IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 February 2023, accepted 27 February 2023, date of publication 6 March 2023, date of current version 15 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3253022

APPLIED RESEARCH

PAACDA: Comprehensive Data Corruption Detection Algorithm

CHARVI BANNUR[1], (Student Member, IEEE), CHAITRA BHAT[1], (Student Member, IEEE), KUSHAGRA SINGH[1], (Student Member, IEEE), SHRIRANG AMBAJI KULKARNI[2], (Senior Member, IEEE), AND MRITYUNJAY DODDAMANI[3]

[1]Department of Computer Science and Engineering, People's Education Society, Bengaluru 560085, India
[2]Department of Computer Science and Engineering, National Institute of Engineering, Mysore 570008, India
[3]School of Mechanical and Materials Engineering, Indian Institute of Technology–Mandi, Mandi, Himachal Pradesh 175075, India

Corresponding author: Shrirang Ambaji Kulkarni (sakulkarni@nie.ac.in)

ABSTRACT With the advent of technology, data and its analysis are no longer just values and attributes strewn across spreadsheets, they are now seen as a stepping stone to bring about revolution in any significant field. Data corruption can be brought about by a variety of unethical and illegal sources, making it crucial to develop a method that is highly effective to identify and appropriately highlight the various corrupted data existing in the dataset. Detection of corrupted data, as well as recovering data from a corrupted dataset, is a challenging problem. This requires utmost importance and if not addressed at earlier stages may pose problems in later stages of data processing with machine or deep learning algorithms. In the following work we begin by introducing the PAACDA: Proximity based Adamic Adar Corruption Detection Algorithm and consolidating the results whilst particularly accentuating the detection of corrupted data rather than outliers. Current state of the art models, such as Isolation forest, DBSCAN also called "Density-Based Spatial Clustering of Applications with Noise" and others, are reliant on fine-tuning parameters to provide high accuracy and recall, but they also have a significant level of uncertainty when factoring the corrupted data. In the present work, the authors look into the most niche performance issues of several unsupervised learning algorithms for linear and clustered corrupted datasets. Also, a novel PAACDA algorithm is proposed which outperforms other unsupervised learning benchmarks on 15 popular baselines including K-means clustering, Isolation forest and LOF (Local Outlier Factor) with an accuracy of 96.35% for clustered data and 99.04% for linear data. This article also conducts a thorough exploration of the relevant literature from the previously stated perspectives. In this research work, we pinpoint all the shortcomings of the present techniques and draw direction for future work in this field.

INDEX TERMS Adamic Adar algorithm, corrupted datasets, outlier detection, probabilistic models, statistical models, unsupervised learning.

I. INTRODUCTION

Ever since technological evolution dawned upon humankind there has been massive progress in about every domain that the human mind can perceive. The major credit for driving the ongoing technological advancement lies in intensive amounts of data, without which the majority of this industry might come to a standstill [1]. Data seems to have reached such a significant level of importance that, it is often deemed that the companies possessing larger amounts of data seem to have a monopoly in that sector. Data often lays the foundation for the development, growth and maturity of an algorithm or technology. In today's world data is significant to all organizations and thereby it becomes all the more crucial to protect this critical entity from being manipulated by malicious means [1].

A dataset can undergo a snowball effect with just a few changes, which could ultimately be detrimental. Even though

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa M. Fouda.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

there are multiple unethical ways to corrupt data, persistent research has been conducted throughout time to identify efficient ways to learn about data corruption, to name a few commendable works in detecting data corruption including [2], [3], [4], and [5]. Before figuring out a unique approach for identifying data corruption we delved deep into other pre-existing methods available for the detection of data corruption primarily concerning outliers. The deep study of various approaches provided us with insightful knowledge of various algorithms having varying levels of accuracy when tested against the dataset containing corrupted data rather than outliers.

K-means clustering uses clusters and their centroids as part of an unsupervised technique to address issues with categories and their classification [6], [7]. DBSCAN is another clustering-based technique however it tends to perform well with data containing clusters with similar density, as it finds core samples of high density and expands upon them. When applied to identify outliers in the supplied dataset, both of these methods provided a satisfactory level of accuracy [8], [9].

Moving along the research we started exploring methodologies such as Isolation forest, Elliptic envelope outlier detection and histogram-based outlier detection. Isolation forest provides us with an algorithm through which we can partition the dataset features in order to identify the outliers which exceed the defined range [10], [11]. The Elliptic envelope model tends to create an ellipse around the scatter plot for the dataset and all points lying outside its boundaries signify the outliers present in the dataset [12], [13]. Another approach involving plotting and analyzing histograms is the Histogram based algorithm for outlier detection (HBOD) method which is also an effective unsupervised method to detect anomalies. These algorithms also have a fairly decent level of accuracy in terms of identifying anomalies in the dataset [14], [15].

Algorithms such as 'Principal Component Analysis' (PCA),'DeepSVDD' and 'Rotation based Outlier Detection'(ROD) were also looked at to tabulate the level of accuracy in predicting the outliers for the synthetically generated dataset. To name a few PCA [16], [17], ROD [18], [19], Local Outlier Factor [20], [21], DeepSVDD [22], [23] and more were used. In spite of the distinct strategies given out by various models, the unique methods proposed as a part of this research stood out in the following metrics Accuracy, Recall, Precision, Sensitivity and F1 score.

Adamic Adar is a promising algorithm for data correlation in graph networks and hence has increasing amount of potential in data corruption detection. The study mentioned above led to the realization that it is feasible to avoid the current work's inefficiencies. The motivation behind this work would be to consolidate the current work in this field of study and enhance the accuracy of the present corruption detection algorithm by leveraging the Adamic Adar algorithm's prominance in data correlation. The novel method proposed as a part of the research largely revolves around a graph-based algorithm is Adamic Adar. Adamic Adar gives us access to the Adamic Adar index, which aids in anticipating links, particularly in areas like social networks [24]. By taking into account how many common links there are between two nodes, the Adamic Adar index is determined [25]. The research puts forth a modified approach of Adamic Adar called PAACDA (Proximity based Adamic Adar Corruption Detection Algorithm) which when put into use detect the data corruption provides us with the best accuracy compared to the above-mentioned algorithms.

After a deep study involving the existing methods for the detection of corruption and the novel method presented as a part of this research work, attention was diverted towards figuring out feasible methods to revert back to the original data for the ones deemed as corrupt. However, this is beyond the scope of this study. The linear regression approach works considerably well for two attributes datasets for data regneration however most datasets deal with humongous amounts of data consisting of multiple features. GANs (Generative Adversarial Networks) can be a plausible approach to regenerating corrupted data using the generator and discriminator model [26], [27]. However, utilizing the various evolved forms of GAN, mainly tabular GANs in order to address the problems of regeneration of contaminated still remains unexplored.

The remainder of this article is divided into the following sections. After highlighting some related work about the approaches taken into consideration for this study in Section II, we go on to illustrate the data and methods used in Section III, as well as the proposed methodology to tackle the issue in Section IV. In Section V, we put forth the results that indicate the cogency of our strategies, and in Section VI, we present the conclusions and future scope of this research.

II. RELATED WORK

A key area of research that has numerous practical applications is anomaly identification in a given dataset. As a result, this topic has frequently been the focus of research. Multiple approaches utilizing various aspects of the dataset have been proposed to detect anomalies however only few methodologies lay emphasis on the detection of corrupted data which would further provide the most efficient results with respect to varying dataset sizes, higher dimensionality or varying degrees of corruption present. A study by Chandola et al. in their publication [2] compares numerous anomaly detection methods for diverse applications. By contrasting the benefits and drawbacks of various techniques, Hodge and Austin [28] conducted a review of outlier detection methods. An overview of cutting-edge methods for spotting suspicious behaviour is presented by Patcha and Park [29] Jiang et al. [30] together with detection scenarios for several real-world settings.

Dimensionality reduction approaches and the underlying mathematical understandings are categorized by Sorzano et al. [31]. The issues with anomaly detection are further laid out by a number of other reports, including papers by Gama et al. [32], Gupta et al. [33], Heydari et al. [34], Jindal and Liu [35], and many more.

Outliers make up the majority of anomalies that can exist in a dataset. The first method based on distance for detection of outliers was put forth by Knorr et al. [36], and Ramaswamy et al. [37] expanded on it by suggesting that the greatest n locations with highest Pk be supposed outliers (Pk(p) signifies the kth nearest neighbour corresponding to p). They used a clustering technique to separate a dataset into several categories. To improve the success of outlier detection for these groups, batch processing and pruning may be beneficial [38]. Deviation-based outlier detection was another method that was suggested for effectively detecting outliers. Objects or data points that vary significantly from the bulk of data points constitute outliers. Therefore, outliers are frequently called deviations [39] as given by the name deviation-based outlier detection.

Several other methods have been invented over the years to detect anomalies, to name Breunig et al. [21] developed a method based on density. Cluster-based anomaly identification methods pinpointed anomalies by eliminating clusters from the actual dataset [40] or by classifying small clusters as outliers [41]. Additionally, Aggarwal and Yu [42] proposed a novel strategy for catching outliers that is remarkably effective for extremely elevated dimensional datasets. Their methodology focuses on finding locally sparse lower dimensional projections which are otherwise difficult to differentiate using brute force methods due to the vast amount of possible combinations. However, the study is inclined towards detection of outliers and does not focus on the detection of corrupted or modified datasets.

Li et al. [43] in their paper proposed a unique outlier detection approach called 'Empirical Cumulative distribution-based Outlier Detection' (ECOD). This method uses the empirical cumulative distribution to measure outlier values present in the dataset. They extensively applied it to 30 datasets which showed that ECOD outperformed the existing state of the model as it is fast and scalable. However, the method doesn't deal with an outlier that might not be in either the left or right tails and demands readers to come up with another promising route that is, to find a mechanism to expand ECOD to such environments while keeping it quick and scalable.

The bulk of them, meanwhile, are primarily focused on outlier identification without paying much attention to data that contains corrupted values. Many cutting edge poisoning and outlier identification practices have been developed and they may generally belong to one of the following categories: distribution based [43], [44], [45], depth based [46], distance based [47], [48], [49], density based [50], [51], cluster-based [52], [53], [54] and generative models [55].

Thus in this work the following research objectives have been addressed:

- To explore the efficacy of various unsupervised models to detect the corrupted data in an efficient manner and provide a comprehensive and detailed review.
- To propose a novel method PAACDA, an unsupervised model to detect corrupted data in a more accurate manner.

Despite the many different approaches that have been suggested, each of which has its own set of benefits and downsides, the search for the ideal, all-encompassing algorithm never seems to stop. However, here we present a novel practice that, when thoroughly compared to prior current edge approaches and evaluated against various sets of data sizes, yields satisfactory to superior results.

III. MATERIALS AND METHODS

In the section III we put forward our proposed approach to pursue the data corruption detection problem. We elucidate the process in detail in the subdivisions below. An overview of our approach is illustrated in FIGURE 1.

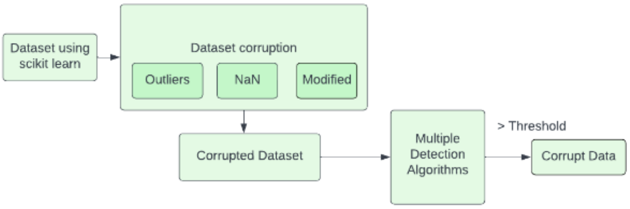


FIGURE 1. Illustration of proposed methodology.

A. EXPERIMENTAL ENVIRONMENT

All the tests were implemented on GoogleColab, Python 3.7.13 [62] version was used for implementing all the algorithms. We used Keras [63] backend as the deep learning framework. We plan to make the research and dataset fully reproducible on GitHub to the research community.

B. DATASETS AND CORRUPTION TECHNIQUE

The scikit-learn library for Python was used to create the synthetic datasets utilised in the following study. The datasets are linear and clustering in nature. Testing must be done on a wide variety of data and corruption rates in order to detect corrupted data in datasets that have been tainted. We use univariate data produced especially for this research paper. The authors would like to mention that this research work focuses solely on corrupted data and not just outliers and anomalies unlike the work mentioned above. However, the corrupted data may contain outliers and anomalies as part of the corruption. In light of this, we will now provide a description of our curated dataset and a discussion of the curation methods.

The linear dataset has the following parameters:

- Number of features =1
- Noise = 10
- The graph for the same is shown in FIGURE 2.

The clustering dataset has the following parameters:

- Number of features = 2

TABLE 1. A review of prominent data corruption detection techniques based on different algorithms.

Author Citation	Adopted Methodology	Challenges	Features
Hans-Peter Kriegel, Jiirg Sander, Martin Ester and Xiaowei Xu. [8]	Density based model DBSCAN, which is intended to find the noise and the clusters in a spatial database, is presented in this work.	The algorithm fails to be effective for larger databases.	The final conclusion presented as a result of the experiments show that DBSCAN is substantially better than the well-known algorithm CLARINS at finding clusters of arbitrary forms. Moreover, the results of the studies indicate that DBSCAN is at least 100 times more efficient than CLARANS.
Youguo Li, Haiyan Wu [56]	The original K-means algorithm's two major drawbacks of increased dependence on the initial focal point selection and ease of trapping oneself in a local minimum were both effectively addressed by the upgraded K-Means algorithm.	The Paper has further potential to work on using the improved K-means for outlier detection.	The authors discovered that the initial cluster focal points of standard K-Means are too random, resulting in the formation of unstable clusters. While the improved K-Means yield a slightly better result.
Jillepalli, Yacine Chakhchoukh, Mohammad Ashrafuzzaman, Saikat Das, Ananth A., Frederick T. Sheldon. [12]	The authors create and test an unsupervised machine intelligence strategy called Elliptical Envelope.	The rate at which the attacks are detected could be improved.	When compared to the other five unsupervised methods considered, the final evaluation of this method based on eliptic envelope puts forth one of the the best detection rate and the lowest false positive rate.
Kai Ming Ting, Fei Tony Liu, Zhi-Hua Zhou. [10]	The authors present a novel approach in this study and show how a tree-like structure could be used to efficiently isolate each and every instance present. As a result of their vulnerability to isolation, inconsistencies are isolated closer to the base of the tree, whereas normal points are separated further down the tree.	This algorithm is not very effective at detecting local anomaly points, which reduces its accuracy.	Isolation Forest outperforms a near linear time complexity technique based on distance , Location of factor, and Random Forest in terms of area under the curve and execution time, primarily in the case of large data sets.
Zhao, Nicola Botta, Zheng Li, Yue Cezar Ionescu, Xiyang Hu [57]	The writers suggest an innovative, interpretable outlier identification algorithm that is parameter-free and has excellent performance. Conduct thorough tests on 30 benchmark datasets to demonstrate that COPOD is one of the fastest methods and outperforms in the majority of scenarios. Make a simple Python implementation available for replication.	The COPOD function could possibly incorporate corruption rate as a parameter in the PyOD library.	After extensive analysis with data pertaining to the real world, the authors concluded that the performance achieved through COPOD is quite competitive when compared to other significant outlier detection models in terms of detection accuracy and computing cost.
Kenneth Joseph PAUL, Harilal R and Ramji M1 [58]	The authors in the present work applied median and mean absolute deviation methodologies for data smoothing. The model implemented as a result of this technique showed significant reduction of the noise levels in the strain fields.	Further extensive investigation and analysis involving the proposed methodology on actuarial raw data is essential in order to highlight its impact.	For the displacement field produced, a complete field smoothing technique has been created. Algorithms utilizing the mean absolute deviation and MAD are employed to smear the displacement data before calculating strain. Here, the disc-under-diametral-compression problem is considered. An algorithm for boundary encoding was created.

TABLE 1. ***(Continued.)*** **A review of prominent data corruption detection techniques based on different algorithms.**

Zheng Li, Yue Zhao [59]	The authors in this work presented a novel method of ECOD (Empirical-Cumulative-distribution-based Outlier Detection), an unsupervised method for outlier detection.	When outliers are evenly distributed throughout normal points or concealed in the center of normal points, ECOD's performance suffers in all dimensions.	The authors in this work conducted extensive tests on 30 benchmark datasets and discovered that, in terms of precision, effectiveness, and scalability, ECOD beats 11 state-of-the-art baselines.
Bryan Hooi, Adam Goodge, Ng See Kiong, Ng Wee Siong [60]	In this paper, the authors suggest LUNAR, a novel outlier detection technique based on graph neural networks. In order to detect abnormalities, LUNAR learns to trainably use data from each node's closest neighbors.	Testing the model on a broader range of datasets is necessary; as the author pointed out, KNN delivered better performance than the proposed approach on one of the datasets considered in this study.	The authors of this paper show how their method performs much better than the baseline set by other approaches, including methodologies involving deep learning, on a variety of datasets. The aforementioned approach is unique as it can learn from all incoming information from neighbors, allowing it to maintain an exceptional performance for varying neighborhood sizes and stand much ahead when compared to other local outlier methods.
Abdenour Bounsiar, Michael G. Madden [61]	In this paper the authors review the One Class SVM algorithm by proposing a geometric justification for the postulate which utilizes a gaussian kernel for the separation of the target data from the rest of the space.	Even while the results supported the hypothesis, it should be highlighted that this was only feasible because the data had two particular qualities that were taken into account.	The study highlighted the consistency of the gaussian kernel when compared to linear,sigmoid or polynomial kernel in effectively segregating the target data with respect to outliers.
Raed Alsini, Omar Alghushairy, Terence Soule and Xiaogang Ma [20]	This study examines the literature on local outlier identification strategies in static and stream contexts, and lays emphasis on algorithms involving LOF method. It gathers and organizes existing LOF techniques before classifying and analyzing their features. The research also expands upon the pros and cons of those methodologies, as well as a number of future lines of research for the advancement of local outlier detection approaches pertaining to streams of data.	The Weakness of various algorithms revolving around Local Outlier Factor Detection has been rightly discussed by the authors.	This research looks at the various proposed local outlier identification techniques and discusses the difficulties of detecting local outliers in stream systems. Based on the review's findings, the report also proposes a strategy to improve the LOF's effectiveness in stream situations.
Nico Gornitz, Lucas Deecke, Lukas Ruff Robert A. Vandermeulen, Shoaib A. Siddiqui Alexander Binder Emmanuel Muller ¨ Marius Kloft [22]	The novel approach proposed in this paper is Deep Support Vector Data Description which is trained with the objective of effective anomaly detection.	The work focuses mainly on anomaly detection and is not inclined towards the detection of corrupted data.	One of the theoretical properties of the authors' methodology that they have demonstrated is the v-property, which allows for the incorporation of a prior assumption regarding the amount of outliers which could be present in the dataset. Deep SVDD performs well both numerically and qualitatively, according to their experiments.

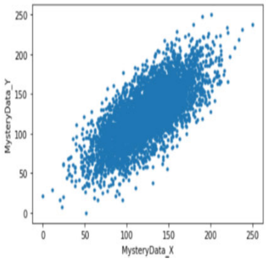


FIGURE 2. Representing linear generated data for small dataset.

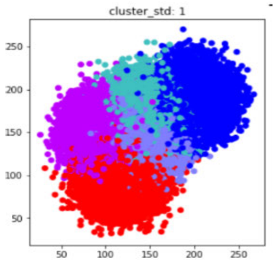


FIGURE 3. Representing clustering data generated for small dataset.

- Number of centres = 5
- The graph for the same is shown in FIGURE 3.

The datasets are of 3 different sizes:

- 10,000 samples (Small)
- 40,000 samples (Medium)
- 75,000 samples (Medium-Large)

The datasets are further categorised on the basis of the percentage of corruption:

- 20 % corrupted values
- 40 % corrupted values
- 60 % corrupted values

TABLE 2. Summarizing the different types of dataset and corruption levels used.

Dataset type	Corruption rate	Dataset size
Small	20% 40% 60%	10,000
Medium	20% 40% 60%	40,000
Medium-Large	20% 40% 60%	75,000

Thus in the present work, a total of 18 datasets of varying sizes and corruptions were used to demonstrate the impact on the 16 underlying models and our proposed Proximity based Adamic Adar Corruption Detection Algorithm (PAACDA). The data points in these provenant datasets are corrupted at random in every conceivable way, including by substituting fake values for actual ones, outliers, and missing data (0 or NaN). The most fundamental form of data corruption includes deleting the data from the datasets, which is similar to how lost system data is frequently experienced. This is extended to more complex situations, including replacing the original value with exaggerated and incorrect ones. To explore in more detail, a piece of code is used to randomly select cells with random rows and columns and replace them with a completely random value. The final contaminated dataset is then retrieved, and different techniques are applied to precisely predict and address the corrupted values.

1) Realistic Data

The authors also conducted the data corruption detection analysis on a Realistic dataset to improve the practicality of the proposed methodology. The dataset used here is a standard corruption detection dataset - "The Complete Pokemon dataset", with 802 instances and a 3% corruption rate. The dataset was not additionally or synthetically corrupted. The various corruption detection models along with PAACDA were applied to preceisely predict the corrupted values.

C. METHODS

1) LOCAL OUTLIER FACTOR

LOF is a type of density-based system [64]. Outliers are segregated in density-based [65], [66] systems because anomalies emerge in low-density areas [67]. LOF compares a location's local density to that of k of its neighbours, indicating points with considerably lower density than their neighbours.

As Breunig et al. [21] in their work emphasised LOF as quite promising as it can detect relevant local outliers that earlier techniques could not find. They demonstrated that their strategy of discovering local outliers is effective for the datasets with closest neighbour searches. For other objects, they provide strict upper and lower constraints on the value of LOF, irrespective of if the MinPts nearest neighbours are from single or several clusters. In addition, they investigated the effect of MinPts parameter e on the value of LOF. The experimental findings show that their heuristic is effective. Eq. (1) calculates Average RD(Reachability Density) and Eq. (2) calculates Local RD which is reciprocal of RD.

$$A(u,v) = \frac{1}{k}\sum \max(\mathit{kth\_distanceof\_A's\_neighbour},\ \mathit{distance}(A, \mathit{kth\_neighbour})) \qquad (1)$$

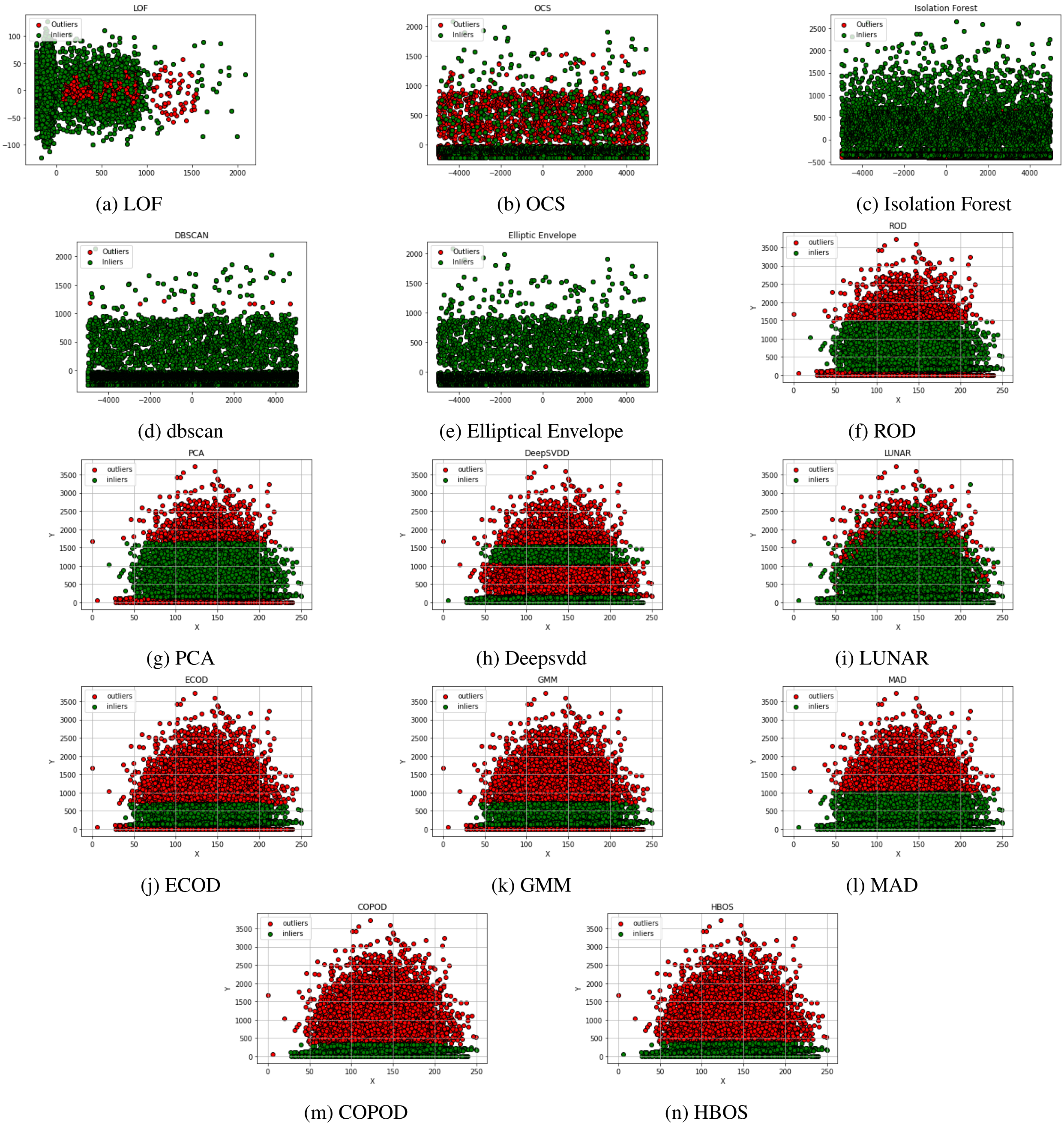
$$LRD = \frac{1}{RD} \qquad (2)$$

Furthermore, we obtain LOF as in Eq. (3), using which the points are classed as an outlier(-1) or not (1) [68].

$$LOF = \frac{1}{k}\frac{\sum_{i=0\ \ to\ \ k} LRD(i)}{LRD(A)} \qquad (3)$$

Typically, We usually recognise A as an anomaly when its LOF is lower than that of its k neighbours [69], [70], i.e. when LOF>1.1, albeit this depends on the context.

Lee and Tukhvatov [71] further proposed three augmentation schemes which are the LOF', LOF'', and GridLOF which optimised known state of the art model, LOF. By offering a new computation technique to find neighbours, the LOF''



(a) LOF (b) OCS (c) Isolation Forest

(d) dbscan (e) Elliptical Envelope (f) ROD

(g) PCA (h) Deepsvdd (i) LUNAR

(j) ECOD (k) GMM (l) MAD

(m) COPOD (n) HBOS

FIGURE 4. Representing corrupted data detected using various methods.

addresses scenarios that the LOF cannot effectively handle. By trimming inliers, the GridLOF enhances the efficacy of outlier identification. Because of its intricacy, this approach has several drawbacks, including a lengthy run time.

In the present work and on experimentation, the improved LOF resulted in a performance of 59.47% on the clustering dataset and 58.79% for linear data with a corruption rate of 20%, decreasing as the corruption percentage grew and imperceptible change in accuracy as the dataset size increased.

FIGURE 4(a) Shows the resulting corrupt data detected using the model.

2) ONE-CLASS SVM

For the past decade, SVM has been one of the most effective machine learning approaches. To discriminate between distinct classes of data, SVMs [72] use hyperplanes in multidimensional space. Naturally, SVM is utilised to handle multi-class classification challenges [73]. Semi-supervised variant of SVM, i.e One-Class SVM exists for the anomaly detection. In this case, the algorithm has been trained to comprehend ''normal,'' so that whenever new data is provided, it will determine if or not it must be included. Otherwise, the new data is labelled as anomalous or out of the norm.It employs the One-vs-All Anomaly Detection concept. It effectively optimises the distance from this hyperplane to (0,0) while keeping all data points away from the (0,0) (in feature space F). As a result, a binary function is generated that may detect input points where the density of the data is discovered. As a result, in a small region, it returns +1, and -1 for others [74], [75], [76]. FIGURE 4(b) depicts the model's detection of fraudulent data. Kernels can also be used to scale data to a higher dimensional for improved performance.

However, it has been found that a one-class SVM is prone to data outliers. Amer et al. [75] in their paper, apply two changes to one-class SVMs to make them more suited for unsupervised anomaly identification: robust and eta one-class SVMs. The main notion behind these changes is that outliers should influence less than regular cases. According to the research on UCI machine learning collections, their alterations are quite promising: The upgraded one-class SVMs outperform other standard unsupervised anomaly detection techniques on 2 of 4 datasets. The suggested eta one-class SVM in particular has yielded encouraging results. In the present work and on experimentation, the One-Class SVM achieved 76.82% on the clustering dataset and 72.28% for linear data with a corruption rate of 20%, dropping as the corruption percentage rose and showing no discernible change in accuracy as the dataset size increased. FIGURE 4(b) Shows the resulting corrupt data detected using the model.

3) K-MEANS CLUSTERING

Another well-known state of the art model wherein the data are divided into k groups in the K-means-based outlier identification [77] approach by allocating them to the closest cluster centres. Once assigned, we can calculate how far each item is from the cluster's centre and select those with the largest gaps as outliers. It determines the distance and its associated centroids. After determining the distance, the threshold ratio is chosen as a percentile. The data is deemed poisoned if the threshold ratio is exceeded.

The Elbow procedure is an empirical method to get the best value of k. Here, a metric known as ''Within Cluster Sum of Squares'' (WCSS) [78] as shown in Eq. (4) is determined with respect to its cluster centroid and recorded.

$$WCSS(m) = arg_o min \sum_{j=1}^{n} \sum_{y_i \in cluster} ||y_i - \bar{y}_j||^2 \tag{4}$$

where o is the collection of observations, m is the total set of predictors, y_i is the observational data point in cluster i and $\bar{y}_i$ is the sample mean in the cluster i [79].

The K-means algorithm should be manually supplied or must employ additional procedures with the number of clusters. Instead of finding global optimum solutions for non-convex problems, the K-means method becomes trapped on local optimum solutions. As Xiong et al. [80] in their paper for optimisation of initial clusters centers of text classification came up with an algorithm wherein the density parameter of the data items was used to calculate the first cluster centres, ensuring the logic of the initial cluster centres. Their new approach, to a considerable part, decreased the susceptibility of the K-means algorithm to the original cluster centres and produced improved text clustering results.

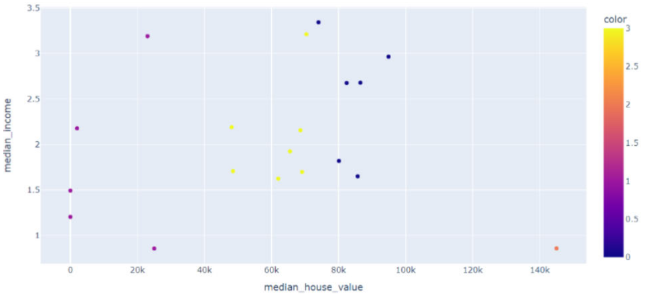


FIGURE 5. The corrupted data detected using K-means Clustering.

When there are anomalies in the data, they do have a crucial impact on all cluster centroids as it focuses on the mean of the values for its centre [80], [81]. Hence, the groups that would be made in the presence and absence of these outliers would vary greatly. The distances of the values from the centres would also vary and a new set of corrupted outliers would be produced every time. In the present work and on experimentation, the K-Means obtained 86.06% on the clustering dataset and 86.70% for linear data with a corruption rate of 20%, declining as the corruption percentage grew and demonstrating no discernable change in accuracy as the dataset size increased. FIGURE 5 depicts the model-detected faulty data.

4) ISOLATION FOREST

An Isolation Tree is a Random Forest variant that may be utilised for anomaly identification. They extract one random characteristic at a time and divide it into homogenous partitions. However, the goal of Isolation Forest [82], [83], [84] is not to create homogeneous partitions, but rather to create partitions in which each datapoint is isolated (That particular isolation contains only the datapoint). The rationale underlying Isolation Trees is that a regular point is more difficult to isolate than an aberrant one.

During the training phase of this approach, we take a sample of the data and generate an itree till each point is visited. Choose a feature at random and split it along at random. The forecast is then completed by calculating the Anomaly Scores as given in Eq. (5) for the new points [85].

$$S(x1, n1) = 2^{\frac{E(p(x1))}{c(n1)}} \tag{5}$$

- x1: data point
- n1: sample size
- PS(x1,n1): Prediction Score
- E(p(x1)): iTrees average search heights for x
- c(n1): Average value of p(x)

The samples that travelled farther into the tree are less likely to be abnormal since it requires more cuttings to separate them. Shorter branches are likely to include anomalies since the tree finds it easier to identify them from other data [86].

$$\text{If } E(p(x1) \ll c(n1) => PS(x1, n1) = 1 => \text{ Anomaly}$$
$$\text{If } E(p(x1) == c(n1) => PS(x1, n1) = 0.5 => \text{ Regular}$$

As a result, Isolation Forest produces a score bound in the range of 0 to 1, where values close to 1 are regarded Anomalous and values less than 0.5 are considered Regular. However, the values produced by sklearn [86] have an inverted interpretation, i.e. numbers less than −0.5 are more regular while values more than −0.5 are more likely to be anomalous. In the present work and on experimentation, the Isolation Forest achieved 82.37% on the clustering dataset and 82.2% with a corruption rate of 20%, with accuracy decreasing as the corruption percentage rose and showing no discernible change as the dataset size increased. The model-detected erroneous data is depicted in FIGURE 4(c).

5) DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE(DBSCAN)

The abbreviation DBSCAN refers to Density-Based Spatial Clustering of Applications with Noise [87]. It is an unsupervised technique that divides a set of points into sets with comparable qualities. It uses density-based clustering to find outliers that do not fit into any of the clusters or sets [88], [89]. FIGURE 4(d) Shows the resulting corrupt data detected using the model.

DBSCAN takes in 2 input parameters-

- ε
- minpts() [90], [91]

where ε represents the radius of the circle formed with data object as centre and minpts() represents the number of points inside the circle.

As a result, three types of datapoints are obtained

(i) Core point - Satisfies the input requirements.

(ii) Boundary point - the core point's neighbour.

(iii) Noise point - Neither centre nor border.

The DBSCAN starts by determining the surroundings starting from an unexplored, random starting point. If the point has enough the neighbours the clustering begins and is labelled as visited or else the point is labelled as noise. This procedure is repeated until all points in a cluster have been realised and all points have been marked visited.

In the present work and on experimentation, the DBSCAN obtained 39.60% accuracy on the clustering dataset and 43.20% with a 20% corruption rate, with accuracy falling as the corruption percentage grew and displaying no noticeable change as the dataset size increased. FIGURE 4(d) depicts the model-detected incorrect data.

6) ELLIPTICAL ENVELOPE

The basis of the Elliptical Envelope algorithm is to create a hypothetical oval shape similar to an ellipse around the given dataset values [92]. The points which fall inside the elliptical shape are regarded as normal data and the values in the dataset outside the elliptical shapes are considered outliers or anomalies. This unsupervised algorithm is mostly used on a gaussian distributed dataset. To find out the data points which are at a further distance from the boundary of the shape minimum-covariance matrix is found. FIGURE 4(e) Shows the resulting corrupt data detected using the model.

In essence, the Elliptical Envelope algorithm fits a Gaussian onto the data and then tries to find the outliers which are the data points which do not fit adequately. Since this is primarily intended to be used for the outlier detection job, our aim is to fetch a reliable estimation of the mean and covariance matrix that will allow us to accept certain outliers in the training dataset while still attempting to recover the true covariance matrix. The Mahobalies distance d_{MH} is used to obtain the distance measure between an instance 'P' and a given allocation denoted by 'D'. It is computed with respect to all the multidimensional data vector x, and the resultant distances(d_{MH}) are sorted in ascending order. The d_{MH} calculated is then used to define a threshold in order to define a boundary which would classify the data points as normal or anomalous. Mahalanobis defined "Mahalanobis distance" [93], [94] as shown in Eq. (6).

$$d_{MH} = \sqrt{(y-\mu)^T (C)^{-1} (y-\mu)} \tag{6}$$

where C denotes the covariance matrix. When the covariance is equal to the identity matrix, d_{MH} simplifies to Euclidean distance [95] and if a covariance matrix is a diagonal matrix, to the normalized Euclidean distance [96].

In the present work and on experimentation, the Elliptic Envelope outlier detection algorithm has performed significantly well for datasets of varying sizes and levels of corruption. It delivered an accuracy of about 86% for datasets of smaller size but 60% corruption rate. It kept up its consistency in the identification of corruption for medium and big datasets as well, maintaining an accuracy of about 80% and 85%, respectively, for the datasets injected with 60% corruption rate.

7) ROTATION-BASED OUTLIER DETECTION

'Rotation Based Outlier Detection' or ROD is an approach that can be used for anomalies and outliers' detection in multivariate data. ROD was first designed to deal with scenarios such as complicated outliers concealed in subspaces [97], [98], taking into account data generated by disparate means [99], and smoothing the detection of outliers in higher dimensions that would otherwise go unnoticed [100]. The robust approach known as "rotation-based outlier detection" rotates the three-dimensional (3D) vectors that represent the data points twice counterclockwise around the geometric median. This rotation is done in accordance with the Rodrigues rotation formula [101]. The rotation produces parallelepipeds, the volumes of which are investigated using mathematical means such as cost functions and utilized to determine the median absolute deviations and generate the outlier score. When the original data space is divided into 3D subspaces, the total score is determined by averaging the 3D-subspace scores for high dimensions. FIGURE 4(f) Shows the resulting corrupt data detected using the model. The algorithm performed fairly well for datasets of varying sizes having lower level of corruption. As per the tests run by the authors ROD provided an accuracy around 62.71% in detecting corruption for clustering data and 62.83% for linear data with 20% corruption, however the accuracy drastically declined as the level of corruption was increased keeping the dataset size fixed.

8) PRINCIPAL COMPONENT ANALYSIS

Principal component analysis or PCA is a traditional mathematical approach where the data matrix is split into principal components. The principal components' poor interpretability and the trade-off between losing crucial information/data and reducing dimensionality, which has a powerful impact on the accuracy, are the main reasons why the method is significantly less effective than other approaches when applied to the synthetic datasets used in this study. The disadvantages of this technique are exacerbated by the requirement to provide the dataset's contamination rate in order to identify outliers. The key elements can be used in multiple situations. As demonstrated in this work, "Principal Component Analysis" (PCA), which is frequently utilized for exploratory analysis and dimension reduction, can also be used to detect corrupted data. FIGURE 4(g) Displays the model's detection of corrupt data.

Principal Component Analysis is primarily used to decrease the dimensionality for the dataset which consists of numerous variables that are correlated, whilst simultaneously retaining the variation and essence of the original dataset. This feat is reached by converting into a new set of variables that are principal components which are not much correlated as well as ordered such that the first few hold the majority of the variation with respect to original variables [102]. FIGURE 6 adequate puts forth the basic steps in order to use PCA for dimension reduction [103], [104], [105], [106]. Anomaly detection using PCA is based on the decomposition of data metrics, However, anomaly detection using this technique is mostly restricted to numerical data which is also one of the drawbacks of this methodology. The various tests conducted as a part of this research to measure the effectiveness of PCA for detection of data corruption revealed some astonishing results. As the level of corruption was increased from 20% to 60% for small, medium and large data sized a significant drop in detecting corruption was observed.The accuracy dropped from around 70% for 20% corruption to around 15-20% for 60% corruption across datasets of all sizes. Thus highlighting the algorithms effectiveness in situations where there are highly corrupted datasets.

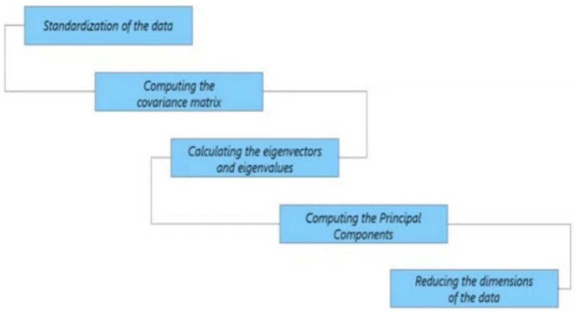

Standardization of the data
Computing the covariance matrix
Calculating the eigenvectors and eigenvalues
Computing the Principal Components
Reducing the dimensions of the data

FIGURE 6. The steps involved in PCA.

9) DEEP SUPPORT VECTOR DATA DESCRIPTION

Deep support vector data description(DeepSVDD) approach proposes a modification of support vector data description model which is another traditional paradigm for anomalies detection. DeepSVDD employs a specific type of neural network to learn appropriate data representations. To distinguish between regular and anomalous data, DeepSVDD [107] employs the hyper-sphere rather than the hyper-plane. DeepSVDD extracts discriminative features from the initial data using a neural network.

$$\min \frac{1}{n}\sum_{i=1}^{n} ||\phi(x_i;\, W){-}a||^2 + \frac{\lambda}{2}\sum_{b=1}^{L}(||W^b||_F)^2 \tag{7}$$

In Eq. (7), "a" represents the center of the sphere, x represents features extracted and W being the weights of the hidden layers and subscript F represents Frobenius norm which cycles through all the entries, adds their squares and then takes the square root as represented in Eq. (8).

$$||A||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}|aij|^2} \tag{8}$$

The first part of Eq. (8) [108] reflects the loss which varies with the distance to the sphere's centre. The next term denotes a W decay regularizer with > 0 inserted as a hyperparameter. The Python PyOD library's Deep SVDD, which was employed to test this model, calls for the specification of the contamination or corruption rate. The default hyper-parameters offered by the PyOD module serve as the foundation for the results that the deepSVDD generates on the aforementioned datasets which are trained for 100 epochs.

DeepSVDD is used to train a neural network [109], [110], which reduces the size of the hypersphere that surrounds the data network representations, driving the network to identify recurring sources of variation. DeepSVDD, like PCA, may be used to discover outlier items in data by calculating their distance from the center. FIGURE 4(h) depicts the model-detected faulty data. DeepSVDD when tested on detection of corruption did not show much deviation in the accuracies for different size and rate of corruption for the dataset. In the present work and on experimentation, it gave an accuracy of 70-75% for 20% corruption of small, medium and large datasets which only dropped slightly to around 50-55% for 40% and 60% corruption for small and medium sized datasets. Whereas it maintained a consistent accuracy of around 70% for large datasets with varying corruption rates.

10) LUNAR

LUNAR stands for Learnable Unified Neighbourhood-based Anomaly Ranking [60], it develops a trainable method for using data from each node's closest neighbours to detect anomalies. Along with the PAACDA algorithm, LUNAR is another graph-based outlier detection technique that offers comparable comparison points.

The LUNAR is unified using K-NN, DBSCAN, LOF and GNNs to provide a faster computing speed and better performance. The outlier score for the KNN is given by Eq. (9). Where y is the data sample and n is the number of neighbours.

$$KNN(y_i) = dist(y_i, y_i^2) \tag{9}$$

The outlier score for the LOF factor is given by the Eq. (10) where lrd is "local reachability density".

$$LOF(y_k) = \frac{\sum_{i \in D} lrd_j(y_i)}{|D_k| lrd_j(y_i)} \tag{10}$$

The edges in the graph component are computed using the Eq. (11) which is the Euclidean distance of the 2 points, where y_i and y_j are the two data samples and N is the number of neighbours.

$$e_{i,j} = dist(y_i, y_j ifj \in N_j), 0 \quad otherwise \tag{11}$$

The K closest neighbours of a sequence of data points are produced as input to the neural network. Finding the k nearest neighbours is one of LUNAR's limitations, as it is with all local outlier approaches. This is mostly a problem in extremely high-dimensional spaces, but because the aforementioned datasets have smaller dimensions, the LUNAR model outperforms conventional probabilistic models in these situations. Other outlier detectors like the local outlier factor (LOF) and DBSCAN have been compared to LUNAR. However, we will focus on the detection of damaged data in tainted datasets in this paper rather than just outliers. In the present work and on experimentation, LUNAR models seems to have performed fairly well in small, medium and large datasets for 20% corruption rate and thereby provided an accuracy of around 85-87%. However the accuracy dropped to about 70% for 40% corruption rate and subsequently to around 50% when the level of corruption was increased to 60% in the datasets. FIGURE 4(i) Shows the resulting corrupt data detected using the model.

11) EMPIRICAL CUMULATIVE OUTLIER DETECTION

It is important to note that for outlier detection with the Empirical Cumulative distribution function, there is a class called ECOD (Empirical Cumulative Outlier Detection) [43]. ECOD is a highly interpretable approach for outlier detection and requires no parameters. The synthetically generated datasets have a rather high sensitivity metric when this method is applied to them. Here the investigators formulated that one of the algorithm's important characteristics is that there are no hyper-parameters, which makes it simpler to implement [111]. However, the function in the Python PyOD package also requires the definition of the corruption percentage, just like other statistical approaches. The authors proposed that ECOD [43] was easily interpretable by looking at the left or right tailed probability which was highly optimised as both of them contributed to the total outlier score. Thus the importance of the tailed probabilities was illustrated in the work we drew inspiration from and thus gained crucial results. In the present work and on experimentation, the optimized ECOD led to a performance of 82.71% on the clustering dataset and 82.30% for linear data with a corruption rate of 20%, decreasing with increase in corruption percentage and indistinguishable change in accuracy when size of the dataset increased.

An overview of the methodology, where data is less likely (low-density) and hence more likely to be corrupted, the ECOD employs information about the distribution of the data. For each variable in the data, ECOD individually estimates an Empirical Cumulative Distribution Function (ECDF) [112], [113]. ECOD uses a univariate ECDF to determine tail probabilities for each variable and then combines them together to produce a score for an observation. The computation takes into account both the left and right tails of each dimension and is performed in log space. Although this method is designed to find outliers, we'll use it to find instances of faulty data in a dataset [114]. FIGURE 4(j) Shows the resulting corrupt data detected using the model. The outlier score for the ECOD algorithm is calculated using the below-mentioned formulae. The Outlier scores are calculated for the left tail, right tail and another measure called auto as shown in Eq. (12), Eq. (13), Eq. (14).

$$O_{left-only}(Y_k) := -\log \widehat{D_{left}(y_k)} = -\textstyle\sum_{i=1}^{a} \log(\hat{D_{left}}^{(i)}(Y_k^{(i)})) \tag{12}$$

$$O_{right-only}(Y_k) := -\log \widehat{D_{right}(y_k)} = -\textstyle\sum_{i=1}^{a} \log(\hat{D_{right}}^{(i)}(Y_k^{(i)})) \tag{13}$$

$$O_{auto}(Y_k) = -\textstyle\sum_{i=1}^{a} [\gamma_i < 0 \log(\hat{D_{left}}^{(i)}(Y_k^{(i)})) + \gamma_i < 0 \log(\hat{D_{right}}^{(i)}(Y_k^{(i)}))] \tag{14}$$

where O is the outlier scores, Y is the input data, D is the corresponding ECDF. The final outlier score is derived by aggregating the above-mentioned outlier scores using the formula mentioned in Eq. (15).

$$O_i = \max\left(O_{left-only}(Y_i), O_{right-only}(Y_i), O_{auto}(Y_i)\right) \tag{15}$$

12) GAUSSIAN MIXTURE MODELS

GMMs or Gaussian mixture models use sets of parameterised probabilistic functions as the weighted components of the pre-trained model using expectation maximisation technique [115]. Previous work on outlier detection has shown that gaussian mixture models have proved effective in detecting outliers providing optimised performance at k=2 gaussians and other parameters being randomly initialised [116]. In the present work and on experimentation, the optimized ECOD led to a performance of 82.71% on the clustering dataset and 82.83% on linear data with a corruption rate of 20%, decreasing with increase in corruption percentage and indistinguishable change in accuracy when size of the dataset increased.

GMMs when used on the aforementioned datasets, it has a very high recall but a very low accuracy. In terms of outlier detection it performs better than other similar clustering techniques like K-Means clustering and DBSCAN clustering. It necessitates the specification of the contamination rate, just like other probabilistic models. This unsupervised clustering method is the Gaussian Mixture Model and can be described using Eq. (16). As a sum of component densities for a particular point. In contrast to K-Means, we fit 'k' Gaussians in the data in this technique. The parameters such as the mean and the variance for each of the cluster as well as the weight of the cluster also called the distribution parameters are then determined [116]. We then determine the odds that each data point will belong to each of the clusters. FIGURE 4(k) Shows the resulting corrupt data detected using the model. The unsupervised method is applied on univariate data and produces an outlier score which then becomes the threshold and the primary criteria to filter the outliers and anomalies from the data [117]. The outlier score for a particular data instance is calculated by the Eq. (17), where p(y) is the PDF or the probability density function and f would be any constant to scale the outlier score and u is the mean [118]. Further the outlier score is normalised to an interval of 0 to 10 in order to ease the comparison.

$$p(y|\lambda) = \sum_{k=1}^{N} v_k g(y|\mu_k, \sum_k) \tag{16}$$

$$OS_y = (\log(p(y)))^{2f} \tag{17}$$

GMMs can also be used to identify potential anomalies in multidimensional datasets. It can also be aggregated with an LSTM (long short term memory) to examine the correlation between the multivariate parameterised data in order to obtain improved outcome in detecting potential outliers [119].

13) MEDIAN ABSOLUTE DEVIATIONS

The MAD or "median absolute deviation" for a set of attributes is the median of that dataset's absolute deviation. This concept is the crux of the Median Absolute Deviation also called MAD algorithm [120]. The absolute deviation of an instant is the pairwise displacement between such a tuple and the distribution's mean. In previous works the MAD is a reliable indicator of the variability in a sample of numeric data, is used in statistics. Because it is so successful and efficient, the Median Absolute Deviation model is frequently employed for this kind of anomaly identification [121]. Instead of pure outliers and anomalies, we will concentrate on corrupt data and provide optimised results [121]. The primary reason this algorithm does well on the aforementioned datasets is that MAD is primarily used for evaluating the distance between data instance and its median in the terms of the median distance and is strictly for univariate data [122]. The time required to compute the MAD score is fairly less and the MAD algorithm is aimed towards symmetric distributions [123]. Unlike the other probabilistic models, the Median Absolute Deviation does not require the corruption rate to be specified which adds to its credibility and provides optimised results [123]. FIGURE 4(l) Shows the resulting corrupt data detected using the model.

The MAD is an alternative to the earlier used threshold equal to the sum of the mean of a distribution and three standard deviations which causes problems as the mean value and the standard deviations are extremely sensitive to outliers. This model is another threshold based outlier detection technique to identify outliers based on the statistical formulae like mean, median, mode. Unfortunately though this algorithm like many other statistical algorithms adds a bias to multiple statistical measures used in outlier detection algorithms which lead to inaccurate results [124].

In the present work and on experimentation, the optimised MAD successfully accomplished an effectiveness of 94.46% on the clustering dataset and 94.77% on linear data with a corruption rate of 20%, decreasing with increasing corruption percentage and indistinguishable change in accuracy as dataset size increased.

14) COPULA-BASED OUTLIER DETECTION

The Python PyOD module contains a set of probabilistic algorithms one of them named COPOD [57], one of which is the Copula-Based Outlier Detector. COPOD is an empirical copula model-based, parameter-free, and highly interpretable outlier detection algorithm. It is important to note that achieving top optimised performance on anomaly detection datasets, interpretable and straightforward corruption visualisation, speed and computational efficiency and scaling to high-dimensional datasets are some of its key distinguishing characteristics [125]. The corruption rate is the only known parameter we use in this investigation. FIGURE 4(m) Shows the resulting corrupt data detected using the model. In previous work the authors have highlighted that the optimised COPOD is highly correlated with the ECOD algorithm [57] and that it outperforms all its variants by being deterministic without any hyper parameters and highly effective for high dimensional datasets.

In statistics and probability, copula is a CDF and a multivariate function where the marginal probability distributions of the variables is uniform in the range [0,1]. Copulas are also used for representing or modeling the dependence of random variables [126]. This is the main motivation behind using copulas to detect anomalies in various application [127].

The COPOD algorithm has three main steps to detect outliers [57]:

- Calculate the cumulative distributive function based on the dataset.
- Calculate the empirical copula.
- Find the tail-probabilities [128] using the above mentioned empirical copula.
- The outlier score is found using the max of the tail probabilities calculated for each data instance.

COPOD (Copula-Based Outlier Detector) does not require pairwise distance measurement, in contrast to other proximity-based algorithms. It mainly applies to multivariate data. To determine and identify the anomalies contained in the dataset, the COPOD generates an empirical copula before calculating the tail-based probability for each data occurrence [129].

In the present work and on experimentation, the optimised COPOD achieved a clustering dataset effectiveness of 92.43% with a corruption rate of 20%, decreasing to 88% with increasing corruption percentage and indistinguishable change in accuracy as dataset size increased. The proposed linear data algorithm had a 92.67% accuracy for a dataset with 20% corrupted data, with comparable trends when the corruption percentage and dataset size were varied.

15) HISTOGRAM-BASED OUTLIER DETECTION

A potent method is called unsupervised histogram-based outlier identification also called HBOS. It establishes the level of corruption while assuming feature independence by producing histograms. The focus of the authors' work will be on the application of histogram-based outlier detection (HBOS), a statistical model that is primarily for outliers, to identify corrupted data in datasets that have been tainted. The histogram algorithm assesses the level of anomalies while assuming feature independence by producing histograms [14]. After multivariate anomaly detection, a histogram for each feature can be generated, graded separately, and aggregated [130]. It is mostly relevant to multivariate data, although outperforming many other probabilistic models when applied with the aforementioned bespoke dataset. Similar to other probabilistic and statistical models, the corruption rate must be given for this histogram based algorithm for outlier detection. FIGURE 4(n) Shows the resulting corrupt data detected using the model.

The histogram algorithm for numerical data is essentially based on two approaches: the first uses the renowned histogram bins with static bin-width that do not vary, and the second uses the bin-width that changes approach (dynamic bin-width). These bins with a wider interval of values or range have lesser density and less height. As a result, the density of each bin is represented by its height, which is then normalised to guarantee that the anomaly is given the same weight and score. The following step involves applying the Eq. (18) to calculate the HBOS value [131].

$$HBOS(q) = \sum_{j=0}^{a} \log \frac{1}{hist_j(q)} \tag{18}$$

where HBOS for instance q with dimension a is calculated by use of height of the bins where q is located. HBOS works well on tasks involving global anomaly identification, but it is unable to identify local outliers since it is unable to model or depict histograms with local anomaly density. The algorithm works well on univariate data [132]. A similar anomaly detection approach can be used there, even though histograms for multidimensional data are computationally intensive and need a large number of operations [133], [134].

In the present work and on experimentation, the optimised HBOS achieved a clustering dataset effectiveness of 95.05% with a corruption rate of 20%, decreasing to 88% with increasing corruption percentage and indistinguishable change in accuracy as dataset size increased. For a dataset with 20% corrupted data, the proposed linear data algorithm had an accuracy of 95%, with comparable trends when the corruption percentage and dataset size were varied.

IV. PROPOSED METHODOLOGY

PAACDA (Proximity based Adamic Adar Corruption Detection Algorithm): Adamic Adar [135], [136] is a graph algorithm used to link nodes in a social network. In this study, we utilise the concept behind this algorithm to detect outliers and missing and modified values while leveraging its prominence in data correlation in graph networks by applying PAACDA to a numerical, tabular dataset. This algorithm is used to compute the accuracy of a particular data instance within a dataset as a whole. The Adamic Adar Index [65] is calculated using Eq. (19).

$$A(x, y) = \sum_{n \in D(x) \cap D(y)} \frac{1}{\log |Deg(n)|} \tag{19}$$

The formula in Eq. (19) presented determines the Adamic Adar index for each node in a network where D(x) represents the neighbors of x and D(y) represents the neighbors of y and Deg(n) is the degree of the common neighbours. Typically, the Adamic Adar Algorithm is utilized to evaluate the closeness of two nodes in a graph. It is based on the notion that values with greater discrepancies between them and values with less in are less likely to be regarded as important than common. The parameter of a value's network closeness is oppositely related [71] to the Adamic-Adar index. Since we are dealing with numerical data, we use the data's mean as a metric to verify the link with each data point and spot the altered or distorted values. The proposed Eq. (20) is used in the PAACDA Algorithms for data corruption detection.

$$PAACDA\ Index = \textstyle\sum_{x=1}^{N} \frac{1}{\log |(Numberofvalues\ in\ x{*}range)} \tag{20}$$

The steps listed below illustrate the method followed:

- The mean is calculated for the column being analysed.

- The range is set as mean/4.
- Each data instance is iterated and Eq. (20) is applied, where x is each data instance.
- If the instance is missing then the PAACDA Index value is set to infinity.
- The PAACDA Index is compared amongst each other and the set of corrupted values is determined.
- The accuracy metric is obtained and confusion matrix is determined.

The reason why defining the range as mean/4 explains intuitively is that the mean accounts for all the components of the provided data and contains information from each observation in a dataset. The mean serves as a link between the actual values and the corrupted data in this manner. Using probe and analysis, we come to the conclusion that mean/4 is the appropriate range for the following method. Again, the algorithm's iteration count is determined experimentally and we utilise 3 iterations for this algorithm and dataset as shown in Algorithm 1. The proposed algorithm loops through twice. The outer loop iterates through each datapoint and the inner loop further iterates and compares each cell data with the one held by outer loop. The time complexity is quadratic. Thus this algorithm runs in Θ(n*n) time, where n is number of data entries in a column.

The PAACDA algorithm is based on the notion that values with fewer similarities [71] are more likely to be viewed as significant than values with greater differences. The performance of this algorithm might vary from dataset to dataset depending on the data distribution as well as the percentage and level of corruption it is subjected to. The PAACDA algorithm uses mean to compute the index which acts as an advantage because each time the data is corrupted the mean varies accordingly. The PAACDA has proven to be more effective than the other clustering and statistical techniques at locating outliers, missing values and corrupted data where accuracy is the primary concern. This makes PAACDA the most suitable for data corruption detection for numerical datasets.

Algorithm 1 Proximity Based Adamic Adar Corruption Detection Algorithm(PAACDA)

```
$indices \leftarrow []$
for each $i$ ∈Corrupted Column do
    $first \leftarrow 1$
    $second \leftarrow 1$
    $third \leftarrow 1$
    for each $j$ ∈Corrupted Column do
        if $abs(j - i) \leq range$ then
            $first \leftarrow first + 1$
        end if
        if $abs(j - i) \leq 2 * range$ then
            $second \leftarrow second + 1$
        end if
        if $abs(j - i) \leq 3 * range$ then
            $third \leftarrow third + 1$
        end if
    end for
    $index \leftarrow 0$
    if $first \neq 1$ then
        $index \leftarrow \frac{1}{\log(first)}$
    end if
    if $second \neq 1$ then
        $index \leftarrow index + \frac{1}{\log(second)}$
    end if
    if $third \neq 1$ then
        $index \leftarrow index + \frac{1}{\log(third)}$
    end if
    indices.append(index)
end for
```

After extensive experiments, the following conclusions were drawn. Table 3 shows the accuracy values of the top performing models. The rest of the results can be found in the appendix.

V. RESULTS AND DISCUSSION

On the various sizes and corruption variations of the synthetically created dataset, several experiments were attempted and put to use. The study primarily focuses on 2 types of datasets: Clustered and Linear

Each of the aforementioned datasets was subsequently examined in various sizes, including,

- Small Dataset - 10000 values
- Medium Dataset - 40000 values
- Medium-Large Dataset - 75000 values

Again, different corruption levels were explored each category of various dataset sizes:

- 20%
- 40%
- 60%

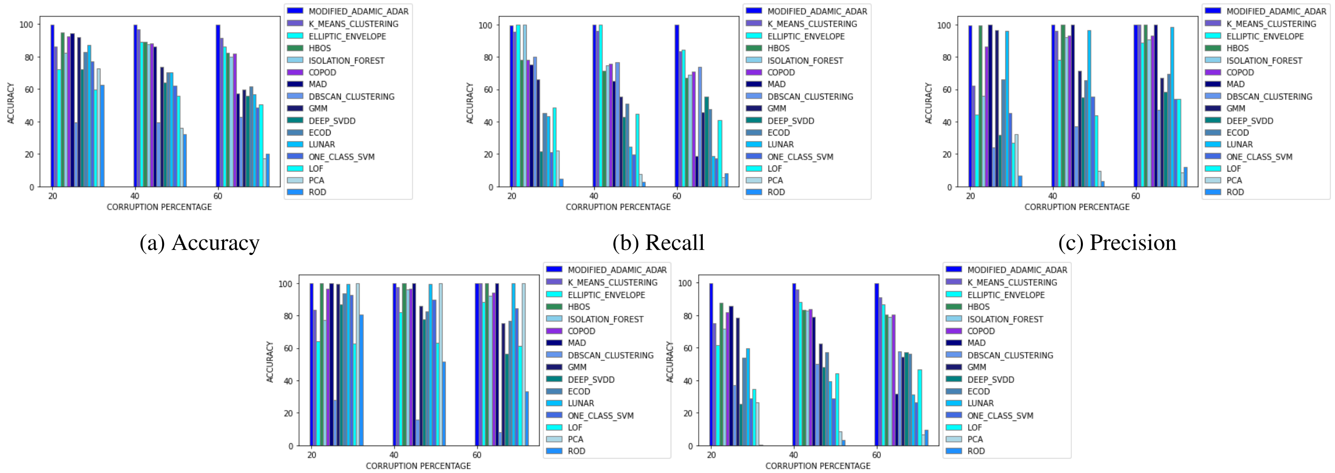
A. RESULTS FOR CLUSTERED DATA

PAACDA, HBOS, MAD perform best in this situation, with accuracy values of 99.74%, 95.05%, and 94.46%, respectively. The middling performers include COPOD, GMM, LUNAR, Elliptic Envelop, K-Means clustering, ECOD and Isolation Forest with accuracy values of 92.43%, 91.95%, 87.01%, 72.17%,86.06%, 82.71% and 82.37% respectively. The One-Class SVM, DeepSVDD, PCA, ROD, LOF and DBSCAN with accuracies of 76.82%, 72.25%, 72.53%, 62.71%, 59.47% and 39.60% respectively generally performed the worst, while the results varied depending on the degree of corruption and the size of the sample.

There were several noticeable variations as the corruption rate went from 20 to 60, including some models that did well in smaller sizes but did poorly as the size expanded. But PAACDA consistently demonstrated its superiority over the competition with unwavering accuracy.

TABLE 3. Accuracy values of the top performing algorithms.

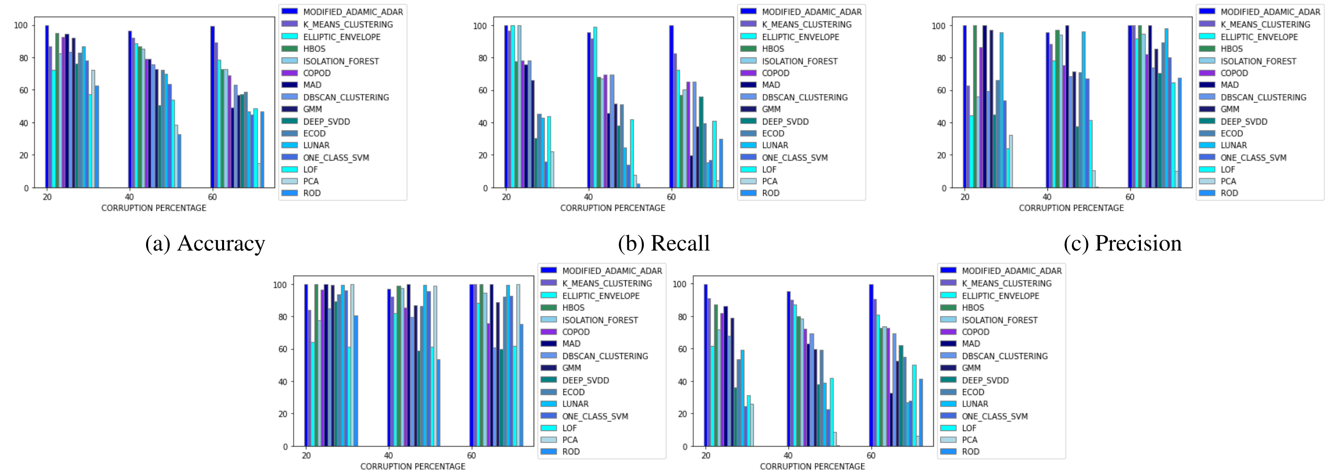
CLUSTERING DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.74	99.82	99.72	99.91	96.35	99.33	99.90	99.77	99.57
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	95.05	88.85	82.56	94.99	86.57	72.75	94.32	89.44	83.71
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	94.46	86.36	57.22	94.58	78.80	49.30	94.18	85.63	54.80
LINEAR DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.94	99.92	99.86	99.98	99.71	99.71	99.94	99.04	95.34
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	95.00	88.74	84.80	94.86	88.21	83.32	93.19	88.56	80.81
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	94.77	87.40	65.25	94.87	87.68	60.90	94.53	87.55	59.61



(a) Accuracy (b) Recall (c) Precision

(d) Sensitivity (e) F1-Score

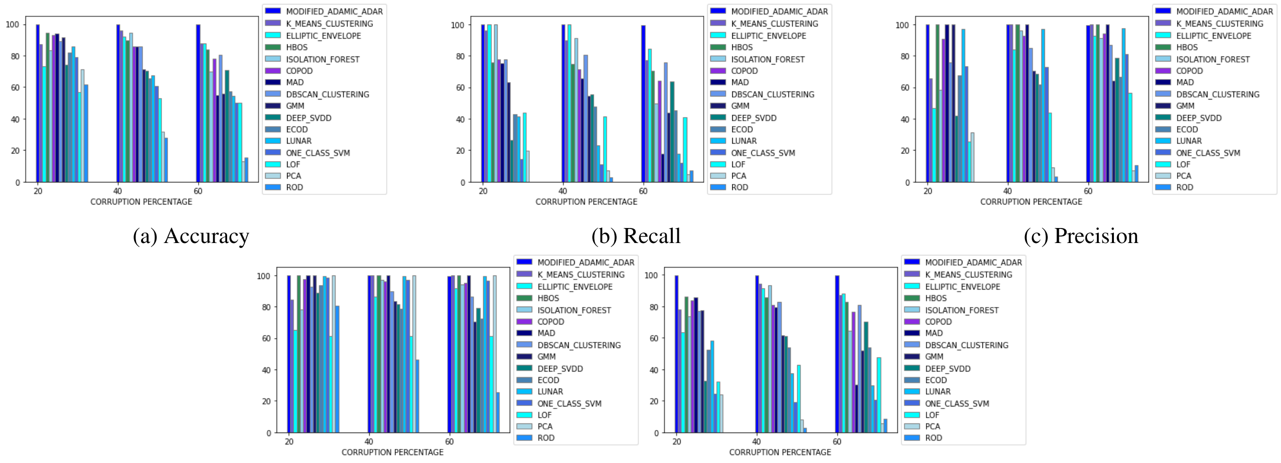
FIGURE 7. Depicts results for the clustering data for small dataset and corruption rates 20%, 40% and 60%.



(a) Accuracy (b) Recall (c) Precision

(d) Sensitivity (e) F1-Score

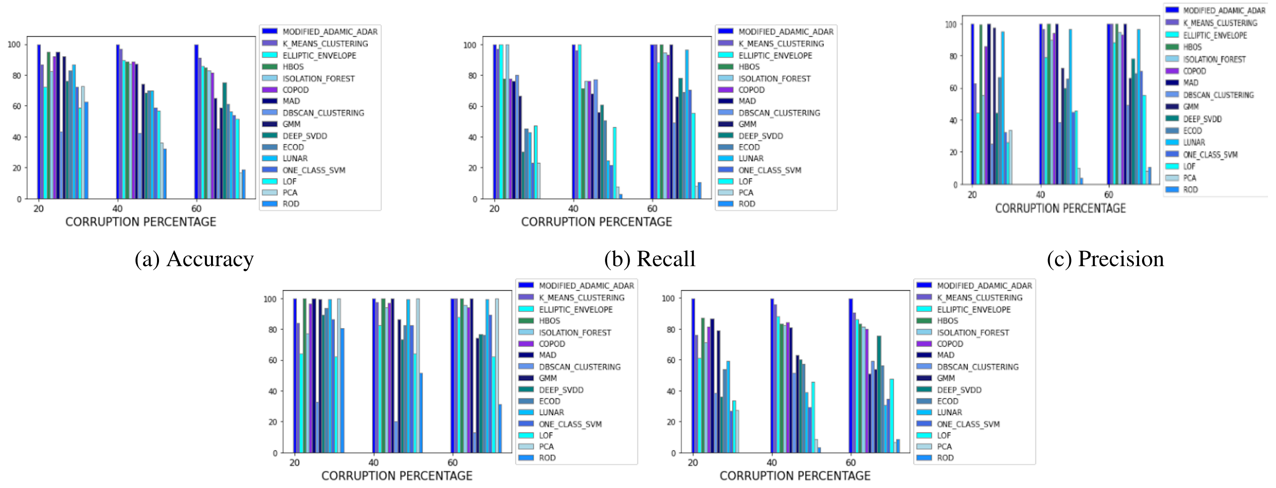
FIGURE 8. Depicts results for the clustering data for medium dataset and corruption rates 20%, 40% and 60%.



(a) Accuracy (b) Recall (c) Precision

(d) Sensitivity (e) F1-Score

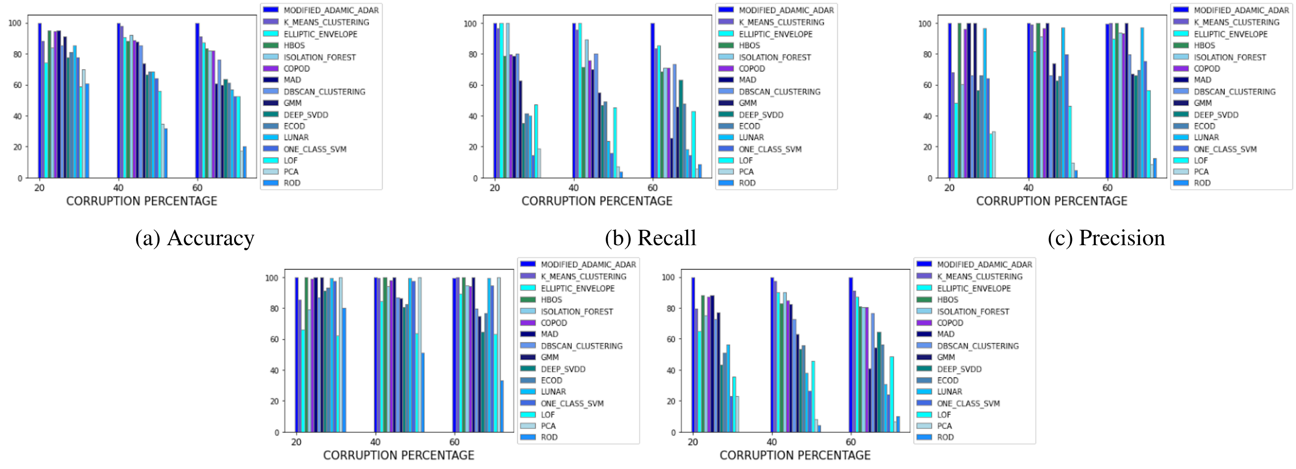
FIGURE 9. **Depicts results for the clustering data for medium-large dataset and for corruption rates 20%, 40% and 60%.**



(a) Accuracy (b) Recall (c) Precision

(d) Sensitivity (e) F1-Score

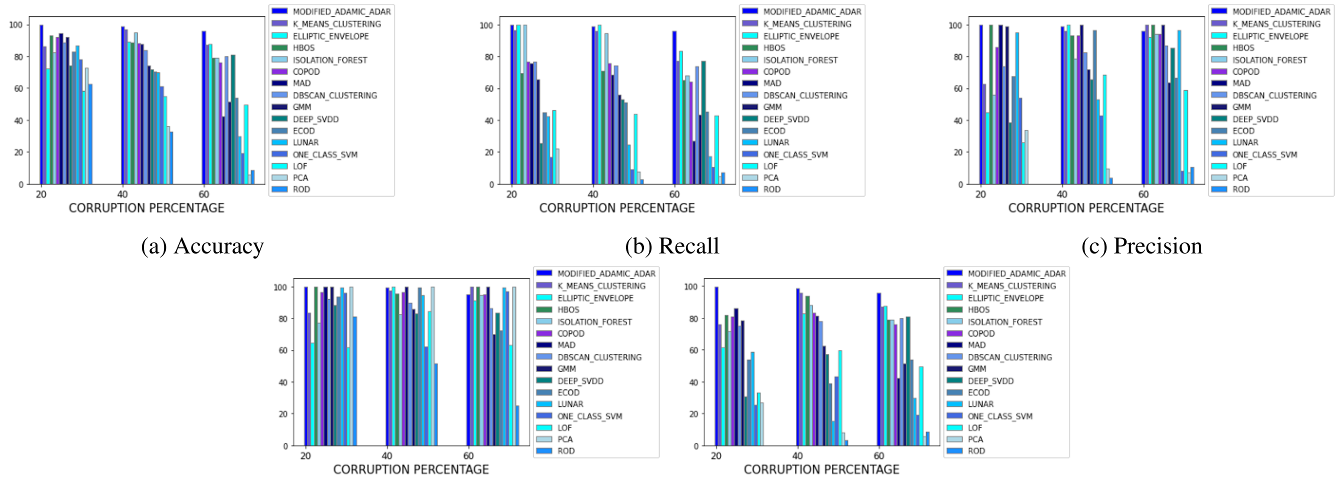
FIGURE 10. **Depicts results for the linear data for small dataset and corruption rates 20%, 40% and 60%.**



(a) Accuracy (b) Recall (c) Precision

(d) Sensitivity (e) F1-Score

FIGURE 11. **Depicts results for the linear data for medium dataset and corruption rates 20%, 40% and 60%.**



(a) Accuracy

(b) Recall

(c) Precision

(d) Sensitivity

(e) F1-Score

FIGURE 12. Depicts results for the linear data for medium-large dataset and for corruption rates 20%, 40% and 60%.

TABLE 4. Results of realistic dataset.

REALISTIC DATA					
Model	Accuracy%	Precision%	Recall%	Sensitivity%	F1 Score%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.75	95.83	95.83	99.87	95.83
K MEANS CLUSTERING	7.74	100	3.01	5.01	5.85
ISOLATION FOREST	98.37	73.91	70.83	99.10	72.34
ELLIPTIC ENVELOPE OUTLIER DETECTION	98.50	78.26	72	99.10	74.99
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	98.12	34.78	100	100	51.61
COPOD (COPULA BASED OUTLIER DETECTION)	99.12	69.56	100	100	82.05
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	98.37	43.47	100	100	60.60
DBSCAN CLUSTERING	21.09	100	3.51	18.76	6.78
GMM (GAUSSIAN MIXTURE MODEL)	99.00	69.56	94.11	99.87	79.99
DeepSVDD	97.62	26.08	75.00	99.74	38.70
ECOD (EMPERICAL CUMMULATIVE OUTLIER DETECTION)	98.00	47.82	73.33	99.48	57.89
LUNAR (UNIFYING LOCAL OUTLIER DETECTION VIA GNN)	94.75	65.21	30.61	95.62	41.66
ONE CLASS SVM (SUPPORT VECTOR MACHINE)	94.25	39.13	21.95	95.88	28.12
LOF (LOCAL OUTLIER FACTOR)	96.37	39.13	37.5	98.07	38.29
PCA (PRINCIPLE COMPONENT ANALYSIS)	97.62	37.78	66.66	100	45.71
ROD (ROTATION BASED OUTLIER DETECTION)	94.00	73.91	28.81	94.60	41.46

Furthermore, the observed pattern shows unequivocally that the PAACDA model's performance declined as the extent of corruption rose as shown in FIGURE 7, 8, 9. Although, in general, the hierarchy of the model's performance was not greatly impacted by the amount of corruption since the same models performed best and worse, the accuracy statistics substantially reduced as the level of corruption rose.

B. RESULTS FOR LINEAR DATA

The same findings as in the previous instance were reached. PAACDA fared better in this instance as well with an accuracy of 99.94%. Most of the higher-performing models from the past including HBOS, MAD, COPOD and GMM did better in this instance with accuracies 95.00%, 94.77%, 92,27% and 92.15% respectively. K-Means clustering, LUNAR, Isolation forest, ECOD and DeepSVDD fared with accuracies of 86.70%, 86.87%, 82.22%, 82.83% and 76.25% respectively. Results were better for models that were more geared toward linear data. Once again, the PCA, One Class SVM, ROD, LOF and DBSCAN Clustering were the worst performers with accuracies of 73.01%, 72.28%, 62.83%, 58.79% and 43.20% respectively. As the dataset size changed, there were no appreciable changes in accuracy.

TABLE 5. Accuracy values for clustering data.

CLUSTERING DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.74	99.82	99.72	99.91	96.35	99.33	99.9	99.77	99.57
K MEANS CLUSTERING	86.06	96.86	91.36	86.58	91.9	88.98	87.2	95.68	87.49
ELLIPTIC ENVELOPE OUTLIER DETECTION	72.17	89.18	86.24	72.18	88.64	78.39	73.34	92	87.56
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	95.05	88.85	82.56	94.99	86.57	72.75	94.32	89.44	83.71
ISOLATION FOREST	82.37	87.62	80.07	82.42	85.5	72.73	83.39	94.58	69.82
COPOD (COPULA BASED OUTLIER DETECTION)	92.43	88.32	81.94	92.43	79.02	68.96	92.98	85.68	78.2
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	94.46	86.36	57.22	94.58	78.8	49.3	94.18	85.63	54.8
DBSCAN CLUSTERING	39.6	39.5	42.7	83.37	75.58	63.34	89	85.89	80.43
GMM (GAUSSIAN MIXTURE MODEL)	91.95	73.89	59.69	92.06	72.91	56.61	91.41	71.3	55.76
ECOD (EMPERICAL CUMMULATIVE OUTLIER DETECTION)	82.71	70.31	61.37	82.69	72.5	58.98	81.84	65.68	57.49
LUNAR (UNIFYING LOCAL OUTLIER DETECTION VIA GNN)	87.01	70.12	57	86.87	69.95	46.49	86.01	67.41	54.55
ONE CLASS SVM (SUPPORT VECTOR MACHINE)	76.82	62.31	48.75	78.26	63.52	44.93	78.81	60.86	50.03
LOF (LOCAL OUTLIER FACTOR)	59.47	55.9	50.7	57.24	53.85	48.6	57.06	52.99	50.31
DeepSVDD	72.25	63.91	55.88	76.17	50.74	57.38	74.22	70.45	70.6
PCA (PRINCIPLE COMPONENT ANALYSIS)	72.53	36.23	17.37	72.42	38.31	14.65	71.07	31.53	12.84
ROD (ROTATION BASED OUTLIER DETECTION)	62.71	32.41	19.91	62.69	32.65	46.61	61.84	27.92	15.3

TABLE 6. Accuracy values for linear data.

LINEAR DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.94	99.92	99.86	99.98	99.71	99.71	99.94	99.04	95.34
K MEANS CLUSTERING	86.7	96.92	91	88.09	97.88	91.25	86.42	96.72	87.33
ISOLATION FOREST	82.22	87.09	82.98	84.12	92.15	82.13	82.45	95.16	79.92
ELLIPTIC ENVELOPE OUTLIER DETECTION	72.11	89.45	85.68	74.08	90.85	87.07	72.35	89.29	87.08
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	95	88.74	84.8	94.86	88.21	83.32	93.19	88.56	80.81
COPOD (COPULA BASED OUTLIER DETECTION)	92.27	88.69	81.63	94.31	88.87	81.88	91.96	88.24	78.03
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	94.77	87.4	65.25	94.87	87.68	60.9	94.53	87.55	59.61
DBSCAN CLUSTERING	43.2	42.61	45.48	85.4	85.4	76.24	88.67	83.65	79.45
GMM (GAUSSIAN MIXTURE MODEL)	92.15	74.3	58.87	91.02	73.59	59.55	92.08	74.03	55.17
DeepSVDD	76.25	68.43	75.16	77.8	66.77	63.62	74.31	72.01	80.17
ECOD (EMPERICAL CUMMULATIVE OUTLIER DETECTION)	82.83	70.02	61.03	80.8	68.67	61.29	82.84	70.22	57.32
LUNAR (UNIFYING LOCAL OUTLIER DETECTION VIA GNN)	86.87	69.85	56.48	85.19	68.55	56.8	86.68	70	54.13
ONE CLASS SVM (SUPPORT VECTOR MACHINE)	72.28	58.58	54.19	77.43	63.93	52.47	78.19	61.09	49.31
LOF (LOCAL OUTLIER FACTOR)	58.79	57.05	51.56	58.68	56.07	52.56	58.42	55.01	51.98
PCA (PRINCIPLE COMPONENT ANALYSIS)	73.01	36.12	17.03	69.86	34.55	17.37	72.79	36.05	12.58
ROD (ROTATION BASED OUTLIER DETECTION)	62.83	32.28	18.57	60.82	31.68	20.12	62.85	32.52	15.2

However, just like in the previous instance, performance suffered as the amount of corruption rose.

Slightly different trends were observed in the case of the other metrics like Precision, Recall, Sensitivity, and F1 score in the case of both Clustered and Linear Datasets. Several models that had previously had moderate and semi-moderate accuracy now had either high precision, recall, or F1 score, despite the ordering hierarchy being relatively constant as these models were geared towards linear data when compared to clustering data as shown in FIGURE 10, 11, 12.

TABLE 7. Recall values for clustering data.

CLUSTERING DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.41	99.77	99.73	99.8	95.34	99.8	99.78	99.73	99.61
K MEANS CLUSTERING	95.39	96.01	83.59	96.32	91.66	82.53	96.21	89.71	77.23
ELLIPTIC ENVELOPE OUTLIER DETECTION	100	100	84.4	100	99.12	72.5	100	99.99	84.19
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	78.03	71.54	66.88	77.45	67.78	56.8	75.69	74.87	70.34
ISOLATION FOREST	100	74.75	69.04	100	67.2	60.09	100	91.13	49.81
COPOD (COPULA BASED OUTLIER DETECTION)	78.03	75.8	71.09	78.02	69.23	65.02	77.8	71.41	64.36
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	75.01	65.18	18.76	75.58	45.95	19.62	75.08	65.8	17.72
DBSCAN CLUSTERING	80.28	76.54	73.77	78.33	69.65	64.99	77.83	80.62	75.79
GMM (GAUSSIAN MIXTURE MODEL)	65.99	55.64	45.89	66.19	51.52	37.76	63.35	54.3	43.93
ECOD (EMPERICAL CUMMULATIVE OUTLIER DETECTION)	45.15	51.07	47.49	45.08	51.01	39.63	42.84	47.61	45.51
LUNAR (UNIFYING LOCAL OUTLIER DETECTION VIA GNN)	43.25	24.62	18.66	42.96	24.43	15.51	41.45	23.12	17.73
ONE CLASS SVM (SUPPORT VECTOR MACHINE)	20.92	19.49	17.52	15.8	13.73	16.88	14.56	10.97	11.75
LOF (LOCAL OUTLIER FACTOR)	48.8	44.76	41.16	43.78	42.15	40.96	43.7	41.65	41.18
DeepSVDD	21.56	42.9	55.58	30.4	38.19	55.84	26.52	55.44	63.81
PCA (PRINCIPLE COMPONENT ANALYSIS)	22.19	7.58	5.71	21.93	7.42	4.49	19.78	6.96	4.86
ROD (ROTATION BASED OUTLIER DETECTION)	4.51	2.7	8.12	0.01	2.1	29.83	0.0005	2.67	7.1

TABLE 8. Recall values for linear data.

LINEAR DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.86	99.89	99.86	99.95	99.95	99.94	99.86	98.78	95.77
K MEANS CLUSTERING	96.96	95.89	82.97	96.3	95.69	83.43	96.41	95.77	77.02
ISOLATION FOREST	100	76.04	71.75	100	89.29	71	100	94.72	68.04
ELLIPTIC ENVELOPE OUTLIER DETECTION	100	100	83.77	100	100	85.23	100	100	83.6
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	77.43	71.45	71.22	78.66	71.14	68.3	69.54	70.88	65.21
COPOD (COPULA BASED OUTLIER DETECTION)	77.74	76.04	70.42	79.71	75.52	70.86	76.75	75.75	64.08
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	76.34	68.06	34.21	78.72	69.86	25.69	75.56	68.32	26.79
DBSCAN CLUSTERING	79.91	77.13	74.42	80.21	80.21	73.4	76.65	74.09	73.82
GMM (GAUSSIAN MIXTURE MODEL)	66.35	56.14	45.3	62.75	54.97	45.87	65.4	55.73	43.36
DeepSVDD	30.39	60.68	73.81	35.29	46.63	62.95	25.66	53.17	77.34
ECOD (EMPERICAL CUMMULATIVE OUTLIER DETECTION)	45.27	50.72	47.34	41.52	48.96	47.52	44.73	50.89	45.31
LUNAR (UNIFYING LOCAL OUTLIER DETECTION VIA GNN)	42.92	24.46	18.26	40.02	23.75	18.45	42.59	24.56	17.49
ONE CLASS SVM (SUPPORT VECTOR MACHINE)	22.97	21.64	22.98	14.17	15.69	14.47	16.77	9.06	10.75
LOF (LOCAL OUTLIER FACTOR)	47.26	46.26	42.01	47.26	45.19	42.93	46.47	43.65	42.73
PCA (PRINCIPLE COMPONENT ANALYSIS)	23.06	7.75	5.69	18.82	7.2	5.78	22.26	7.4	4.76
ROD (ROTATION BASED OUTLIER DETECTION)	0.04	2.88	7.15	0.06	3.68	8.4	0.03	2.92	7.13

C. RESULTS FOR REALISTIC DATA

The same set of experiments with 16 different models were carried out on Realistic existing dataset with outliers. PAACDA performed the best with 99.75% accuracy. Unlike the synthetic data, the close competitors for our proposed model was not HBOS and MAD, but instead COPOD and Elliptical Envelope with accuracy of 99.12% and 98.50% respectively. Every model performed decently well except K means which has accuracy of 7.74% as it is not suitable for all kinds of dataset. Further, for each of the above models other metrics such as precision, recall, sensitivity and F1 score are tabulated in the Table 4.

TABLE 9. Precision values for clustering data.

CLUSTERING DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.41	99.77	99.73	99.8	95.34	99.8	99.78	99.73	99.61
K MEANS CLUSTERING	62.07	95.96	100	62.92	88.2	100	65.34	100	100
ELLIPTIC ENVELOPE OUTLIER DETECTION	44.34	78.36	88.9	44.37	77.97	91.47	46.68	84.01	92.49
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	99.53	100	100	100	97.11	100	100	100	100
ISOLATION FOREST	55.7	92.16	90.92	55.8	94.16	94.77	58.42	95.77	91.3
COPOD (COPULA BASED OUTLIER DETECTION)	86.5	93.1	92.94	86.56	75.3	82.04	90.8	92.84	94.09
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	100	100	100	100	100	100	100	100	100
DBSCAN CLUSTERING	24.1	36.88	47.18	59.52	68.7	73.76	75.74	85.01	86.92
GMM (GAUSSIAN MIXTURE MODEL)	96.63	71.4	67.15	97.09	71.43	85.24	99.77	70.6	64.23
ECOD (EMPERICAL CUMMULATIVE OUTLIER DETECTION)	66.11	65.54	69.49	66.12	70.72	89.47	67.47	61.9	66.53
LUNAR (UNIFYING LOCAL OUTLIER DETECTION VIA GNN)	95.9	96.5	98.3	95.32	95.85	97.85	96.76	97.13	97.44
ONE CLASS SVM (SUPPORT VECTOR MACHINE)	45.09	55.4	54.13	53.44	67.1	80.18	73.14	72.61	81.21
LOF (LOCAL OUTLIER FACTOR)	27.05	43.85	54.2	24.28	41.33	64.6	25.5	43.75	56.55
DeepSVDD	31.57	55.06	58.54	44.58	37.45	70.46	41.77	68.26	78.64
PCA (PRINCIPLE COMPONENT ANALYSIS)	32.49	9.72	8.36	32.17	10.29	10.15	31.15	9.05	7.11
ROD (ROTATION BASED OUTLIER DETECTION)	6.6	3.47	11.89	0.01	0.29	67.34	0.0005	3.47	10.39

TABLE 10. Precision values for linear data.

LINEAR DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.86	99.89	99.86	99.95	99.95	99.5	99.86	98.78	95.77
K MEANS CLUSTERING	62.92	96.28	99.97	67.8	99.08	99.99	62.78	95.87	100
ISOLATION FOREST	55.42	89.65	94.75	60.27	91.31	93.46	56.02	78.59	93.88
ELLIPTIC ENVELOPE OUTLIER DETECTION	44.22	78.9	88.5	48.17	81.7	89.68	44.71	100	92.25
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	99.44	100	100	100	100	100	100	93.09	100
COPOD (COPULA BASED OUTLIER DETECTION)	85.95	94.16	93.11	96.01	96.45	93.05	85.8	93.04	94.27
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	100	100	100	100	100	100	100	100	100
DBSCAN CLUSTERING	25.23	38.61	48.94	66.27	66.27	79.81	73.72	82.52	86.96
GMM (GAUSSIAN MIXTURE MODEL)	97.28	72.5	66.15	100	73.69	66.85	98.78	71.86	63.79
DeepSVDD	44.56	59.85	77.98	56.25	62.51	66.24	38.75	65.62	85.34
ECOD (EMPERICAL CUMMULATIVE OUTLIER DETECTION)	66.37	65.49	69.14	66.17	65.63	69.25	67.56	96.53	66.66
LUNAR (UNIFYING LOCAL OUTLIER DETECTION VIA GNN)	94.9	96.5	96.5	96.4	97.02	97.1	95.22	52.89	96.54
ONE CLASS SVM (SUPPORT VECTOR MACHINE)	32.21	44.82	70.29	64.31	79.82	75.06	53.98	42.89	8.04
LOF (LOCAL OUTLIER FACTOR)	26.12	45.62	55.47	28.46	46.16	56.47	25.97	68.56	58.95
PCA (PRINCIPLE COMPONENT ANALYSIS)	33.81	10.01	8.32	30	9.65	8.43	33.62	9.55	7.01
ROD (ROTATION BASED OUTLIER DETECTION)	0.06	3.73	10.45	0.09	4.93	12.24	0.05	3.77	10.5

D. LIMITATIONS AND CONSTRAINTS

The proposed methodology however has some limitations. The PAACDA requires the specification of the corruption percentage as one of the parameters which further will require parameter tuning. In addition to this PAACDA works well with uni-variate data unlike ROD, GMM and HBOD which can handle multi-variate data at the a time.

TABLE 11. Sensitivity values for clustering data.

CLUSTERING DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.83	99.85	99.7	99.94	96.99	99.94	99.93	99.8	99.52
K MEANS CLUSTERING	83.39	97.4	100	83.8	92.05	100	84.46	100	100
ELLIPTIC ENVELOPE OUTLIER DETECTION	64.24	82.2	88.27	64.25	81.84	88.45	65.22	86.21	91.67
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	99.89	100	100	100	98.7	100	100	100	100
ISOLATION FOREST	77.34	95.9	92.33	77.41	97.31	94.33	78.33	97.08	94.21
COPOD (COPULA BASED OUTLIER DETECTION)	96.53	96.38	94	96.54	85.34	75.67	97.6	96.01	95.08
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	100	100	100	100	100	100	100	100	100
DBSCAN CLUSTERING	28	15.63	8.13	84.8	79.42	60.51	92.4	89.7	86.1
GMM (GAUSSIAN MIXTURE MODEL)	99.34	85.64	75.03	99.43	86.7	88.83	99.95	83.62	70.17
ECOD (EMPERICAL CUMMULATIVE OUTLIER DETECTION)	93.4	82.7	76.8	93.41	86.37	92.03	93.71	78.77	72.1
LUNAR (UNIFYING LOCAL OUTLIER DETECTION VIA GNN)	99.47	99.42	99.64	99.39	99.31	99.41	99.57	99.5	99.43
ONE CLASS SVM (SUPPORT VECTOR MACHINE)	92.74	89.88	84.48	96.07	95.65	92.86	98.37	97	96.68
LOF (LOCAL OUTLIER FACTOR)	62.5	63.07	61.3	61.07	61.39	61.64	61.12	61.2	61.44
DeepSVDD	86.68	77.44	56.21	89.22	58.83	59.99	88.74	81.32	78.87
PCA (PRINCIPLE COMPONENT ANALYSIS)	99.89	100	100	100	98.7	100	100	100	100
ROD (ROTATION BASED OUTLIER DETECTION)	80.56	51.54	33.01	80.56	53.59	75.28	80.66	46.22	25.29

TABLE 12. Sensitivity values for linear data.

LINEAR DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.96	99.93	99.85	99.98	99.98	99.45	99.96	99.21	94.8
K MEANS CLUSTERING	83.78	97.58	99.97	85.49	99.38	99.99	83.54	97.33	100
ISOLATION FOREST	77.17	94.28	95.54	79.08	94.13	94.48	77.39	82.37	94.54
ELLIPTIC ENVELOPE OUTLIER DETECTION	64.19	82.57	87.81	65.86	84.53	89.11	64.39	100	91.35
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	99.98	100	100	100	100	100	100	95.45	100
COPOD (COPULA BASED OUTLIER DETECTION)	96.39	96.92	94.17	98.94	98.08	94.12	96.34	96.33	95.21
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	100	100	100	100	100	100	100	100	100
DBSCAN CLUSTERING	32.77	20.11	13.07	87.04	87.04	79.38	92.13	89.84	86.38
GMM (GAUSSIAN MIXTURE MODEL)	99.47	86.12	74.05	100	86.44	74.74	99.76	85.87	69.7
DeepSVDD	89.26	73.47	76.66	91.28	80.68	64.37	88.32	82.74	83.65
ECOD (EMPERICAL CUMMULATIVE OUTLIER DETECTION)	93.49	82.59	76.34	93.26	82.29	76.57	93.81	99.42	72.1
LUNAR (UNIFYING LOCAL OUTLIER DETECTION VIA GNN)	99.34	99.42	99.25	99.52	99.49	99.38	99.38	94.77	99.22
ONE CLASS SVM (SUPPORT VECTOR MACHINE)	86.27	82.64	89.12	97.5	97.26	94.66	95.88	62.36	96.78
LOF (LOCAL OUTLIER FACTOR)	62.06	64.07	62.25	62.3	63.59	63.26	61.86	84.21	63.37
PCA (PRINCIPLE COMPONENT ANALYSIS)	99.98	100	100	100	100	100	100	100	100
ROD (ROTATION BASED OUTLIER DETECTION)	80.65	51.42	31.34	80.1	51.02	33.14	80.94	51.69	25.12

VI. CONCLUSION AND FUTURE SCOPE

Data that is reliable and accurate is essential for conducting effective research. This is because faulty and untrustworthy data produces erroneous or false results. Inadvertently entering incorrect data into a computer will produce an output, which could be fatal in fields such as healthcare and defence. While being written, edited, or transferred to another drive, data might become corrupted. Additionally, a virus can damage files. Usually, this is done on purpose to harm crucial system files. Finding outliers that are silently existing in a

TABLE 13. F1-Score values for clustering data.

CLUSTERING DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.41	99.77	99.73	99.8	95.34	99.8	99.78	99.73	99.61
K MEANS CLUSTERING	75.21	95.99	91.06	91.06	89.9	90.42	77.83	94.58	87.15
ELLIPTIC ENVELOPE OUTLIER DETECTION	61.43	87.86	86.59	61.47	87.28	80.89	63.65	91.31	88.14
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	87.48	83.41	80.15	87.29	79.84	72.45	86.16	85.63	82.59
ISOLATION FOREST	71.55	82.55	78.84	71.63	78.43	73.55	73.75	93.39	64.45
COPOD (COPULA BASED OUTLIER DETECTION)	82.04	83.56	80.56	82.07	72.13	72.55	83.8	80.73	76.44
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	85.72	78.92	31.59	86.09	62.96	32.81	85.76	79.37	30.11
DBSCAN CLUSTERING	37.08	49.78	57.55	67.64	69.17	69.1	76.77	82.76	80.97
GMM (GAUSSIAN MIXTURE MODEL)	78.42	62.54	54.52	78.72	59.87	52.33	77.49	61.39	52.18
ECOD (EMPERICAL CUMMULATIVE OUTLIER DETECTION)	53.65	57.4	56.42	53.61	59.27	54.93	52.41	53.82	54.05
LUNAR (UNIFYING LOCAL OUTLIER DETECTION VIA GNN)	59.62	39.24	31.37	59.23	38.94	26.77	58.04	37.35	30.01
ONE CLASS SVM (SUPPORT VECTOR MACHINE)	28.58	28.84	26.48	24.4	22.79	27.89	24.29	19.06	20.53
LOF (LOCAL OUTLIER FACTOR)	34.8	44.3	46.79	31.24	41.74	50.13	32.21	42.67	47.65
DeepSVDD	25.62	48.22	57.02	36.15	37.81	62.3	32.45	61.18	70.45
PCA (PRINCIPLE COMPONENT ANALYSIS)	26.37	8.52	6.79	26.08	8.62	6.23	24.19	7.87	5.77
ROD (ROTATION BASED OUTLIER DETECTION)	0.53	3.04	9.65	0.01	0.24	41.34	0.0005	3.02	8.44

dataset is only half the issue; data with high rates of corruption can seriously impair model accuracy and the outcomes of data analytics. In this case, accuracy is a necessary requirement to verify the information from the sources. Research accuracy ensures that the information gathered is accurate or inaccurate. Therefore, it is crucial to check the accuracy of any survey.

In this article, firstly we introduce the primary concepts of outlier detection and how these models and techniques can be used to detect corrupted data. Then, we encapsulate the quality improvement approaches of data corruption detection and split the data to two categories based on their behavior: linearly distributed data and clustered data on 3 high structured synthetic datasets- small, medium, medium-large. The results showed That the PAACDA outperformed the other algorithms with an accuracy of 96.35% for clustered data and 99.04% for linear data. Lastly, we lay-out an experiment-based comparison of multiple cutting edge quality improvement approaches using a plethora of quality evaluation metrics, the authors have combined the findings of various other probabilistic and statistical models and explained how they used the innovative PAACDA algorithm to get desired results on the data. With accuracy values of 95.05% and 94.46%, respectively, HBOS and MAD are among the other top performers for the clustering dataset. COPOD, GMM, LUNAR, Elliptic Envelop, K-Means clustering, ECOD, and Isolation Forest are among the middle performers, with accuracy values of 92.43%, 91.95%, 87.01%, 72.17%, 86.06%, 82.71, and 82.37%, respectively. The One-Class SVM, DeepSVDD, PCA, ROD, LOF, and DBSCAN performed the worst, with accuracies of 76.82%, 72.25%, 72.53%, 62.71%, 59.47%, and 39.60%, respectively. Most previous higher-performing models, including HBOS, MAD, COPOD, and GMM, performed better on the linear dataset, with accuracies of 95.00%, 94.77%, 92.27%, and 92.15%, respectively. Accuracy rates for K-Means clustering, LUNAR, Isolation forest, ECOD, and DeepSVDD were 86.70%, 86.87%, 82.22%, 82.83%, and 76.25%, respectively. Models that were more geared toward linear data produced better results. PCA, One Class SVM, ROD, LOF, and DBSCAN Clustering performed the worst, with accuracies of 73.01%, 72.28%, 62.83%, 58.79%, and 43.20%, respectively. There were no discernible changes in accuracy as the dataset size increased. However, as in the previous case, performance suffered as the level of corruption increased.

To the authors knowledge, the paper is the first one that systematically addressed the detection of corrupted data from different aspects such as data distribution, dataset size and also variations of corruption rates. We reviewed most of the published papers in well reputed libraries. With an accuracy of 96.35% for clustered data and 99.04% for linear data, the PAACDA algorithm exceeds the other models. In this work an exhaustive review of many unsupervised and probabilistic models was conducted. The other top performing algorithms are the Histogram based outlier detection model, K-Means Clustering, Elliptical Envelope outlier detection and Isolation forest.

This study correctly identified the PAACDA algorithm as one of the better methods and provided a comprehensive compilation of numerous alternative approaches for solving

TABLE 14. F1-Score values for linear data.

LINEAR DATA									
	Small (10,000 values)			Medium (40,000 values)			Medium-Large (75,000 values)		
Model	20%	40%	60%	20%	40%	60%	20%	40%	60%
PAACDA (PROXIMITY BASED ADAMIC ADAR CORRUPTION DETECTION ALGORITHM)	99.86	99.89	99.86	99.95	99.95	99.72	99.86	98.78	95.77
K MEANS CLUSTERING	76.32	96.08	90.68	79.58	97.36	90.96	76.04	95.82	87.02
ISOLATION FOREST	71.32	82.29	81.66	75.21	90.29	80.7	71.81	88.01	78.9
ELLIPTIC ENVELOPE OUTLIER DETECTION	61.32	88.2	86.07	65.02	89.93	87.4	61.79	82.96	87.71
HBOS (HISTOGRAM BASED OUTLIER DETECTION)	87.25	83.35	83.19	88.05	83.14	81.16	82.03	93.9	78.94
COPOD (COPULA BASED OUTLIER DETECTION)	81.64	84.13	80.19	87.11	84.73	80.45	81.02	83.51	76.3
MAD (MEDIAN ABSOLUTE DEVIATION) ALGORITHM	86.58	80.99	50.98	88.09	82.25	40.88	86.08	81.17	42.26
DBSCAN CLUSTERING	38.35	51.46	59.05	72.58	72.58	76.47	75.16	78.08	79.85
GMM (GAUSSIAN MIXTURE MODEL)	78.89	63.28	53.78	77.11	62.97	54.41	78.69	62.78	51.63
DeepSVDD	36.13	60.26	75.84	43.37	53.41	64.55	30.87	57.33	81.15
ECOD (EMPERICAL CUMMULATIVE OUTLIER DETECTION)	53.83	57.17	56.2	51.02	56.08	56.36	53.82	39.16	53.95
LUNAR (UNIFYING LOCAL OUTLIER DETECTION VIA GNN)	59.1	39.02	30.72	56.56	38.15	31.01	58.85	15.48	29.62
ONE CLASS SVM (SUPPORT VECTOR MACHINE)	26.82	29.19	34.64	23.23	26.22	24.26	25.59	43.27	18.97
LOF (LOCAL OUTLIER FACTOR)	33.64	45.94	47.81	35.52	45.67	48.78	33.32	59.89	49.55
PCA (PRINCIPLE COMPONENT ANALYSIS)	27.42	8.74	6.76	23.13	8.25	6.86	26.78	8.34	5.67
ROD (ROTATION BASED OUTLIER DETECTION)	0.05	3.25	8.49	0.07	4.21	9.96	0.04	3.29	8.49

the problem of correctly differentiating corrupted data in the dataset. The future work in this field must focus on the following aspects: 1. To find tainted values that stray just a little from the original values, more algorithmic research could be explored. 2. The current effort leans towards outliers and focuses mostly on the detection of tainted data. 3. It is possible to conduct additional studies on the topic of recovering the original data using tools like backpropagation and GANs (Generative Adversarial Networks). When the output feature is known, backpropagation can be used to create the input features. 4. The missing values can be filled in using GANs. Furthermore, traditional restoration models are comparitively complex, which limits our propensity to expand pragmatic studies and applications. The potential of these strategies is genuinely tremendous, and listing them only scratches the surface. This study can be expanded further to include categorical and even picture datasets in addition to numerical data. 5. In the realm of picture collections, GANs have a wide range of uses, particularly for identifying false images and producing Deep fakes.

APPENDIX

The entire tables under the results and discussion section have been included in the Appendix A below.

APPENDIX A TABLES

Additional tables that support the experiment can be found here. All the experimental results including the results for small, medium and large datasets for all corruption rates (20%, 40%, 60%) are included in this document. Table 1,2,3,4,5,6,7,8,9,10 represent the accuracy, recall, precision, sensitivity and F1-score of both linear and clustering data.

REFERENCES

[1] E. Burgdorf, "Predicting the impact of data corruption on the operation of cyber-physical systems," Missouri Univ. Sci. Technol., Rolla, MO, USA, Tech. Rep. 27929030, 2017.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009.

[3] M. Pang-Ning and V. Steinbach, *Introduction to Data Mining*. London, U.K.: Pearson, 2016.

[4] H. M. Touny, A. S. Moussa, and A. S. Hadi, "Fuzzy multivariate outliers with application on BACON algorithm," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2020, pp. 1–7.

[5] S. Thudumu, P. Branch, J. Jin, and J. Singh, "A comprehensive survey of anomaly detection techniques for high dimensional big data," *J. Big Data*, vol. 7, no. 1, pp. 1–30, 2020, doi: 10.1186/s40537-020-00320-x.

[6] O. J. Oyelade, O. O. Oladipupo, and I. C. Obagbuwa, "Application of K means clustering algorithm for prediction of students academic performance," 2010, *arXiv:1002.2425*.

[7] H. L. Sari, D. Suranti, and L. N. Zulita, "Implementation of k-means clustering method for electronic learning model," *J. Phys., Conf. Ser.*, vol. 930, Dec. 2017, Art. no. 012021, doi: 10.1088/1742-6596/930/1/012021.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96, Jan. 1996, pp. 226–231.

[9] D. Deng, "DBSCAN clustering algorithm based on density," in *Proc. 7th Int. Forum Electr. Eng. Autom. (IFEEA)*, Sep. 2020, pp. 949–953.

[10] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.

[11] R. Gao, T. Zhang, S. Sun, and Z. Liu, "Research and improvement of isolation forest in detection of local anomaly points," *J. Phys., Conf. Ser.*, vol. 1237, no. 5, Jun. 2019, Art. no. 052023, doi: 10.1088/1742-6596/1237/5/052023.

[12] M. Ashrafuzzaman, S. Das, A. A. Jillepalli, Y. Chakhchoukh, and F. T. Sheldon, "Elliptic envelope based detection of stealthy false data injection attacks in smart grid control systems," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2020, pp. 1131–1137.

[13] C. McKinnon, J. Carroll, A. McDonald, S. Koukoura, D. Infield, and C. Soraghan, ''Comparison of new anomaly detection technique for wind turbine condition monitoring using gearbox SCADA data,'' *Energies*, vol. 13, no. 19, p. 5152, Oct. 2020, doi: 10.3390/en13195152.

[14] M. Goldstein and A. Dengel, ''Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm,'' in *Proc. KI, Poster Demo Track*, vol. 9, 2012, pp. 59–63.

[15] N. Paulauskas and A. Baskys, ''Application of histogram-based outlier scores to detect computer network anomalies,'' *Electronics*, vol. 8, no. 11, p. 1251, Nov. 2019, doi: 10.3390/electronics8111251.

[16] I. T. Jolliffe and J. Cadima, ''Principal component analysis: A review and recent developments,'' *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 374, no. 2065, Apr. 2016, Art. no. 20150202, doi: 10.1098/rsta.2015.0202.

[17] S. Mishra, U. Sarkar, S. Taraphder, S. Datta, D. Swain, R. Saikhom, S. Panda, and M. Laishram, ''Principal component analysis,'' *Int. J. Livestock Res.*, vol. 2, no. 4, pp. 433–459, 2017, doi: 10.5455/ijlr.20170415115235.

[18] A. Karimian, Z. Yang, and R. Tron, ''Rotational outlier identification in pose graphs using dual decomposition,'' in *Computer Vision ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 391–407.

[19] Y. Almardeny, N. Boujnah, and F. Cleary, ''A novel outlier detection method for multivariate data,'' *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 9, pp. 4052–4062, Sep. 2022, doi: 10.1109/tkde.2020.3036524.

[20] O. Alghushairy, R. Alsini, T. Soule, and X. Ma, ''A review of local outlier factor algorithms for outlier detection in big data streams,'' *Big Data Cognit. Comput.*, vol. 5, no. 1, p. 1, Dec. 2020, doi: 10.3390/bdcc5010001.

[21] M. M. Breunig, R. T. Kriegel, and J. Ng, ''LOF: Identifying density-based local outliers,'' in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104.

[22] L. Ruff, ''Deep one-class classification,'' in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.

[23] Z. Zhang and X. Deng, ''Anomaly detection using improved deep SVDD model with data structure preservation,'' *Pattern Recognit. Lett.*, vol. 148, pp. 1–6, Aug. 2021, doi: 10.1016/j.patrec.2021.04.020.

[24] L. Adamic and E. Adar, ''How to search a social network,'' *Social Netw.*, vol. 27, no. 3, pp. 187–203, 2005, doi: 10.1016/j.socnet.2005.01.007.

[25] F. Gao, K. Musial, C. Cooper, and S. Tsoka, ''Link prediction methods and their accuracy for different social networks and network metrics,'' *Sci. Program.*, vol. 2015, pp. 1–13, Jan. 2015, doi: 10.1155/2015/172879.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, ''Generative adversarial networks,'' *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020, doi: 10.1145/3422622.

[27] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, ''Modeling tabular data using conditional GAN,'' 2019, *arXiv:1907.00503*.

[28] V. Hodge and J. Austin, ''A survey of outlier detection methodologies,'' *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, Oct. 2004, doi: 10.1007/s10462-004-4304-y.

[29] A. Patcha and J.-M. Park, ''An overview of anomaly detection techniques: Existing solutions and latest technological trends,'' *Comput. Netw.*, vol. 51, no. 12, pp. 3448–3470, Aug. 2007, doi: 10.1016/j.comnet.2007.02.001.

[30] M. Jiang, P. Cui, and C. Faloutsos, ''Suspicious behavior detection: Current trends and future directions,'' *IEEE Intell. Syst.*, vol. 31, no. 1, pp. 31–39, Jan. 2016, doi: 10.1109/mis.2016.5.

[31] C. O. S. Sorzano, J. Vargas, and A. P. Montano, ''A survey of dimensionality reduction techniques,'' 2014, *arXiv:1403.2877*.

[32] J. Gama, A. Ganguly, O. Omitaomu, R. Vatsavai, and M. Gaber, ''Knowledge discovery from data streams,'' *Intell. Data Anal.*, vol. 13, no. 3, pp. 403–404, May 2009, doi: 10.3233/ida-2009-0372.

[33] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, ''Outlier detection for temporal data: A survey,'' *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014, doi: 10.1109/tkde.2013.184.

[34] A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari, ''Detection of review spam: A survey,'' *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3634–3642, 2015, doi: 10.1016/j.eswa.2014.12.029.

[35] N. Jindal and B. Liu, ''Review spam detection,'' in *Proc. 16th Int. Conf. World Wide Web*, May 2007, pp. 1189–1190.

[36] E. M. Knorr, R. T. Ng, and V. Tucakov, ''Distance-based outliers: Algorithms and applications,'' *Very Large Data Bases J.*, vol. 8, nos. 3–4, pp. 237–253, 2000, doi: 10.1007/s007780050006.

[37] S. Ramaswamy, R. Rastogi, and K. Shim, ''Efficient algorithms for mining outliers from large data sets,'' in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, May 2000, pp. 93–104.

[38] C. C. Aggarwal and P. S. Yu, ''Outlier detection for high dimensional data,'' in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, May 2001, pp. 37–46.

[39] J. Han and M. Kamber, *Data Mining Concepts and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2001.

[40] D. Yu and G. Sheikholeslami, ''A find out: Finding outliers in very large datasets,'' in *Knowledge and Information Systems*, 2002, pp. 387–412.

[41] M. F. Jiang, S. S. Tseng, and C. M. Su, ''Two-phase clustering process for outlier detection,'' *Pattern Recognit. Lett.*, vol. 22, no. 6–7, pp. 691–700, 2001.

[42] C. C. Aggarwal and P. S. Yu, ''An effective and efficient algorithm for high-dimensional outlier detection,'' *Int. J. Very Large Data Bases*, vol. 14, no. 2, pp. 211–221, 2005, doi: 10.1007/s00778-004-0125-5.

[43] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen, ''ECOD: Unsupervised outlier detection using empirical cumulative distribution functions,'' *IEEE Trans. Knowl. Data Eng.*, early access, Mar. 16, 2022, doi: 10.1109/tkde.2022.3159580.

[44] G. Dudek and J. Szkutnik, ''Daily load curves in distribution networks—Analysis of diversity and outlier detection,'' in *Proc. 18th Int. Sci. Conf. Electr. Power Eng. (EPE)*, May 2017, pp. 1–5.

[45] E. Andersen, M. Chiarandini, M. Hassani, S. Janicke, P. Tampakis, and A. Zimek, ''Evaluation of probability distribution distance metrics in traffic flow outlier detection,'' in *Proc. 23rd IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2022, pp. 64–69.

[46] Y. Chen, X. Dang, H. Peng, H. L. Bart, and H. L. Bart, ''Outlier detection with the kernelized spatial depth function,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 288–305, Feb. 2009, doi: 10.1109/TPAMI.2008.72.

[47] S. Lu, L. Liu, J. Li, and T. D. Le, ''Effective outlier detection based on Bayesian network and proximity,'' in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 134–139.

[48] M. Kim, S. Jung, and S. Kim, ''Fault detection method using inverse distance weight-based local outlier factor,'' in *Proc. Int. Conf. Fuzzy Theory Appl. (iFUZZY)*, Oct. 2021, pp. 1–5.

[49] X. Wang, Y. Chen, and X. L. Wang, ''A centroid-based outlier detection method,'' in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2017, pp. 1411–1416.

[50] M. A. Haque and H. Mineno, ''Proposal of online outlier detection in sensor data using kernel density estimation,'' in *Proc. 6th IIAI Int. Congr. Adv. Appl. Informat. (IIAI-AAI)*, Jul. 2017, pp. 1051–1052.

[51] Y. Tao and D. Pi, ''Unifying density-based clustering and outlier detection,'' in *Proc. 2nd Int. Workshop Knowl. Discovery Data Mining*, Jan. 2009, pp. 644–647.

[52] G. Liu, J. Pang, X. Piao, and S. Huang, ''The discovery of attribute feature cluster for any clustering result based on outlier detection technique,'' in *Proc. Int. Conf. Internet Comput. Sci. Eng.*, Jan. 2008, pp. 68–72.

[53] R. Pamula, J. K. Deka, and S. Nandi, ''An outlier detection method based on clustering,'' in *Proc. 2nd Int. Conf. Emerg. Appl. Inf. Technol.*, Feb. 2011, pp. 253–256.

[54] B. Angelin and A. Geetha, ''Outlier detection using clustering techniques—K-means and K-median,'' in *Proc. 4th Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, May 2020, pp. 373–378.

[55] Y. Wang, B. Dai, G. Hua, J. Aston, and D. Wipf, ''Recurrent variational autoencoders for learning nonlinear generative models in the presence of outliers,'' *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1615–1627, Dec. 2018, doi: 10.1109/jstsp.2018.2876995.

[56] Y. Li and H. Wu, ''A clustering method based on K-means algorithm,'' *Phys. Proc.*, vol. 25, pp. 1104–1109, Jan. 2012, doi: 10.1016/j.phpro.2012.03.206.

[57] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, ''COPOD: Copula-based outlier detection,'' in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 1118–1123.

[58] K. J. Paul and R. Harilal, ''Implementation of MAD and mean absolute deviation based smoothing algorithm for displacement data in digital image correlation technique,'' Indian Inst. Technol. Hyderabad, Hyderabad, India, Tech. Rep., 2014, pp. 1–6.

[59] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen, "ECOD: Unsupervised outlier detection using empirical cumulative distribution functions," *IEEE Trans. Knowl. Data Eng.*, early access, Mar. 16, 2022, doi: 10.1109/tkde.2022.3159580.

[60] A. Goodge, B. Hooi, S. K. Ng, and W. S. Ng, "LUNAR: Unifying local outlier detection methods via graph neural networks," 2021, *arXiv:2112.05355*.

[61] A. Bounsiar and M. G. Madden, "One-class support vector machines revisited," in *Proc. Int. Conf. Inf. Sci. Appl. (ICISA)*, May 2014, pp. 1–4.

[62] *Welcome to*. Python.org. Accessed: Dec. 24, 2022. [Online]. Available: http://www.python.org

[63] F. Chollet, *Deep Learning for Humans*. Mountain View, CA, USA: Keras, 2017.

[64] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2000, pp. 93–104.

[65] Z. Cheng, C. Zou, and J. Dong, "Outlier detection using isolation forest and local outlier factor," in *Proc. Conf. Res. Adapt. Convergent Syst.*, Sep. 2019, pp. 161–168.

[66] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: Local outlier probabilities," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1649–1652.

[67] R. Gupta and K. Pandey, "Density based outlier detection technique," in *Advances in Intelligent Systems and Computing*, New Delhi, India: Springer, 2016, pp. 51–58.

[68] O. Alghushairy, R. Alsini, T. Soule, and X. Ma, "A review of local outlier factor algorithms for outlier detection in big data streams," *Big Data Cognit. Comput.*, vol. 5, no. 1, p. 1, Dec. 2020, doi: 10.3390/bdcc5010001.

[69] H. Xu, L. Zhang, P. Li, and F. Zhu, "Outlier detection algorithm based on k-nearest neighbors-local outlier factor," *J. Algorithms Comput. Technol.*, vol. 16, Jan. 2022, Art. no. 174830262210781.

[70] A. Liu and J. Zhang, "An outlier mining algorithm based on local weighted k-density," in *Proc. 8th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Jul. 2011, pp. 1504–1508.

[71] J. Y. Lee and R. Tukhvatov, "Evaluations of similarity measures on VK for link prediction," *Data Sci. Eng.*, vol. 3, no. 3, pp. 277–289, 2018, doi: 10.1007/s41019-018-0073-5.

[72] E. M. Jordaan and G. F. Smits, "Robust outlier detection using SVM regression," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2004, pp. 2017–2022.

[73] X.-Y. Yang, J. Liu, M.-Q. Zhang, and K. Niu, "A new multi-class SVM algorithm based on one-class SVM," in *Computational Science ICCS 2007*. Berlin, Germany: Springer, 2007, pp. 677–684.

[74] E. H. Budiarto, A. E. Permanasari, and S. Fauziati, "Unsupervised anomaly detection using K-means, local outlier factor and one class SVM," in *Proc. 5th Int. Conf. Sci. Technol. (ICST)*, Jul. 2019, pp. 1–5.

[75] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in *Proc. ACM SIGKDD Workshop Outlier Detection Description*, Aug. 2013, pp. 8–15.

[76] H. Lukashevich, S. Nowak, and P. Dunker, "Using one-class SVM outliers detection for verification of collaboratively tagged image training sets," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jun. 2009, pp. 682–685.

[77] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *Appl. Statist.*, vol. 28, no. 1, p. 100, 1979, doi: 10.2307/2346830.

[78] D. Marutho, S. H. Handaka, E. Wijaya, and Muljono, "The determination of cluster number at k-mean using elbow method and purity evaluation on headline news," in *Proc. Int. Seminar Appl. Technol. Inf. Commun.*, Sep. 2018, pp. 533–538.

[79] T. M. Kodinariya and P. R. Makwana, "Review on determining number of cluster in K-means clustering," *Int. J.*, vol. 1, no. 6, pp. 90–95, 2013.

[80] C. Xiong, Z. Hua, K. Lv, and X. Li, "An improved K-means text clustering algorithm by optimizing initial cluster centers," in *Proc. 7th Int. Conf. Cloud Comput. Big Data (CCBD)*, Nov. 2016, pp. 265–268.

[81] A. Kuraria, N. Jharbade, and M. Soni, "Centroid selection process using WCSS and elbow method for K-mean clustering algorithm in data mining," *Int. J. Sci. Res. Sci., Eng. Technol.*, pp. 190–195, Dec. 2018, doi: 10.32628/ijsrset21841122.

[82] R. Gao, T. Zhang, S. Sun, and Z. Liu, "Research and improvement of isolation forest in detection of local anomaly points," *J. Phys. Conf. Ser.*, vol. 1237, no. 5, 2019, Art. no. 052023, doi: 10.1088/1742-6596/1237/5/052023.

[83] M. Tokovarov and P. Karczmarek, "A probabilistic generalization of isolation forest," *Inf. Sci.*, vol. 584, pp. 433–449, Jan. 2022, doi: 10.1016/j.ins.2021.10.075.

[84] W. S. Al Farizi, I. Hidayah, and M. N. Rizal, "Isolation forest based anomaly detection: A systematic literature review," in *Proc. 8th Int. Conf. Inf. Technol., Comput. Electr. Eng. (ICITACEE)*, Sep. 2021, pp. 118–122.

[85] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.

[86] M. U. Togbe, "Anomaly detection for data streams based on isolation forest using scikit-multiflow," in *Computational Science and Its Applications ICCSA 2020*. Cham, Switzerland: Springer, 2020, pp. 15–30.

[87] S. Wibisono, M. T. Anwar, A. Supriyanto, and I. H. A. Amin, "Multivariate weather anomaly detection using DBSCAN clustering algorithm," *J. Phys., Conf. Ser.*, vol. 1869, no. 1, Apr. 2021, Art. no. 012077, doi: 10.1088/1742-6596/1869/1/012077.

[88] M. Celik, F. Dadaser-Celik, and A. S. Dokuz, "Anomaly detection in temperature data using DBSCAN algorithm," in *Proc. Int. Symp. Innov. Intell. Syst. Appl.*, Jun. 2011, pp. 91–95.

[89] D. Birant and A. Kut, "Spatio-temporal outlier detection in large databases," in *Proc. 28th Int. Conf. Inf. Technol. Interface*, 2006, pp. 291–297.

[90] Z. Akbari and R. Unland, "Automated determination of the input parameter of DBSCAN based on outlier detection," in *IFIP Advances in Information and Communication Technology*. Cham, Switzerland: Springer, 2016, pp. 280–291.

[91] T. Manh Thang and J. Kim, "The anomaly detection by using DBSCAN clustering with multiple parameters," in *Proc. Int. Conf. Inf. Sci. Appl.*, Apr. 2011, pp. 1–5.

[92] J. Dugundji, "Envelopes and pre-envelopes of real waveforms," *IRE Trans. Inf. Theory*, vol. 4, no. 1, pp. 53–57, Mar. 1958, doi: 10.1109/tit.1958.1057435.

[93] P. Mahalanobis, "On the generalized distance in statistics," Tech. Rep., 1936.

[94] G. J. McLachlan, "Mahalanobis distance," *Resonance*, vol. 4, no. 6, pp. 20–26, Jun. 1999, doi: 10.1007/bf02834632.

[95] M. D'Agostino and V. Dardanoni, "What's so special about Euclidean distance?: A characterization with applications to mobility and spatial voting," *Social Choice Welfare*, vol. 33, no. 2, pp. 211–233, Aug. 2009, doi: 10.1007/s00355-008-0353-5.

[96] R. Hidayat, I. T. R. Yanto, A. A. Ramli, M. F. M. Fudzee, and A. S. Ahmar, "Generalized normalized Euclidean distance based fuzzy soft set similarity for data classification," *Comput. Syst. Sci. Eng.*, vol. 38, no. 1, pp. 119–130, 2021, doi: 10.32604/csse.2021.015628.

[97] E. Müller, I. Assent, P. Iglesias, Y. Mulle, and K. Bohm, "Outlier ranking via subspace analysis in multiple views of the data," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 529–538.

[98] C. C. Aggarwal, "High-dimensional outlier detection: The subspace method," in *Outlier Analysis*. Cham, Switzerland: Springer, 2017, pp. 149–184.

[99] H.-P. Kriegel, P. Kroger, E. Schubert, and A. Zimek, "Outlier detection in arbitrarily oriented subspaces," in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 379–388.

[100] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: A review," *ACM SIGKDD Explorations Newslett.*, vol. 6, no. 1, pp. 90–105, Jun. 2004.

[101] Y. Almardeny, N. Boujnah, and F. Cleary, "A novel outlier detection method for multivariate data," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 9, pp. 4052–4062, Sep. 2022, doi: 10.1109/tkde.2020.3036524.

[102] Q. Wang, Q. Gao, X. Gao, and F. Nie, "Angle principal component analysis," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2936–2942.

[103] S. Dray, "On the number of principal components: A test of dimensionality based on measurements of similarity between matrices," *Comput. Statist. Data Anal.*, vol. 52, no. 4, pp. 2228–2237, Jan. 2008, doi: 10.1016/j.csda.2007.07.015.

[104] H. T. Eastment and W. J. Krzanowski, "Cross-validatory choice of the number of components from a principal component analysis," *Technometrics*, vol. 24, no. 1, p. 73, 1982, doi: 10.2307/1267581.

[105] J. Gower, "Statistical methods of comparing different multivariate analyses of the same data," *Math. Archaeolog. historical Sci.*, vol. 138, p. 149, Jan. 1971.

[106] R. J. Harris, *A Primer of Multivariate Statistics*. Mahway, NJ, USA: Lawrence, 2001.

[107] W.-C. Chang, C.-P. Lee, and C.-J. Lin, "A revisit to support vector data description," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2013.

[108] C. Liu and K. Gryllias, "A deep support vector data description method for anomaly detection in helicopters," in *Proc. PHM Soc. Eur. Conf.*, vol. 6, 2021, p. 9.

[109] S. Kim, Y. Choi, and M. Lee, "Deep learning with support vector data description," *Neurocomputing*, vol. 165, pp. 111–117, Oct. 2015, doi: 10.1016/j.neucom.2014.09.086.

[110] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

[111] K. Song, Y. Qin, B. Xu, N. Zhang, and J. Yang, "Study on outlier detection method of the near infrared spectroscopy analysis by probability metric," *Spectrochimica Acta A, Mol. Biomolecular Spectrosc.*, vol. 280, Nov. 2022, Art. no. 121473, doi: 10.1016/j.saa.2022.121473.

[112] S. N. Lahiri, M. S. Kaiser, N. Cressie, and N.-J. Hsu, "Prediction of spatial cumulative distribution functions using subsampling," *J. Amer. Stat. Assoc.*, vol. 94, no. 445, p. 86, 1999, doi: 10.2307/2669680.

[113] W. W. Esty and J. D. Banfield, "The box-percentile plot," *J. Stat. Softw.*, vol. 8, no. 17, pp. 1–14, 2003, doi: 10.18637/jss.v008.i17.

[114] C. Reimann, P. Filzmoser, and R. G. Garrett, "Background and threshold: Critical comparison of methods of determination," *Sci. Total Environ.*, vol. 346, nos. 1–3, pp. 1–16, Jun. 2005, doi: 10.1016/j.scitotenv.2004.11.023.

[115] R. Chellappa, "Gaussian mixture models," in *Encyclopedia Biometrics*. Boston, MA, USA: Springer, 2009, pp. 659–663.

[116] D. W. Scott, "Outlier detection and clustering by partial mixture modeling," in *COMPSTAT 2004 Proceedings in Computational Statistics*, Heidelberg, Germany: Physica-Verlag, 2004, pp. 453–464.

[117] L. Li, J. Hansman, R. Palacios, and R. Welsch, "Anomaly detection via a Gaussian mixture model for flight operation and safety monitoring," *Transp. Res. C, Emerg. Technol.*, vol. 64, pp. 45–57, Mar. 2016, doi: 10.1016/j.trc.2016.01.007.

[118] A. Reddy, M. Ordway-West, M. Lee, M. Dugan, J. Whitney, R. Kahana, B. Ford, J. Muedsam, A. Henslee, and M. Rao, "Using Gaussian mixture models to detect outliers in seasonal univariate network traffic," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2017, pp. 229–234.

[119] N. Ding, H. Ma, H. Gao, Y. Ma, and G. Tan, "Real-time anomaly detection based on long short-term memory and Gaussian mixture model," *Comput. Electr. Eng.*, vol. 79, Oct. 2019, Art. no. 106458, doi: 10.1016/j.compeleceng.2019.106458.

[120] D. C. Howell, "Median absolute deviation," in *Encyclopedia of Statistics in Behavioral Science*. Hoboken, NJ, USA: Wiley, 2005.

[121] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *J. Experim. Social Psychol.*, vol. 49, no. 4, pp. 764–766, Jul. 2013, doi: 10.1016/j.jesp.2013.03.013.

[122] K. Kannan, K. Senthamarai, and S. Manoj, "Labeling methods for identifying outliers," *Int. J. Statist. Syst.*, vol. 10, no. 2, pp. 231–238, 2015.

[123] P. J. Rousseeuw and C. Croux, "Alternatives to the median absolute deviation," *J. Amer. Stat. Assoc.*, vol. 88, no. 424, p. 1273, 1993, doi: 10.2307/2291267.

[124] J. Yang, S. Rahardja, and P. Fränti, "Outlier detection: How to threshold outlier scores?" in *Proc. Int. Conf. Artif. Intell., Inf. Process. Cloud Comput.*, Dec. 2019, pp. 1–6.

[125] A. Kharitonov, A. Nahhas, M. Pohl, and K. Turowski, "Comparative analysis of machine learning models for anomaly detection in manufacturing," *Proc. Comput. Sci.*, vol. 200, pp. 1288–1297, Jan. 2022, doi: 10.1016/j.procs.2022.01.330.

[126] J.-D. Fermanian, D. Radulovic, and M. Wegkamp, "Weak convergence of empirical copula processes," *Bernoulli*, vol. 10, no. 5, pp. 847–860, Oct. 2004, doi: 10.3150/bj/1099579158.

[127] X. Wang, L. Wang, J. Wang, K. Sun, and Q. Wang, "Hyperspectral anomaly detection via background purification and spatial difference enhancement," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, doi: 10.1109/lgrs.2021.3140087.

[128] M. L. Katz, "The probability in the tail of a distribution," *Ann. Math. Statist.*, vol. 34, no. 1, pp. 312–318, Mar. 1963, doi: 10.1214/aoms/1177704268.

[129] A. Agarwal and N. Gupta, "Comparison of outlier detection techniques for structured data," 2021, *arXiv:2106.08779*.

[130] J. Zhang, "Advancements of outlier detection: A survey," *ICST Trans. Scalable Inf. Syst.*, vol. 13, no. 1, p. e2, Feb. 2013, doi: 10.4108/trans.sis.2013.01-03.e2.

[131] N. Paulauskas and A. Baskys, "Application of histogram-based outlier scores to detect computer network anomalies," *Electronics*, vol. 8, no. 11, p. 1251, Nov. 2019, doi: 10.3390/electronics8111251.

[132] Y. Wang, S. Zhu, and C. Li, "Research on an ensemble anomaly detection algorithm," *J. Phys., Conf. Ser.*, vol. 1314, no. 1, Oct. 2019, Art. no. 012198, doi: 10.1088/1742-6596/1314/1/012198.

[133] M. Gebski and R. K. Wong, "An efficient histogram method for outlier detection," in *Advances in Databases: Concepts, Systems and Applications*. Berlin, Germany: Springer, 2007, pp. 176–187.

[134] X. Zhao, Y. Zhang, S. Xie, Q. Qin, S. Wu, and B. Luo, "Outlier detection based on residual histogram preference for geometric multi-model fitting," *Sensors*, vol. 20, no. 11, p. 3037, May 2020, doi: 10.3390/s20113037.

[135] L. Pappalardo, G. Rossetti, and D. Pedreschi, "'How well do we know each other?' detecting tie strength in multidimensional social networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2012, pp. 1040–1045.

[136] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Soc. Netw.*, vol. 25, no. 3, pp. 211–230, 2003, doi: 10.1016/s0378-8733(03)00009-1.

CHARVI BANNUR (Student Member, IEEE) is currently pursuing the B.Tech. degree in computer science engineering with People's Education Society (PES), Bengaluru, India.

Since 2021, she has been a Research Assistant with the Research Laboratory of PES. She is the author of multiple research articles in the fields of machine learning and artificial intelligence. Her research interests include deep neural networks, graph theory applications in the realm of social network analysis, data mining, and information retrieval.

Ms. Bannur has received numerous academic accolades and scholarships at her university and strives toward academic excellence. She was a recipient of the 7th IEEE International Conference on Recent Advances and Innovations in Engineering Best Paper Award, in 2022.

CHAITRA BHAT (Student Member, IEEE) is currently pursuing the B.Tech. degree in computer science and engineering with People's Education Society (PES) University, Bengaluru.

Since 2021, she has been a Research Assistant with PES University, in the field of natural language processing. She is the author of multiple research articles in the fields of machine learning and graph theory. Her research interests include machine learning, deep learning, image classification and recognition, graph theory and applications, and natural language processing.

Ms. Bhat received the Best Paper Award at the 7th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), in 2022.

KUSHAGRA SINGH (Student Member, IEEE) is currently pursuing the bachelor's degree in computer science with People's Education Society (PES) University, Bengaluru. In addition to being a motivated learner, who places a strong focus on academic performance, he favors exploring a number of areas, particularly those related to computer science. He has initiated and collaborated in numerous projects that make use of his knowledge in machine learning, deep learning, blockchain technology, and big data. He has taken part in numerous hackathons and extracurricular events, where he earned honors in many of them. The IEEE Bangalore Chapter's Internship Program was successfully completed by him. His research article "Data Regeneration From Poisoned Dataset," written along with his colleagues, was presented at the ICRAIE 2022 Conference and was chosen for the Best Paper Award.

SHRIRANG AMBAJI KULKARNI (Senior Member, IEEE) received the B.E. degree in computer science and engineering from Karnatak University, Dharwad, in 2000, the M.Tech. degree in computer science and engineering from Visvesvaraya Technological University, Belgaum, in 2004, and the Ph.D. degree from the Faculty of Computer and Information Science, Visvesvaraya Technological University, in 2012.

From 2001 to 2018, he has worked in various capacities as an assistant professor, an associate professor, and a professor in multiple universities and engineering institutes in India. From 2021 to 2022, he was a Postdoctoral Research Fellow with the Health Informatics Laboratory, University of Central Florida, USA. He is currently an Associate Professor with the Department of Computer Science and Engineering, National Institute of Engineering, Mysore, India. He is also a Project Consultant with Mobirey Technologies Pvt. Ltd., USA, on AI/ML technologies applied to the financial domain. He is the author of three books, more than 40 articles, and many patents under review. He is a Senior Member of ACM.

MRITYUNJAY DODDAMANI received the Ph.D. degree in mechanical engineering from the National Institute of Technology Karnataka, Surathkal, in 2012. He is currently an Associate Professor with the School of Mechanical and Materials Engineering, Indian Institute of Technology (IIT), Mandi, Himachal Pradesh, India. He has published more than 75 articles in the areas of materials development for specified applications, additive manufacturing, and machine learning. He was funded by various funding agencies in India for his research works.