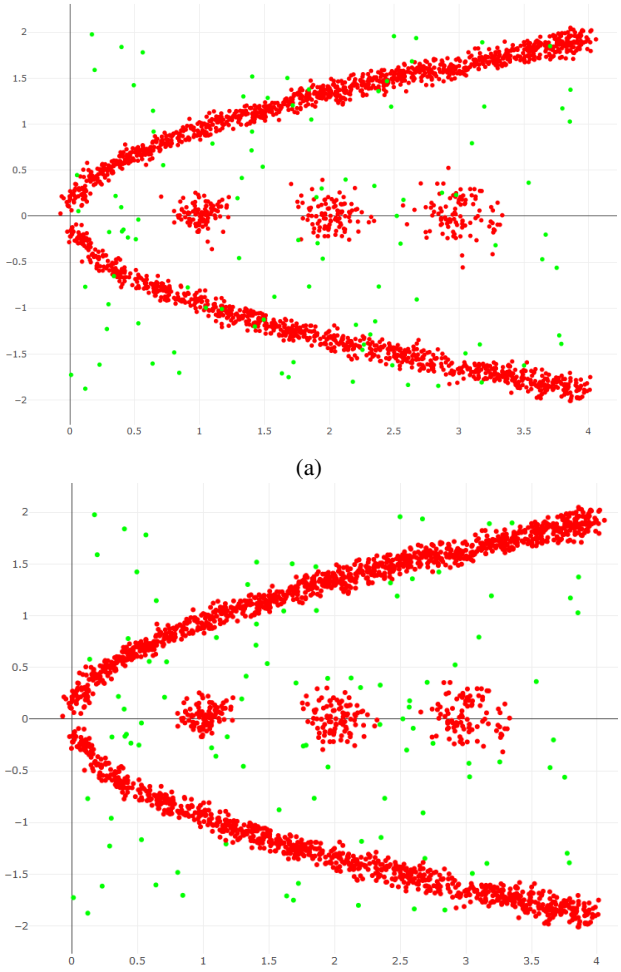


Fig. 2: Detection performance (AUC scores) of 5 methods on synthetic data 1.



(a)

(b)

Fig. 3: (a) Original data 2, and (b) Outlier detected by our proposed algorithm.

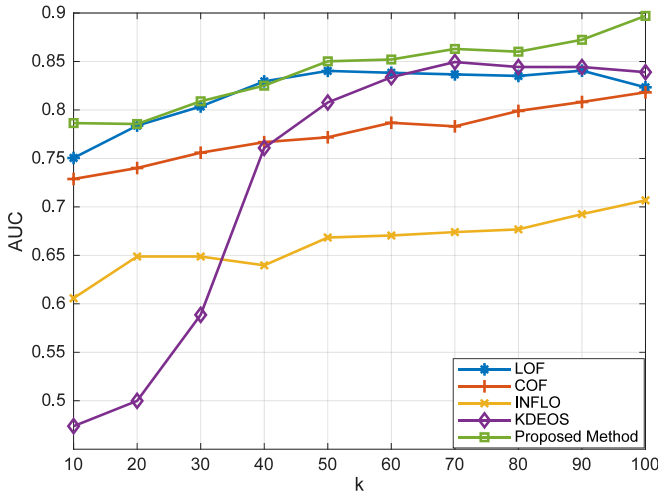


Fig. 4: Detection performance (AUC scores) of 5 methods on synthetic data 2.

Most of these datasets were primarily used for the evaluation of classification methods. For outlier detection, the objects from minor class was considered as outliers, and the objects of other class(es) were called as regular or inlier ones. For instance, a dataset of Arrhythmia that is classified as common or cardiovascular arrhythmias. There are a total of 14 types of arrhythmias and 1 type that combine all other kinds. As we see minor class as outliers, we treat healthy persons as inliers and arrhythmic patients as outliers. The same technical trick was done on the other datasets also.

For a comparative study, we run a number of experiments for outlier detection over each real datasets and ranked the observations according to their corresponding scores. Since all these outlier detection approaches along with the proposed one have a specific parameter i.e. k, the AUC scores are calculated at various k values ranging from 5 to 100 (or less, if the number of instances are less than 100). We summarizes the AUC scores of 5 detection methods on each experimental datasets in Fig.5. From these experimental results shown in Fig.5, it can be observed that a single outlier detection algorithm can not perform well for each dataset type.

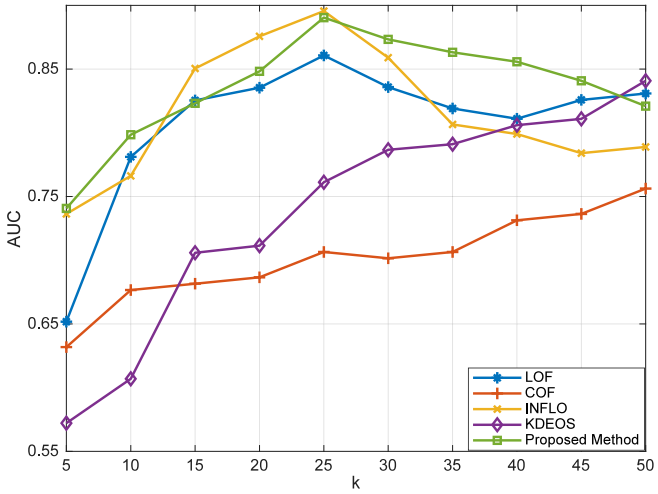
Although the proposed method is not giving the best detection result for each dataset, It is important to note that in contrast to other existing methods (like LOF, COF, etc.), the proposed method has superior performances for most datasets. Further, we can see that our proposed approach has a small variation in the AUC scores in comparison to other comparable approaches, suggesting that the proposed approach is more stable in comparison to the change in parameter k.

The most popular density-based approach, LOF, performs well, especially for a dataset with fewer outliers, but the efficiency is, on average, not superior to our algorithm. The COF approach demonstrates fairly good efficiency with the Arrhythmia dataset (AUC value 0.82 at $k = 10$) but is inconsistent with k. In most of the datasets, the peak performance of INFLO is much lower than the proposed approach, but it shows stability in terms of k. The KDEOS algorithm demonstrates comparably poor efficiency over most datasets in all comparable algorithms because KDEOS is a kernel-based approach where kernel selection can be customized to different issues.

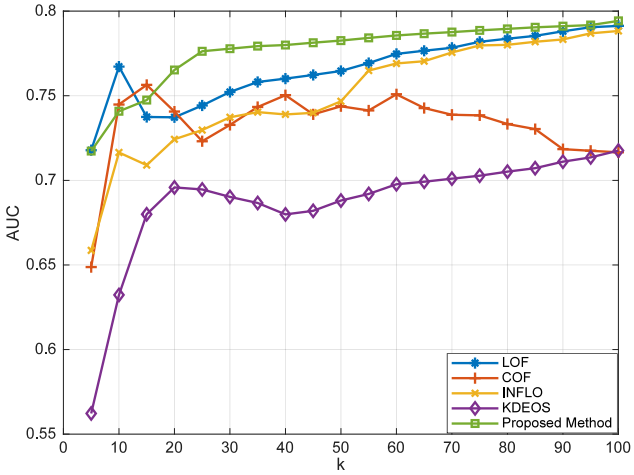
These experimental results demonstrate that our proposed approach can attain better performance over both synthetic and real datasets, and it can also solve the problem of low-density patterns and local density patterns up to some extent. Further, we performed a statistical test for analyzing the significant differences in the results.

B. *Statistical test*

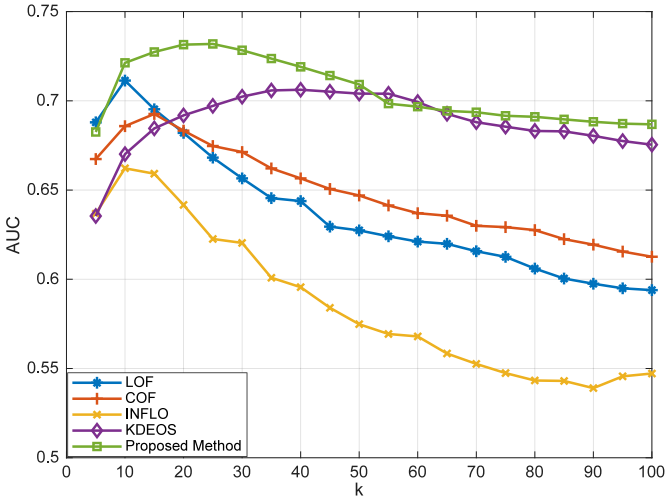
To check a significant difference between the results of outlier detection algorithms with a 90% confidence level ($\alpha = 0.10$), we conducted a paired $t-test$ Table. II. Each entity in a paired $t-test$ is evaluated twice, which results in *pairs* of observations. As with many statistical procedures, the paired $t-test$ also has two competing hypotheses: null hypothesis (H_0) and the alternative hypothesis (H_1). The null hypothesis



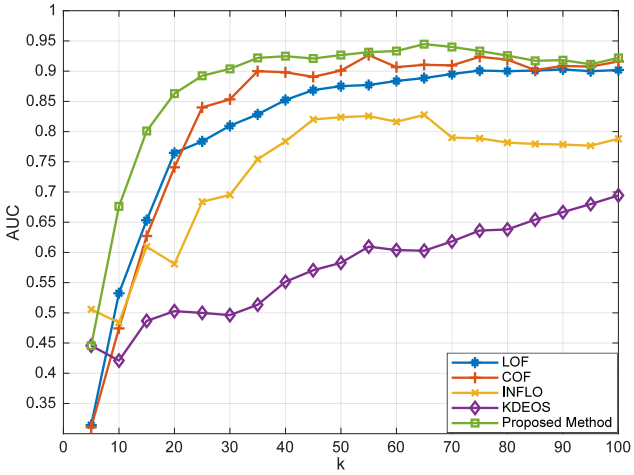
(a)



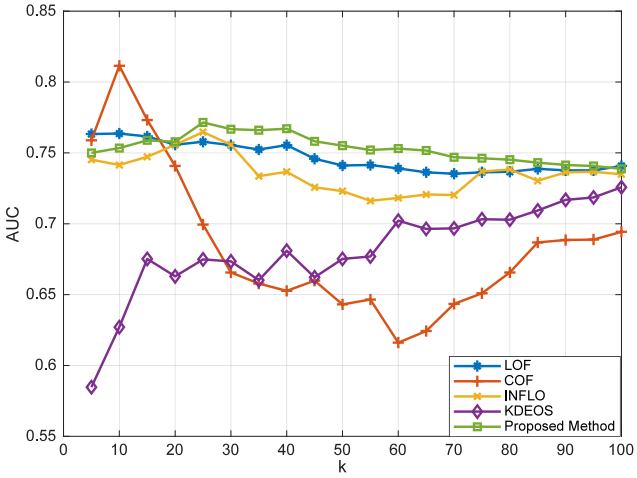
(b)



(c)



(d)



(e)

Fig. 5: Detection performance (AUC) of 5 methods for (a) Hepatitis (b) Cardiotocography (c) Annthyroid (d) HeartDisease and (e) Arrhythmia datasets.

(H_0) implies that in a specified metric, the performance of each algorithm is the same. In contrast, the alternative hypothesis (H_1) defines a distinct performance of at least one technique. If the calculated probability is low (the value of p is below the level of confidence) then the hypothesis (H_0) is discarded, which indicates that two or more techniques are considerably different.

Let's assume that, the performance score over the i^{th} data (out of N samples) for two different methods is ${s_1}^i$ and ${s_2}^i$. In our study, we considered AUC as a performance score. For i^{th} dataset, the difference between two performance scores can be computed as:

$$\delta_{12}^i = {s_1}^i - {s_2}^i. \tag{14}$$

Now, the average difference will be:

$$\overline{\delta_{12}} = \frac{1}{n}\sum_{i=1}^{n}\delta_{12}^i. \tag{15}$$

Then, the $t-value$ for two methods will be:

$$t_{12} = \frac{\overline{\delta_{12}}}{\sigma_{12}/\sqrt{N}} \tag{16}$$

where σ_{12} is the standard deviation over N datasets, which is defined as:

$$\sigma_{12} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\delta_{12}^i - \overline{\delta_{12}})^2} \tag{17}$$

TABLE II: The t-values for each pair of methods

	LOF	***COF***	***INFLO***	***KDEOS***	***PROPOSED***
LOF		0.684	0.937	1.880	−2.894
COF			0.011	0.969	−2.201
INFLO				1.773	−2.649
KDEOS					−2.184
PROPOSED					

This t-statistic is distributed following t-distribution of $N-1$ degrees of freedom. In Table. III, the associated p-values are given.

TABLE III: p-values for each pair of methods.

	LOF	***COF***	***INFLO***	***KDEOS***	***PROPOSED***
LOF		0.2657	0.2009	0.0666	0.0221
COF			0.4957	0.1937	0.0462
INFLO				0.0754	0.0285
KDEOS					0.0471
PROPOSED					

TABLE IV: Significance results for 90% level of confidence

	LOF	***COF***	***INFLO***	***KDEOS***	***PROPOSED***
LOF		0	0	−	+
COF			0	0	+
INFLO				−	+
KDEOS					+
PROPOSED					

The "+" sign indicates a substantial difference in the performance and suggests that the row method is superior to the column method at 90% confidence level. The "−" sign shows a significant difference in the results at the same (90%) confidence level and suggests that column method is superior to the corresponding row method, and the "0" indicates there is no significant difference in the results. From Table. IV, we can see that the proposed method improved the performance to a significant level. In conclusion, the proposed approach delivers better detection performance for high-dimensional, imbalanced datasets, and it solves the issues associated with existing methods up to some extent.

V. Conclusion

This paper presents a new approach for detecting local outliers in a dataset. The proposed approach is based on the notion of kNN kernel density (NKD) estimation, resulting in a new metric scoring the degree of outlier-ness. In the proposed approach, each object is assigned a local outlying degree. In particular, the local outlying degree is focused on the notion of local density in which its neighbours give the locality. Further, our proposed algorithm forms an extended neighbourhood in a novel way by combining three neighbours: kNN, RNN, and SNN of an object for estimating their local density. The proposed algorithm calculates the local outlying degree as a relative measure of an object's local density to its neighbour's local densities. An object that is higher in density compared to its neighbours will most likely be surrounded by dense regions and indicates that it would not be an outlier. The lower density data points in comparison with their neighbours are a promising outlier candidate.

With the use of synthetic datasets, the proposed technique has confirmed that the outliers of low-density patterns and local density patterns are precisely detected. The experimental results clearly show that the proposed method solves the problem of low-density patterns and local density patterns up to some extent. Our algorithm shows superior performance in real datasets too. In the future, the proposed algorithm can be extended to parallel and distributed environments so that our algorithm can efficiently process high-dimensional data.

Acknowledgment

We would like to show our gratitude to Professor Jin Ningour from Department of CSE, University of Electronic Science and Technology of China, and Professor J. Huang from College of Computer Science, Chongqing University, China for providing some datasets for this research work. We would also like to show our gratitude and thanks to the Department of Computer Science and Engineering, IIT (ISM), Dhanbad, India, for providing the facility and support for this research work.

References

[1] D. M. Hawkins, *Identification of outliers*. Springer, 1980, vol. 11.

[2] D. Ramotsoela, A. Abu-Mahfouz, and G. Hancke, "A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study," *Sensors*, vol. 18, no. 8, p. 2491, 2018.

[3] M. Kirlidog and C. Asuk, "A fraud detection approach with data mining in health insurance," *Procedia-Social and Behavioral Sciences*, vol. 62, pp. 989–994, 2012.

[4] M. Artís, M. Ayuso, and M. Guillén, "Detection of automobile insurance fraud with discrete choice models and misclassified claims," *Journal of Risk and Insurance*, vol. 69, no. 3, pp. 325–340, 2002.

[5] D. E. Denning, "An intrusion-detection model," *IEEE Transactions on software engineering*, vol. SE-13, no. 2, pp. 222–232, 1987.

[6] E. W. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," *Decision support systems*, vol. 50, no. 3, pp. 559–569, 2011.

[7] K. Y. Chan, C. Kwong, and T. C. Fogarty, "Modeling manufacturing processes using a genetic programming-based fuzzy regression with detection of outliers," *Information Sciences*, vol. 180, no. 4, pp. 506–518, 2010.

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[9] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

[10] T. N. Tran, R. Wehrens, and L. M. Buydens, "Knn-kernel density-based clustering for high-dimensional multivariate data," *Computational Statistics & Data Analysis*, vol. 51, no. 2, pp. 513–525, 2006.

[11] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *ACM sigmod record*, vol. 29. ACM, 2000, pp. 93–104.

[12] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 535–548.

[13] W. Jin, A. K. Tung, J. Han, and W. Wang, "Ranking outliers using symmetric neighborhood relationship," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2006, pp. 577–593.

[14] E. Schubert, A. Zimek, and H.-P. Kriegel, "Generalized outlier detection with flexible kernel density estimates," in *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 2016, pp. 542–550.

[15] A. Wahid, A. Rao, and K. Deb, "A relative kernel-density based outlier detection algorithm," in *2018 12th International Conference on Software, Knowledge, Information Management & Applications (SKIMA)*. IEEE, 2018, pp. 1–7.

[16] J. Gao, W. Hu, Z. M. Zhang, X. Zhang, and O. Wu, "Rkof: robust kernel-based local outlier detection," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2011, pp. 270–283.

[17] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2007, pp. 61–75.

[18] A. Wahid and A. C. S. Rao, "Rkdos: A relative kernel density-based outlier score," *IETE Technical Review*, pp. 1–12, 2019.

[19] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 535–548.

[20] J. Ha, S. Seok, and J.-S. Lee, "Robust outlier detection using the instability factor," *Knowledge-Based Systems*, vol. 63, pp. 15–23, 2014.

[21] H.-P. Kriegel, P. Kroger, E. Schubert, and A. Zimek, "Interpreting and unifying outlier scores," in *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM, 2011, pp. 13–24.

[22] J.-S. Lee and S. Olafsson, "A meta-learning approach for determining the number of clusters with consideration of nearest neighbors," *Information Sciences*, vol. 232, pp. 208–224, 2013.

[23] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel, "On evaluation of outlier rankings and outlier scores," in *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 2012, pp. 1047–1058.