Scientific African 26 (2024) e02386

Contents lists available at ScienceDirect

Scientific African

journal homepage: www.elsevier.com/locate/sciaf

Anomaly detection using unsupervised machine learning algorithms: A simulation study

Edmund Fosu Agyemang *

School of Mathematical and Statistical Science, College of Sciences, University of Texas Rio Grande Valley, USA
Department of Statistics and Actuarial Science, College of Basic and Applied Sciences, University of Ghana, Ghana
Department of Computer Science, Ashesi University, No. 1 University Avenue, Berekuso, Accra, Ghana

ARTICLE INFO

Editor name: DR B Gyampoh

Keywords:
Anomaly detection
Unsupervised machine learning algorithms
One-class support vector machine
Isolation forest
Local outlier factor
Robust covariance

ABSTRACT

This study presents a comprehensive evaluation of five prominent unsupervised machine learning anomaly detection algorithms: One-Class Support Vector Machine (One-Class SVM), One-Class SVM with Stochastic Gradient Descent (SGD), Isolation Forest (iForest), Local Outlier Factor (LOF), and Robust Covariance (Elliptic Envelope). Through systematic analysis on a synthetically simulated dataset, the study assessed each algorithm's predictive performance using accuracy, precision, recall, and F1 score specifically for outlier detection. The evaluation reveals that One-Class SVM, Isolation Forest, and Robust Covariance are more effective in identifying outliers in the synthetic simulated dataset, with Isolation Forest slightly outperforming the other algorithms in terms of balancing precision and recall. One-Class SVM with SGD shows promise in precision but needs adjustment to improve recall. Local Outlier Factor may require parameter tuning or may not be as suitable for this particular dataset's characteristics. The findings reveal significant variations in performance, highlighting the strengths and limitations of each method in identifying anomalies. This research contributes to the field of machine learning by demonstrating that the selection of an anomaly detection algorithm should be a considered decision, taking into account the specific characteristics of the data and the operational context of its application. Future work should explore parameter optimization, the impact of dataset characteristics on model performance, and the application of these models to real-world datasets to validate their efficacy in practical anomaly detection scenarios.

Introduction

Anomaly detection, a critical component of data analysis, plays a pivotal role in identifying irregularities that deviate from normal patterns in datasets [1]. In the era of digital transformation, the ability to automatically identify unusual patterns or anomalies in data has become increasingly crucial across various sectors, including finance, healthcare, cybersecurity, and manufacturing. Anomalies can indicate significant, often critical, information ranging from fraudulent transactions to malfunctioning equipment. The challenge, however, lies in detecting these irregularities, especially when the definition of 'normal' is constantly evolving and the nature of anomalies can be highly unpredictable. Traditional anomaly detection methods, which often rely on predefined thresholds or specific assumptions about data distribution, are increasingly inadequate due to their lack of flexibility and scalability. These anomalies can indicate significant, often critical, actionable insights across various domains. In cybersecurity, anomaly detection systems identify unusual patterns that may signify security breaches, such as unauthorized access or malware activities [2]. In the

* Correspondence to: Department of Statistics and Actuarial Science, College of Basic and Applied Sciences, University of Ghana, Ghana.
E-mail address: edmundfosu6@gmail.com.

https://doi.org/10.1016/j.sciaf.2024.e02386
Received 15 February 2024; Received in revised form 6 September 2024; Accepted 13 September 2024
Available online 19 September 2024

realm of fraud detection, these algorithms help in spotting unusual transactions that could indicate credit card fraud [3], insurance fraud, election fraud [4–6] or other types of financial malfeasance [7]. Moreover, in industrial fault detection, anomaly detection facilitates the early identification of equipment failures, ensuring timely maintenance and reducing downtime [8]. The ability to accurately and efficiently identify anomalies can lead to significant cost savings, improved safety, and enhanced operational efficiency.

The evolution of anomaly detection techniques has been marked by a transition from traditional statistical methods to advanced machine learning approaches. Early methods relied on statistical models to define normalcy, using distribution-based or distance-based metrics to identify outliers. For instance, techniques such as Z-score and IQR (Interquartile Range) were widely used for detecting anomalies in univariate data [9,10]. However, the advent of complex, high-dimensional data in modern applications highlighted the limitations of these traditional methods, leading to the development of more sophisticated machine learning algorithms. Recent advancements in machine learning and artificial intelligence have introduced a plethora of anomaly detection techniques capable of handling the complexity and volume of contemporary datasets. These include supervised methods, where models are trained on labeled datasets comprising both normal and anomalous instances [11,12], and unsupervised methods, which do not require labeled data and are particularly useful in scenarios where anomalies are rare or not well-defined [13]. Several studies have compared different anomaly detection algorithms, providing valuable insights into their relative performance across various settings. These comparative analyses are crucial for understanding the strengths and limitations of each method, guiding practitioners in selecting the most appropriate algorithm for their specific needs. For example, research has shown that isolation-based methods, such as the Isolation Forest algorithm, perform well in high-dimensional settings, offering an efficient and effective means of identifying anomalies [14]. In contrast, density-based methods like the Local Outlier Factor (LOF) excel in datasets where anomalies form clusters, allowing for a refined detection of local outliers [15]. Moreover, studies have also explored hybrid models that combine the benefits of multiple approaches, such as integrating machine learning algorithms with traditional statistical methods to improve detection accuracy.

The continuous evolution of anomaly detection techniques, fueled by ongoing research and the increasing complexity of datasets, highlights the importance of comparative studies in advancing the field. The mathematical underpinnings of anomaly detection algorithms form the foundation of their operational principles. One-Class Support Vector Machine (SVM), for instance, constructs a decision function that separates the majority of the data points from the outliers, effectively creating a boundary around the normal data [16]. This method, and its variant incorporating Stochastic Gradient Descent (SGD), offers a robust way to handle large-scale datasets through efficient optimization techniques [17]. Similarly, the Isolation Forest algorithm isolates anomalies by randomly selecting features and splitting values, requiring fewer splits for anomalies than for normal points, thus capitalizing on the anomalies' inherent 'susceptibility' to isolation [18]. The Local Outlier Factor (LOF) algorithm assesses the local density deviation of a given data point with respect to its neighbors, identifying regions of similar density and highlighting points that stand out [19]. Lastly, the Robust Covariance method, or Elliptic Envelope, assumes a Gaussian distribution of the dataset, identifying outliers as deviations from this model [20]. The theoretical diversity of these algorithms illustrates the rich landscape of anomaly detection techniques, each with its unique strengths and applicability to different types of data and anomalies.

The advent of machine learning (ML) has introduced advanced capabilities in identifying complex patterns in data. Among these, unsupervised ML algorithms, which do not require labeled data, are particularly promising for anomaly detection. They have the potential to autonomously adapt to evolving data patterns and detect anomalies without explicit prior knowledge of what constitutes an anomaly [21]. However, the application of unsupervised ML algorithms in anomaly detection faces several challenges. These include the selection of appropriate algorithms, the handling of high-dimensional data, the differentiation between noise and true anomalies, and the interpretation of the results in a meaningful way. Despite the potential of unsupervised ML for anomaly detection, there is a noticeable gap in the comprehensive evaluation of these algorithms under varied simulation conditions. A systematic exploration of their performance across different data types, anomaly characteristics, and industry domains is lacking. This gap hinders the ability of practitioners to select and implement the most effective unsupervised ML strategies for anomaly detection in real-world scenarios. This research seeks to bridge this gap by conducting a thorough simulation study of unsupervised ML algorithms for anomaly detection.

This study aims to evaluate the performance of various unsupervised ML algorithms, identify best practices in their application, and develop guidelines for their implementation in practical settings. By focusing on a simulation approach, this study will allow for the controlled manipulation of data characteristics and anomaly parameters, thereby providing a deep understanding of the algorithms' strengths and limitations under different conditions. The importance of this research lies in its potential to advance the field of anomaly detection. By providing a comprehensive evaluation of unsupervised ML algorithms, the study will contribute valuable insights into their applicability, and efficiency. This, in turn, will enable organizations to better protect against fraud, improve operational reliability, enhance customer satisfaction, and prevent significant financial losses. Moreover, the findings of this study could pave the way for future research directions, including the development of new algorithms, the refinement of existing methodologies, and the exploration of hybrid approaches. The remainder of the paper is organized as follows: Section "Data and methods" discusses the data and methods used for the study. Section "Mathematical framework" provides the mathematical framework of the study. Section "Results and discussion" discusses the results of the study whilst Section "Conclusion & recommendation of the study" concludes the study and provide recommendations for further work.

Data and methods

The dataset was synthetically generated to simulate a two-dimensional feature space, comprising 100 normal data points centered around two distinct means (+2 and −2) with a standard deviation of 0.3, and 20 outliers uniformly distributed (+4 and −4) across the feature space. The *np.r_*$[X + 2, X - 2]$ function was used to concatenate the two sets of shifted points along the first axis (vertically), resulting in a 200×2 dataset. This setup was chosen to mimic real-world scenarios where anomalies are sparse and not centered around the majority class. It is worth knowing that as the size of the dataset increases, anomaly detection techniques such as One-Class SVM, One-Class SVM with SGD, Isolation Forest, Local Outlier Factor, and Robust Covariance encounter both advantages and challenges. Larger datasets typically enhance detection accuracy by better representing the "normal" data distribution, which helps in more accurately identifying anomalies.

However, this increase in data volume also leads to longer training times and greater memory usage, posing scalability issues particularly for methods like One-Class SVM that are computationally intensive. Some techniques, like Isolation Forest and Local Outlier Factor, scale more efficiently with large datasets, while adaptations such as One-Class SVM with SGD are designed to address scalability by using stochastic gradient descent, which reduces memory demands. Additionally, larger datasets can bolster the robustness of models, making them less susceptible to noise and minor data variations, though careful parameter tuning becomes crucial to maintain performance.

The synthetic simulated data was then applied to the five (5) unsupervised machine learning algorithms namely One-Class Support Vector Machine (SVM), One-Class SVM with Stochastic Gradient Descent (SGD), Isolation Forest, Local Outlier Factor (LOF), and Robust Covariance (Elliptic Envelope). The One-Class SVM model was implemented using the *sklearn.svm.OneClassSVM* library in Python, with a Radial Basis Function (RBF) kernel. Extensive hyperparameter tuning was conducted and this resulted in the following optimal parameters. For the One-Class SVM, the optimal hyperparameters were as follows: $nu = 0.1$, indicating the expected fraction of outliers in the dataset, and $gamma = 0.1$, determining the inverse of the radius of influence of samples selected by the model as support vectors. The One-Class SVM with SGD model leverages the *sklearn.linear_model.SGDOneClassSVM* library in Python for implementation. This variant optimizes the One-Class SVM objective using Stochastic Gradient Descent, making it suitable for large datasets.

Key hyperparameters after fine tuning include $nu = 0.1$, which estimates the proportion of outliers in the dataset, and *max_iter = 1000*, determining the number of passes over the training data (epochs). To ensure convergence, the model was allowed to run through the data 1000 times. The *learning_rate = optimal* was employed to allow the algorithm to automatically adjust the learning rate over time for optimial convergence. For the Isolation Forest, key hyperparameters after fine tuning include *n_estimators = 100* which deployed 100 base estimators in the ensemble to ensure a comprehensive learning from the data. *max_samples = 'auto'* allowed the model to use a default value to draw samples for training each base estimator. Also, for the LOF, *n_neighbors = 20* was chosen to consider 20 closest neighbors to estimate the local density, providing a balance between sensitivity and specificity. For the Robust Covariance, *contamination* hyperparameter was set to 0.1, indicating that the model expects approximately 10% of the data points to be outliers. This value helps the model determine the threshold for identifying whether a data point should be considered an outlier based on the statistical properties of the data.

All the five (5) unsupervised anomaly detection algorithms were trained exclusively on the normal data points to learn the region of the feature space occupied by the majority class. Subsequently, it was used to predict anomalies, classifying each data point as either a normal observation or an outlier based on the learned decision function.

Fig. 1 summarizes the working architecture of the five (5) unsupervised machine learning algorithms.

Below is a summary of the conceptual workflow adopted for the study;

1. **Data Generation and Preparation:** The study starts with the synthetic generation of data to simulate real-world scenarios where anomalies are sparse and distinct from the majority class. This includes creating a dataset with normal data points centered around two distinct means and outliers uniformly distributed across the feature space.
2. **Algorithm Implementation and Hyperparameter Tuning:** Each algorithm—One-Class SVM, One-Class SVM with SGD, Isolation Forest, Local Outlier Factor, and Robust Covariance—is implemented using Python. Extensive tuning of hyperparameters is conducted to optimize each model's performance based on the expected fraction of outliers and other relevant parameters.
3. **Model Fitting:** The models are trained exclusively on the normal data points. This step is crucial as it allows the algorithms to learn the region of the feature space occupied by the majority class without being influenced by the outliers.
4. **Anomaly Prediction:** Using the trained models, the study then predicts anomalies by classifying each data point in the extended dataset (including outliers) as either normal or an outlier. This step tests each model's ability to generalize from the training data to unseen data.
5. **Visualization and Evaluation:** The results are visualized and the performance of each algorithms is evaluated based on accuracy, precision, recall, and F1 score, providing a comprehensive view of their performance.
6. **Comparative Analysis:** The study includes a comparative analysis of the models, discussing their strengths and weaknesses in detecting anomalies. This analysis is supported by visual representations and statistical metrics that highlight each algorithm's suitability for different types of data anomalies.

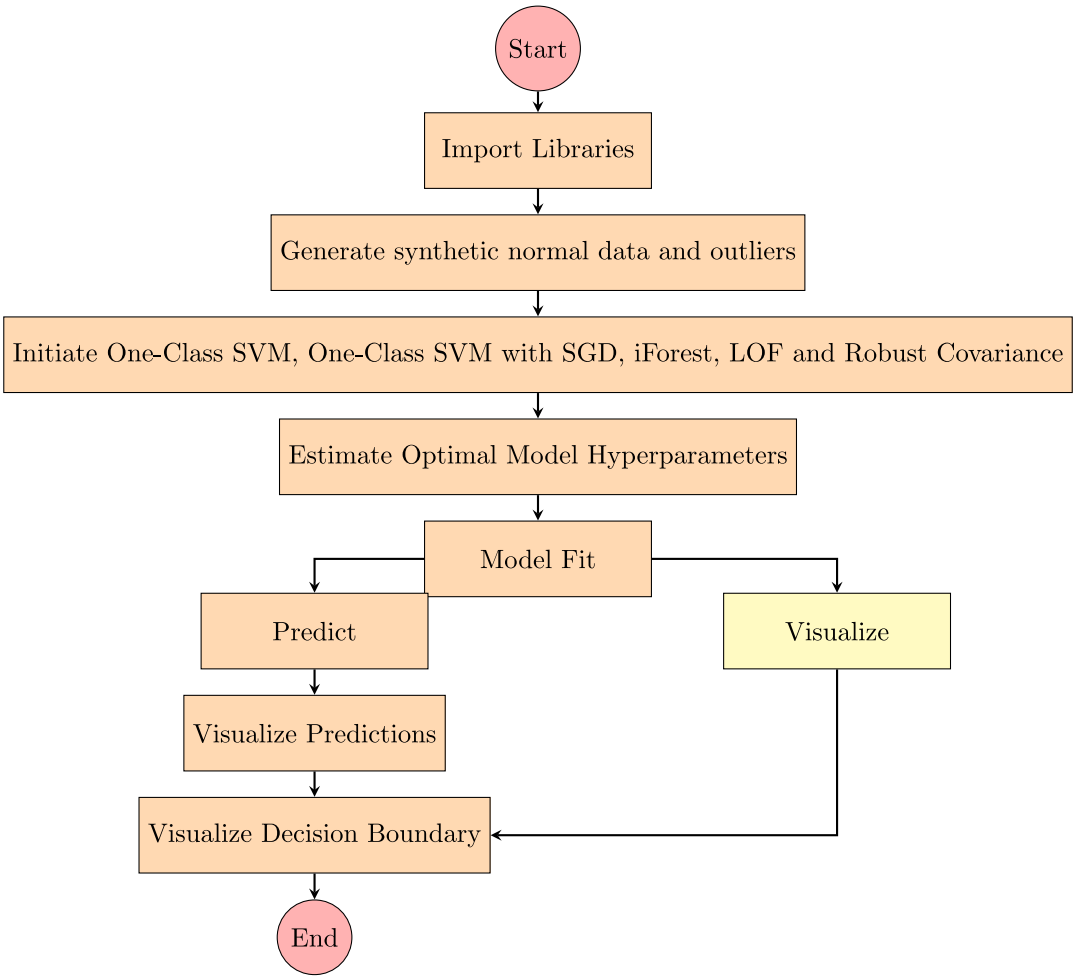


Fig. 1. Working architecture of the anomaly detection algorithms.

Mathematical framework

One-Class Support Vector Machine (SVM)

One-Class Support Vector Machine (SVM) is fundamentally a boundary-based method that aims to find the optimal hyperplane or boundary separating the normal data points from the outliers [22]. Unlike traditional SVM that focuses on maximizing the margin between two classes, One-Class SVM seeks to condense the majority of data points in a way that distances them from the origin in a high-dimensional feature space, effectively isolating outliers. This method relies on the concept of kernel trick to transform data into a higher-dimensional space where it is easier to segregate outliers from normal observations, making it particularly effective for datasets where anomalies are sparse and not well-defined [23].

Let $X = \{x_1, x_2, \ldots, x_n\}$ be the synthetic simulated dataset in a d-dimensional space where $x_i \in \mathbb{R}^d$. The objective of One-Class SVM is to find a function that returns +1 for a region capturing most of the data points and −1 elsewhere. A mapping $\phi : \mathbb{R}^d \to \mathbb{F}$ is applied to transform the input space into a higher-dimensional feature space $\mathbb{F}$, where linear separation is more feasible. The Kernel trick is employed to facilitate this transformation, allowing the computation of the dot product in the feature space without explicitly performing the transformation, defined as $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. The One-Class SVM solves the following primal optimization problem in (1):

$$\min_{w,\xi_i,\rho} \frac{1}{2}\|w\|^2 - \rho + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i \tag{1}$$

subject to the constraints:

$$(w \cdot \phi(x_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n$$

where w is the normal vector to the hyperplane, ρ is the decision threshold, ξ_i are the slack variables allowing for the soft margin, and ν is a parameter that controls the trade-off between maximizing the distance of the hyperplane from the origin and minimizing the fraction of outliers. The dual formulation of the problem, which is solved in practice, is given in (2) by:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i K(x_i, x_i) - \sum_{i,j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) \tag{2}$$

subject to:

$$0 \leq \alpha_i \leq \frac{1}{\nu n}, \quad \sum_{i=1}^{n} \alpha_i = 1$$

where α_i are the Lagrange multipliers. The decision function for a new data point x is given in (3) by:

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i K(x_i, x) - \rho \right) \tag{3}$$

A data point x is classified as an anomaly if $f(x) = -1$ and as normal if $f(x) = +1$. The choice of kernel $K(x_i, x_j)$ significantly affects the performance of the One-Class SVM. Common choices include linear, polynomial, and Radial Basis Function (RBF) kernels. This study utilizes the Gaussian kernel. Parameters such as ν and kernel-specific parameters were carefully selected to balance sensitivity to outliers and generalization to unseen data.

One-Class SVM with Stochastic Gradient Descent (SGD)

One-Class SVM with Stochastic Gradient Descent (SGD) enhances the traditional One-Class SVM by incorporating SGD for optimization, making it more scalable and efficient for large datasets [24]. SGD optimizes the One-Class SVM objective function by iteratively updating the model parameters using a subset of the data, significantly reducing computation time without sacrificing accuracy [25]. This variant is especially suitable for streaming data or situations where the computational efficiency is paramount, offering a pragmatic solution for real-time anomaly detection tasks. Let $X = \{x_1, x_2, \ldots, x_n\}$ represent the synthetic simulated dataset in d-dimensional space, with each $x_i \in \mathbb{R}^d$. The goal of the One-Class SVM is to find a decision function that isolates the majority of data points from the origin, identifying any significant deviations as anomalies.

With reference to the optimization problem in (1), but with SGD, w represents the weight vector perpendicular to the hyperplane, ρ is the offset of the hyperplane from the origin, and ξ_i are slack variables allowing for a margin of tolerance. The parameter ν controls the trade-off between the margin size and the proportion of outliers. SGD iteratively updates the model parameters (w, ξ, and ρ) based on a subset of the training data, significantly reducing computational complexity for large datasets. The update rules for w and ρ at each iteration t are given by:

- Update w: $w_{t+1} = w_t - \eta_t \nabla_w L(w_t, x_{i_t}, \xi_{i_t}, \rho_t)$
- Update ρ: Adjust ρ_t based on specific conditions derived from the dual formulation constraints.

where η_t is the learning rate at iteration t, L is the loss function derived from the primal objective, and i_t denotes the index of the selected sample at iteration t. The loss function for One-Class SVM with SGD was chosen based on the Hinge loss variant, suitable for the one-class setting given in (4):

$$L(w, x_i, \xi_i, \rho) = \max(0, \rho - (w \cdot \phi(x_i)) + \xi_i) \tag{4}$$

The gradients of L with respect to w and ρ are used to perform the updates. While traditional One-Class SVM formulations benefit from the kernel trick, directly applying it with SGD requires careful consideration. Approximate kernel method (RBF) was used to maintain computational efficiency. The selection of ν, learning rate η, and batch size for SGD were critical to achieving a balance between detection performance and computational efficiency in the study.

Robust Covariance (Elliptic Envelope)

The Robust Covariance method was adopted to model the synthetic simulated data as an ellipse, assuming that the regular (non-outlying) data points follow a Gaussian distribution. Anomalies are identified as those points that lie outside the ellipse, effectively capturing the multivariate outlier detection problem [26].

Let $X = \{x_1, x_2, \ldots, x_n\}$ be the synthetic simulated dataset consisting of n observations of d-dimensional real vectors, $x_i \in \mathbb{R}^d$. The goal is to estimate the parameters of the underlying multivariate Gaussian distribution in a robust way, that is, without being unduly influenced by outliers. The core of the Robust Covariance method is the robust estimation of the distribution's location (mean) and covariance matrix. This can be achieved using the Minimum Covariance Determinant (MCD) estimator [27]. The MCD estimator seeks to find the subset of observations $H \subset X$ with size $h > n/2$ that minimizes the determinant of the covariance matrix of the observations in H.

Let μ_{robust} and Σ_{robust} denote the robust estimates of the location and covariance matrix, respectively. μ_{robust} and Σ_{robust} are computed from the subset H by (5) and (6) as:

- Robust Location (Mean):

$$\mu_{\text{robust}} = \frac{1}{h} \sum_{x_i \in H} x_i, \quad \forall i = 1, \ldots, n \tag{5}$$

-

$$\text{Robust Covariance: } \Sigma_{\text{robust}} = \frac{1}{h-1} \sum_{x_i \in H} (x_i - \mu_{\text{robust}})(x_i - \mu_{\text{robust}})^T, \quad \forall i = 1, \ldots, n \tag{6}$$

An observation x_i is considered an outlier if its Mahalanobis distance to μ_{robust} with respect to Σ_{robust} is too large, i.e., it exceeds a predefined threshold $\chi^2_{d,\alpha}$, where α is the significance level and d is the dimensionality of the data. The Mahalanobis distance for a point x_i is given in (7) by:

$$\text{MD}(x_i) = \sqrt{(x_i - \mu_{\text{robust}})^T \Sigma^{-1}_{\text{robust}} (x_i - \mu_{\text{robust}})} \tag{7}$$

The threshold $\chi^2_{d,\alpha}$ is derived from the χ^2 distribution with d degrees of freedom at the α significance level, effectively determining the "elliptic envelope". The selection of h, the number of observations used for the MCD estimate, is crucial. A common choice is $h \approx [0.5n, 0.75n]$, balancing robustness and efficiency.

Local Outlier Factor (LOF)

Local Outlier Factor (LOF) introduces a density-based approach, focusing on the local density deviation of a point with respect to its neighbors [28]. It calculates the local density of each data point and compares it to the densities of its neighbors, identifying points that have a significantly lower density as outliers [29]. Let $X = \{x_1, x_2, \ldots, x_n\}$ be the synthetic simulated dataset consisting of n observations of d-dimensional real vectors, $x_i \in \mathbb{R}^d$.. The LOF algorithm involves several key steps and definitions:

1. k-distance

For each point x_i, the k-distance, denoted as $\delta_k(x_i)$, is defined as the distance of x_i from its kth nearest neighbor. This metric helps in identifying the immediate neighborhood of x_i.

2. Reachability Distance

The reachability distance of x_i with respect to x_j, denoted as $\text{rd}_k(x_i, x_j)$, is defined in (8) as:

$$\text{rd}_k(x_i, x_j) = \max\{\delta_k(x_j), \text{dist}(x_i, x_j)\} \tag{8}$$

where $\text{dist}(x_i, x_j)$ is the Euclidean distance between x_i and x_j.

3. Local Reachability Density (LRD)

The LRD of a point x_i, denoted as $\text{lrd}_k(x_i)$, quantifies the density of an area by considering the reachability distances of x_i to its neighbors. It is inversely proportional to the average reachability distance of x_i to its neighbors given in (9):

$$\text{lrd}_k(x_i) = \frac{1}{\frac{\sum_{x_j \in N_k(x_i)} \text{rd}_k(x_i, x_j)}{|N_k(x_i)|}} \tag{9}$$

Where $N_k(x_i)$ denotes the set of k nearest neighbors of x_i.

The LOF of x_i, denoted as $\text{LOF}_k(x_i)$, is then calculated by comparing the LRD of x_i with the LRDs of its neighbors is given in (10) by:

$$\text{LOF}_k(x_i) = \frac{\sum_{x_j \in N_k(x_i)} \frac{\text{lrd}_k(x_j)}{\text{lrd}_k(x_i)}}{|N_k(x_i)|} \tag{10}$$

A $\text{LOF}_k(x_i)$ significantly greater than 1 indicates that x_i is an anomaly, as its density is considerably lower than that of its neighbors. The selection of k (the number of neighbors considered) is crucial and can significantly influence the detection of outliers. A small k makes the algorithm sensitive to local outliers, whereas a larger k may capture the global context better but can miss local anomalies. For this study $k = 20$ was used.

Isolation Forest (iForest)

Isolation Forest diverges from the density or boundary-based methodologies by employing an isolation mechanism. iForest exploits the fact that anomalies are few and different, which makes them easier to isolate [30]. Using a forest of random trees, iForest recursively partitions the feature space, with the number of splits required to isolate a sample serving as an indicator of its anomaly score [14]. This approach is inherently efficient for high-dimensional data, as it does not require distance or density calculations, making it a highly effective and scalable option for anomaly detection.

For the purpose of this study, let $X = \{x_1, x_2, \ldots, x_n\}$ be the synthetic simulated dataset with n samples, where each $x_i \in \mathbb{R}^d$. The iForest algorithm was adopted to constructs a forest of binary trees. For each tree, a random subset of X was selected and recursively partitioned by randomly selecting a feature and then randomly selecting a split value between the minimum and maximum values

of the selected feature. The path length $h(x)$ of a data point x in a tree is the number of edges x traverses from the root node to the terminal node. The anomaly score of a data point is based on the average path length $h(x)$ over all trees in the forest is given in (11) by:

$$s(x,n) = 2^{-\frac{E[h(x)]}{c(n)}} \tag{11}$$

where $E[h(x)]$ is the average path length of x over all trees in the forest, n is the number of external nodes, and $c(n)$ is the average path length of unsuccessful search in a Binary Search Tree (BST) given in (12) by:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n} \tag{12}$$

with $H(i)$ being the harmonic number approximated by $\ln(i) + 0.5772156649$ (Euler's constant). The anomaly score $s(x,n)$ was used to determine if a data point is an anomaly or not. A score close to 1 indicates an anomaly, whereas a score much smaller than 0.5 indicates a normal observation. In this study, a threshold of 0.5 was set to distinguish between anomalies and normal observations. The number of trees in the forest, typically denoted as T, is a critical parameter that affects both the performance and the computational cost of the algorithm. The size of the data subset used for building each tree affects the algorithm's ability to isolate anomalies. A smaller size leads to more isolation but can also increase variance.

Model evaluation metrics

In this study, various metrics were utilized to evaluate the performance of the five (5) unsupervised machine learning algorithms. These metrics include Accuracy, Precision (specifically for outliers), Recall (specifically for outliers), and the F1 Score computed by (13), (14), (15) and (16). Below, we define each metric mathematically and explain their meanings in the context of anomaly detection.

1.
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$
Accuracy measures the proportion of true results (both true positives TP and true negatives TN) among the total number of cases examined [31]. It is a measure of the overall performance of the model across both classes (normal and anomalies).
TP (True Positives): The number of outliers correctly identified as outliers.
TN (True Negatives): The number of normal observations correctly identified as normal.
FP (False Positives): The number of normal observations incorrectly identified as outliers.
FN (False Negatives): The number of outliers incorrectly identified as normal observations.
2.
$$\text{Precision} = \frac{TP}{TP + FP} \tag{14}$$
Precision for outliers is the ratio of correctly predicted positive observations (true positives) to the total predicted positives (the sum of true positives and false positives). This metric specifically focuses on the model's ability to not label a normal observation as an outlier.
3.
$$\text{Recall} = \frac{TP}{TP + FN} \tag{15}$$
Recall for outliers is the ratio of correctly predicted positive observations (true positives) to all observations in the actual class (the sum of true positives and false negatives). This metric evaluates the model's ability to identify all actual outliers.
4.
$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$
The F1 Score is the weighted average of Precision and Recall. This score is particularly useful because it takes both false positives and false negatives into account. It is a measure of the model's accuracy and is higher when both precision and recall are high. The F1 Score is especially useful for situations where an even balance between precision and recall is desired.

Results and discussion

Synthetic simulated data

Fig. 2 shows the synthetic dataset we simulated, consisting of normal data points (in blue) and outliers (in red). This synthetic dataset is designed to test the predictive power (accuracy, precision, recall and F1-score) of the five unsupervised machine learning algorithms for anomaly detection.

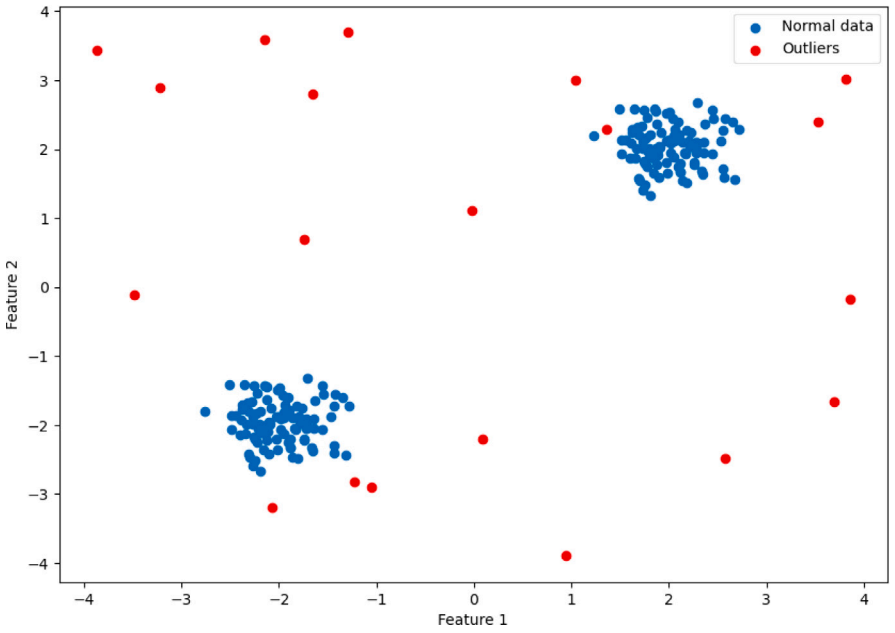


Fig. 2. Synthetic data with outliers. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

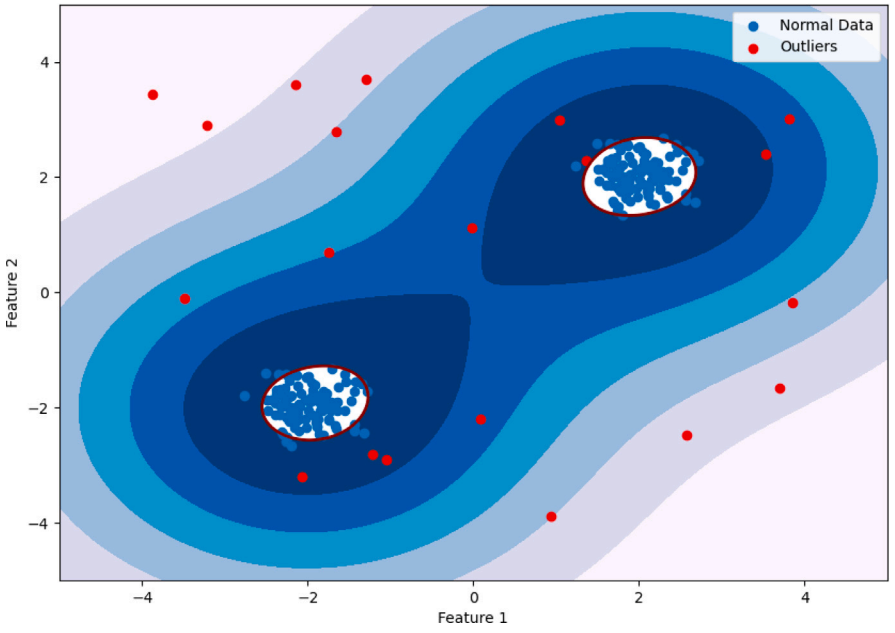


Fig. 3. Anomaly detection with One-Class SVM. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Analysis of One-Class SVM anomaly algorithm

In Fig. 3, the One-Class SVM model is applied to the synthetic simulated data. The model was trained on the normal data (blue points), and then it predicted which data points are outliers. The dark red contour circle delineates the boundary between normal data points and anomalies as determined by the model. Points outside this boundary are considered anomalies by the model. As seen, the One-Class SVM successfully identified the majority of the outliers (red points) - almost all of them while correctly classifying most of the normal data points. This demonstrates the model's ability to distinguish between normal data and anomalies in a dataset, making it a useful tool for anomaly detection tasks.

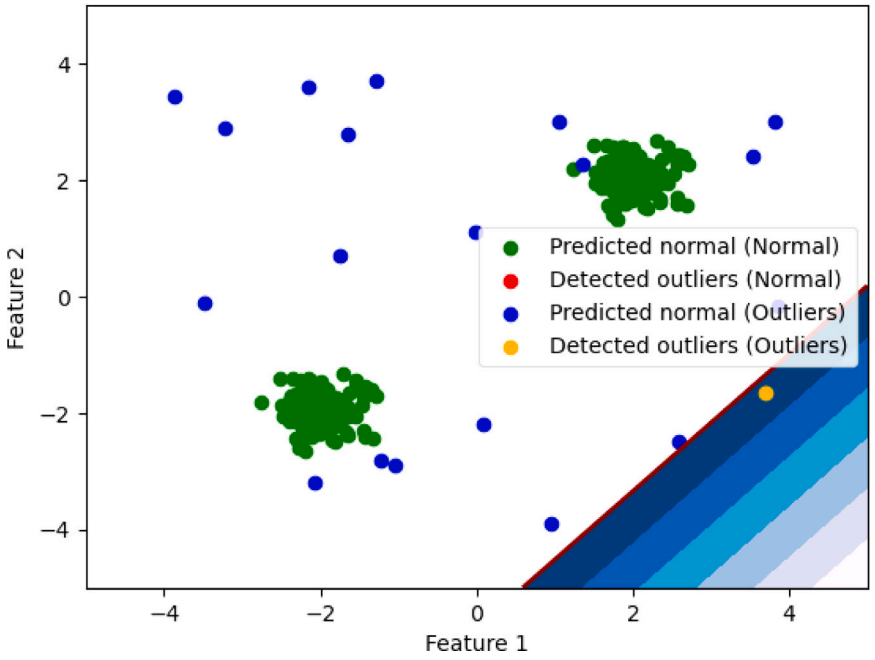


Fig. 4. One-Class SVM using SGD decision function and predicted outliers.

In the subsequent sections, "Predicted normal (Normal)" represents an actual normal observation that is predicted to be a normal observation by the model. Likewise "Detected outliers (Normal)" is an actual normal observation that is detected as an outlier by the model. Similar interpretations are used for all the labels that follows this format.

Analysis of One-Class SVM with SGD anomaly algorithm

Fig. 4 illustrates the results of anomaly detection using a One-Class SVM with SGD on the same synthetically simulated dataset. Similar to the previous analysis with the traditional One-Class SVM, this approach aims to identify outliers from the normal data points in the feature space. The decision function's boundary, delineated by the dark red contour line, represents the threshold beyond which data points are considered anomalies by the model. The contour plot background provides a gradient of how decision values change across the feature space, giving visual feedback on the margin of the classification. As observed, the One-Class SVM with SGD effectively separates most of the outliers from the normal data points, demonstrating its capability in anomaly detection. This approach, leveraging SGD, offers computational advantages, especially for large datasets, by iteratively updating the model's parameters without the need for the entire dataset to be in memory. The results affirm the efficacy of the One-Class SVM with SGD in distinguishing between normal observations and anomalies, highlighting its utility in scenarios where efficient computation is paramount. The application of this method showcases its potential in real-world anomaly detection tasks, providing a viable option for handling large-scale data with the need for efficient processing.

Analysis of Isolation Forest anomaly algorithm

Fig. 5 displays the results of anomaly detection using an iForest on our synthetically simulated dataset. This method operates on a different principle compared to the One-Class SVM; it isolates anomalies instead of creating a boundary to separate them from normal observations. The iForest model effectively identified many of the synthetic outliers as anomalies, demonstrating its capability to isolate abnormal observations from normal data. This method's strength lies in its ability to handle multi-dimensional data and its efficiency in detecting anomalies without the need to specify a contamination rate explicitly, although a rough estimate can help in tuning the model.

Analysis of Local Outlier Factor (LOF) anomaly algorithm

Fig. 6 illustrates the application of the LOF method for anomaly detection on the synthetically simulated dataset. This method assesses the local density deviation of a given data point with respect to its neighbors, aiming to identify regions of similar density and highlight points that stand out as anomalies. The LOF method effectively identifies outliers by focusing on the local neighborhood of each data point. This technique is particularly useful in datasets where the density around normal observations and anomalies

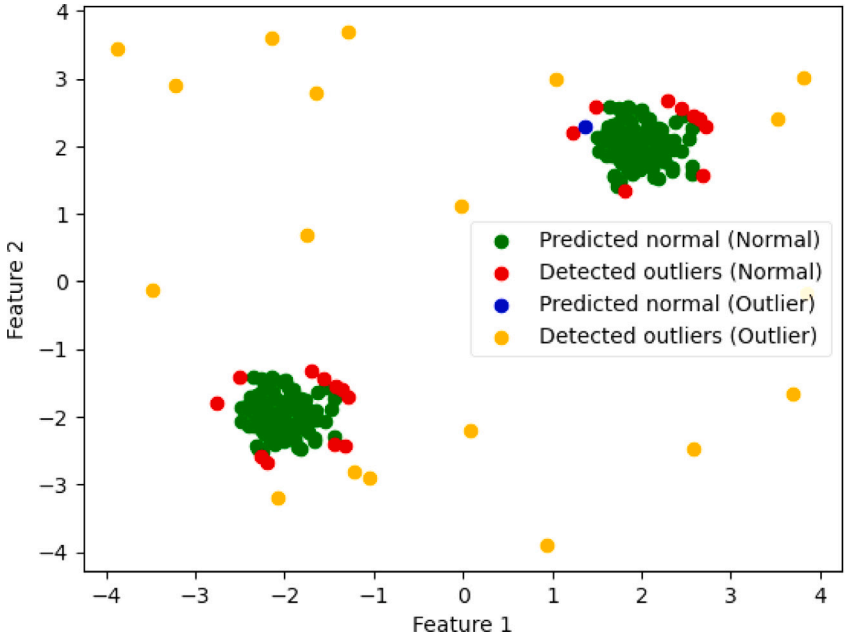


Fig. 5. Isolation Forest Predicted normal points and outliers.

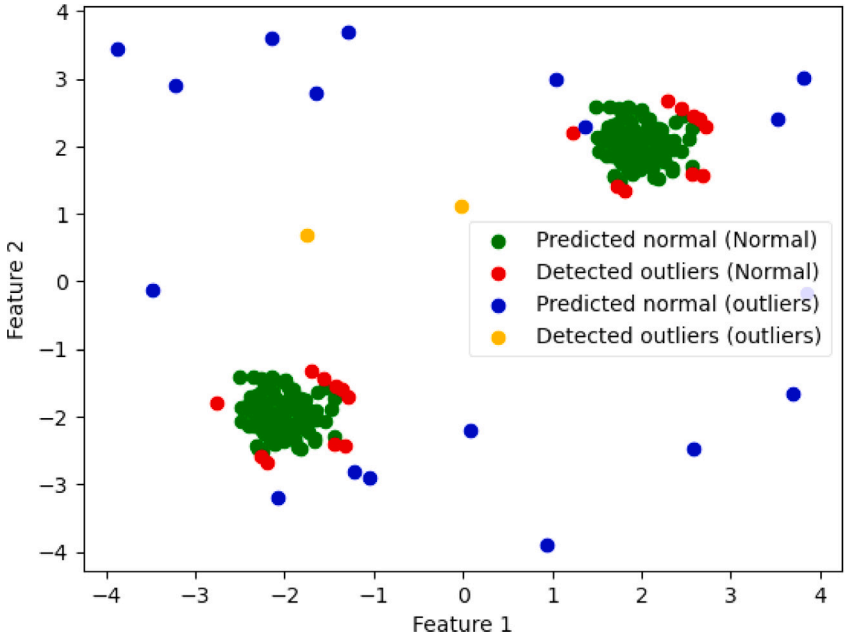


Fig. 6. Local Outlier Factor (LOF) Predicted normal points and outliers.

differs significantly. As shown, the LOF model has successfully flagged minimal synthetic outliers as outliers. LOF is advantageous in scenarios where the anomaly pattern is not globally uniform but varies across different regions of the dataset. This makes it a versatile tool for anomaly detection in complex datasets where anomalies may not be detectable through global density or distribution-based methods. The results showcase the importance of LOF in identifying outliers, highlighting its potential for better anomaly detection tasks

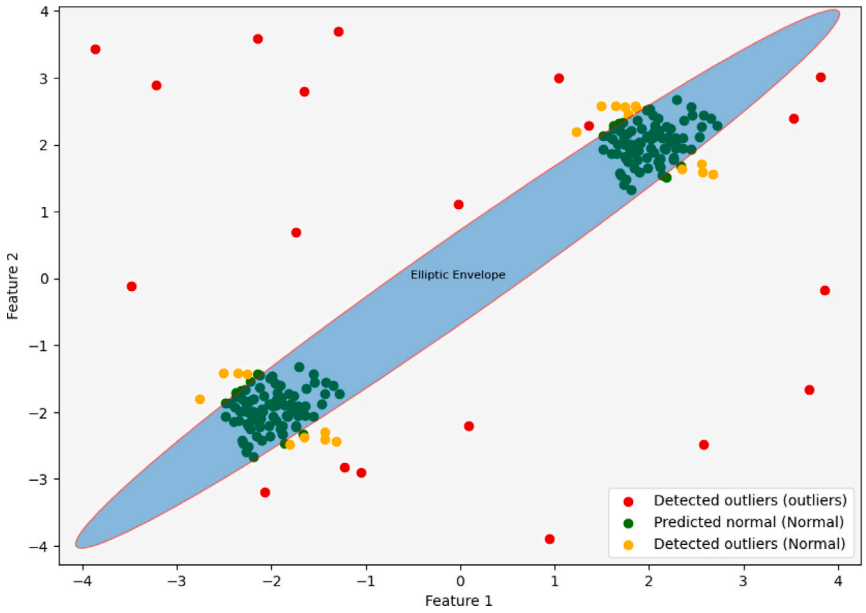


Fig. 7. Anomaly detection with Robust Covariance (Elliptic Envelope). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Analysis of Robust covariance anomaly algorithm

Fig. 7 demonstrates the results of anomaly detection using Robust Covariance, also known as the Elliptic Envelope method, on our synthetically simulated dataset. This approach assumes that the normal data points follow a Gaussian distribution and attempts to enclose the majority of these points within an ellipse, identifying points lying outside as anomalies. The Elliptic Envelope method effectively identifies a significant number of the synthetic outliers, showcasing its capability to detect anomalies based on the deviation from the assumed Gaussian distribution of the dataset.

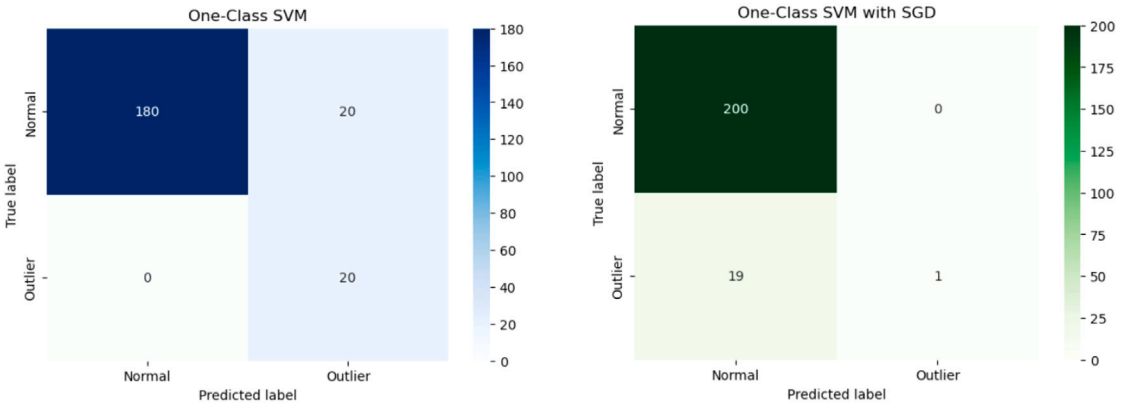
This method is particularly useful in applications where the data is expected to exhibit a "normal" distribution, allowing for the detection of outliers as deviations from this model. Robust Covariance is advantageous for its sensitivity to the shape of the data distribution, making it suitable for datasets where the normal observations are densely clustered. The results highlight the role of the Robust Covariance method in identifying outliers, underlining its utility in scenarios where a robust estimation of the data's distribution is critical for anomaly detection. In Fig. 7, it is however to be noted that "Predicted normal (Normal)" represents all actual normal observation that are predicted to be normal observations by the model and are enclosed by the elliptic envelope. This excludes green data points that lies outside the elliptic envelope.

Model performance evaluation

The One-Class SVM and Robust Covariance models have perfectly identified all true outliers (100% recall) but have also misclassified 20 normal points as outliers (lower precision). The One-Class SVM with SGD has the highest number of false negatives, meaning it failed to identify most of the true outliers (lower recall), although it did not misclassify any normal points as outliers (high precision). The Isolation Forest model has a good balance, with only one false negative and 19 true positives, indicating it is quite effective at identifying outliers while maintaining a reasonable false positive rate. The Local Outlier Factor model has the highest number of false negatives (lowest recall), indicating it is the least effective at identifying outliers among these models as represented by Fig. 8.

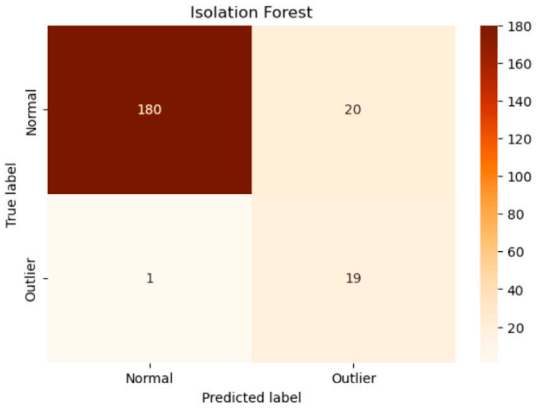
Table 1 and Fig. 9 summarize the performance metrics of the five unsupervised anomaly detection algorithms. The One-Class SVM and Robust Covariance models exhibit identical accuracy and precision, and both successfully recall all outliers, resulting in an F1 score of approximately 66.67%. Remarkably, the One-Class SVM with SGD achieves the highest accuracy at 91.36% and perfect precision, indicating it does not falsely label normal observations as outliers. However, it has the lowest recall of 5.00%, suggesting it fails to identify the majority of true outliers, which is also reflected in its low F1 score of 9.52%. The Isolation Forest model shows a balanced profile with decent accuracy (90.45%) and the second-highest recall (95.00%), leading to an F1 score of 64.41%. The Local Outlier Factor model struggles in this comparison, with the lowest accuracy (82.73%) and F1 score (9.52%), alongside a recall penultimate to One-Class SVM with SGD.

One-Class SVM and Robust Covariance show a balanced performance with equal recall and F1 scores (100.00% and 66.67% respectively), indicating robust performance in detecting true outliers. However, their precision is moderate, suggesting some normal

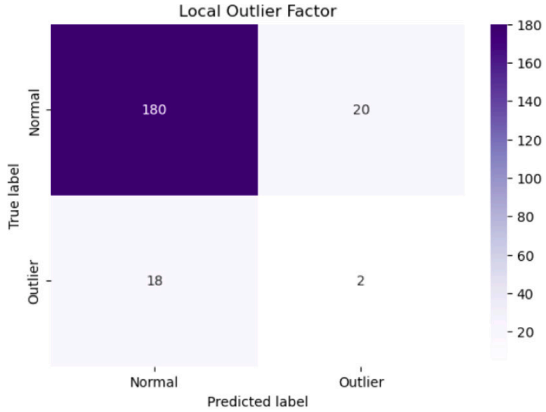


(a) One Class SVM

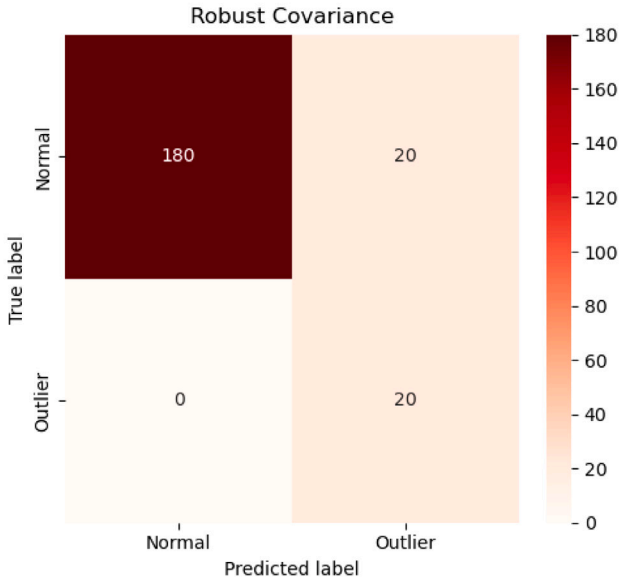
(b) One Class SVM with SGD



(c) Isolation Forest



(d) Local Outlier Factor



(e) Robust Covariance

Fig. 8. Confusion matrices of the 5 anomaly detection algorithms.

Table 1
Model performance of the unsupervised anomaly detection algorithms.

Model	Accuracy	Precision (Outliers)	Recall (Outliers)	F1 score (Outliers)
One-Class SVM	90.91%	50.00%	100.00%	66.67%
One-Class SVM with SGD	91.36%	100.00%	5.00%	9.52%
Isolation Forest	90.45%	48.72%	95.00%	64.41%
Local Outlier Factor	82.73%	9.09%	10.00%	9.52%
Robust Covariance	90.91%	50.00%	100.00%	66.67%

Table 2
Computational efficiency of the unsupervised anomaly detection algorithms.

Model	Model fit time (s)	Outlier prediction time (s)	Results plot time (s
One-Class SVM	0.0020	0.0001	1.1252
One-Class SVM with SGD	0.0020	0.0001	0.4558
Isolation Forest	0.2036	0.0040	0.4048
Local Outlier Factor	0.0075	0.3012	0.0503
Robust Covariance	0.0602	0.0009	0.0140

points were incorrectly labeled as anomalies. One-Class SVM with SGD exhibits high precision (100.00%) but significantly low recall and F1 score (5.00% and 9.52% respectively), highlighting its conservative approach in labeling outliers; it misses many true outliers but is very accurate when it does label a point as an outlier. Isolation Forest provides a good balance between recall and precision, achieving an F1 score of 64.41%. This indicates a strong capability in identifying outliers while maintaining a reasonable rate of false positives. Local Outlier Factor demonstrates the lowest performance across all metrics, indicating difficulties in accurately distinguishing between normal points and outliers in this synthetic simulated dataset.

The evaluation reveals that One-Class SVM, Isolation Forest, and Robust Covariance are more effective in identifying outliers in the simulated dataset, with Isolation Forest slightly outperforming the others in terms of balancing precision and recall. One-Class SVM with SGD shows promise in precision but needs adjustment to improve recall. Local Outlier Factor may require parameter tuning or may not be as suitable for this particular dataset's characteristics.

Table 2 presents a comparative analysis of the computational efficiency of the five (5) different unsupervised anomaly detection algorithms. The One-Class SVM and One-Class SVM with SGD demonstrate the fastest model fitting and outlier prediction times, both clocking in at only 0.0020 s for model fitting and 0.0001 s for outlier prediction, suggesting high efficiency in simpler computational environments. Notably, both algorithms have drastically different times for plotting results, with One-Class SVM taking significantly longer (1.1252 s) compared to its SGD counterpart (0.4558 s). The Isolation Forest, while slower in model fitting (0.2036 s) and outlier prediction (0.0040 s), also requires less time for result visualization (0.4408 s). The Local Outlier Factor shows a modest model fitting time (0.0075 s) but has a relatively high outlier prediction time (0.3012 s), indicating a possible trade-off between fitting speed and prediction complexity. Robust Covariance, although it has the slowest model fitting time (0.0602 s), showcases efficient outlier prediction (0.0009 s) and the quickest results plotting (0.0140 s). This computational efficiency analysis highlights the varying performance of trade-offs between the five (5) different anomaly detection models in terms of computational cost and efficiency, providing valuable insights for selecting appropriate models based on specific application requirements.

The application of One-Class SVM in this study, noted for its high recall but moderate precision, aligns with findings from [16], where the algorithm demonstrated robustness in anomaly detection within tightly defined feature spaces. Similarly, our study's application of SGD in One-Class SVM aligns with [24], which highlighted the scalability and efficiency of SGD for large datasets. Isolation Forest's performance in our study is consistent with [14], emphasizing its strength in high-dimensional settings and its efficiency in isolating anomalies without extensive parameter tuning. This echoes the utility of Isolation Forest in handling complex data structures. However, the current study extends these findings by quantitatively assessing the balance between recall and precision, which has been less frequently addressed in the literature. Local Outlier Factor's lower performance in our study contrasts with its efficacy reported in [29], where LOF excelled in datasets with pronounced local density variations. This discrepancy could stem from differences in the dataset characteristics used in the two studies, underlining the sensitivity of LOF to the underlying data distribution. Robust Covariance's effectiveness in our study, especially in handling data assumed to follow a Gaussian distribution, supports the findings of [26]. However, the results of this study also highlight the limitations of this assumption, as real-world datasets often exhibit more complex distributions, potentially affecting the algorithm's performance outside controlled experimental conditions.

It is to be noted that the synthetic dataset, while not entirely representative of real-world data due to its simplified structure and assumptions as clearly specified in Section "Limitations of the study", still holds relevance in specific practical scenarios such as algorithm testing and benchmarking like this one presented in this study. This setting allows for controlled experimentation and parameter tuning in a known environment, facilitating scalability assessments and optimization of anomaly detection algorithms. However, its simplicity in feature interactions and absence of real-world data challenges like noise and heterogeneous features means that while useful for preliminary testing of algorithms, it is crucial to validate and refine these algorithms with real-world data to ensure their practical predictive power and robustness.

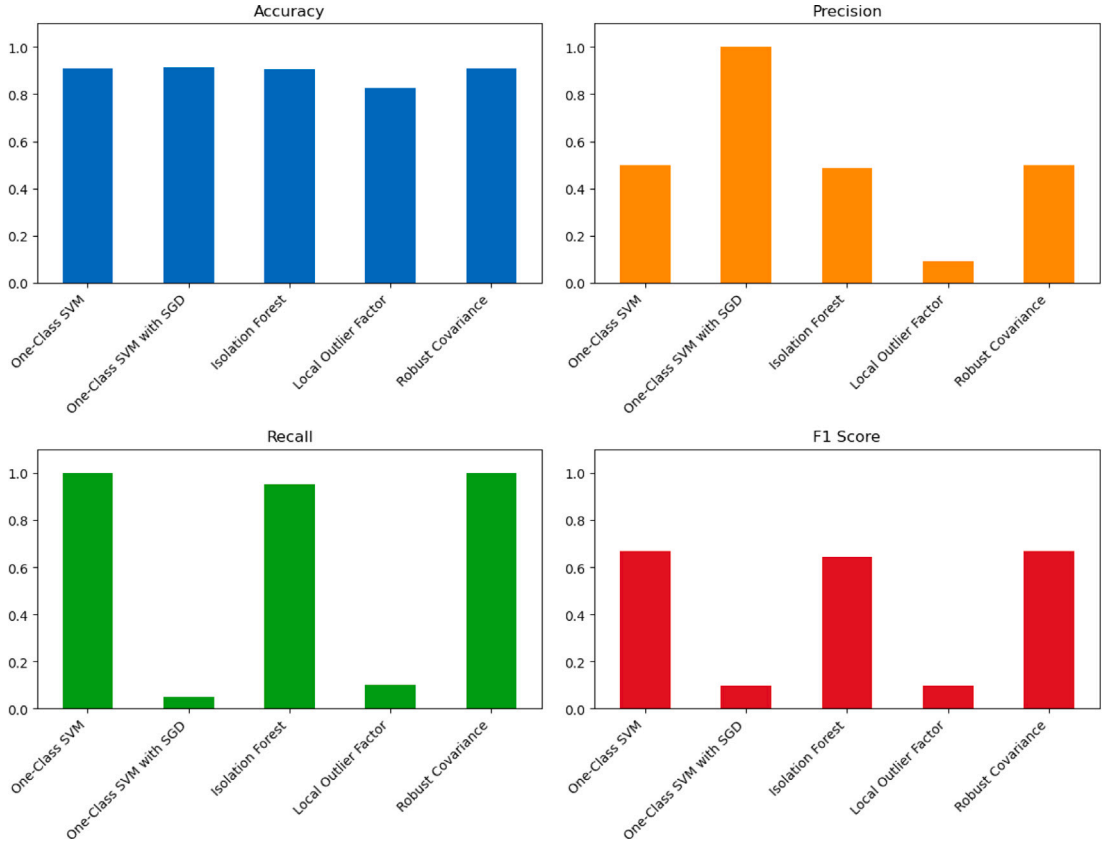


Fig. 9. Model performance metrics.

Conclusion & recommendation of the study

This study evaluates the performance of five unsupervised machine learning anomaly detection algorithms: One-Class SVM, One-Class SVM with Stochastic Gradient Descent (SGD), Isolation Forest, Local Outlier Factor (LOF), and Robust Covariance (Elliptic Envelope) on a synthetic simulated data. The evaluation is based on accuracy, precision, recall, and F1 score, focusing on the algorithms' ability to correctly identify anomalies (outliers) in a synthetically simulated dataset. The comparative analysis of the five unsupervised machine learning anomaly detection algorithms provide insights into their performance and applicability across various anomaly detection tasks. The evaluation based on accuracy, precision, recall, and F1 score highlights the diverse capabilities of these algorithms in identifying outliers, each influenced by the underlying data distribution, algorithm complexity, and parameter selection. One-Class SVM and Robust Covariance demonstrated similar performance metrics, excelling in recall by correctly identifying a high percentage of true outliers. This high recall indicates their efficacy in detecting anomalies within the dataset. However, their moderate precision suggests a propensity to falsely label normal points as anomalies. These algorithms are well-suited for applications where missing an outlier is costlier than false alarms, such as fraud detection or preventive maintenance. Their performance can be significantly influenced by the choice of kernel (in One-Class SVM) and the assumption of a Gaussian distribution (in Robust Covariance), highlighting the importance of understanding data distribution before algorithm application.

One-Class SVM with SGD presented an interesting trade-off with high precision but low recall, indicating it is highly accurate in its anomaly predictions but misses a substantial number of outliers. This characteristic makes it suitable for scenarios where false positives are a greater concern than missed detections, such as in certain security applications where false alarms must be minimized [32]. The algorithm's efficiency, driven by SGD, offers scalability and speed, which are critical in large-scale or real-time processing environments. Isolation Forest showed a balanced performance with relatively high precision and recall, suggesting it effectively identifies outliers without excessively mislabeling normal points. Its unique approach of isolating anomalies makes it less sensitive to the specific data distribution, enabling it to perform well across a variety of datasets. This algorithm is particularly advantageous in high-dimensional data or for applications requiring quick anomaly detection without extensive parameter tuning [33]. Local Outlier Factor (LOF), despite its lower overall accuracy, provides valuable insights into local anomalies, which might not be detected by other algorithms focusing on global outliers. Its lower performance in this analysis could be attributed to its sensitivity to parameter settings and the neighborhood size, which significantly impacts its efficacy. LOF's advantage

lies in its ability to detect anomalies in datasets with varying densities, making it ideal for applications like intrusion detection or identifying rare events in spatial data [34].

The analysis highlights several critical factors influencing the efficacy of anomaly detection. Firstly, the assumption about data distribution (e.g., Gaussian in Robust Covariance) can significantly impact algorithm performance. Algorithms like Isolation Forest, which do not rely on such assumptions, offer more flexibility across different datasets. Secondly, the complexity of an algorithm and its sensitivity to parameter settings can affect both its performance and practical applicability. While complex models may offer higher accuracy, they often require extensive tuning and computational resources. Last but not least, the choice of anomaly detection algorithm should be guided by the specific requirements of the application domain, including the cost of false positives versus false negatives, data dimensionality, and the need for real-time processing. The study has shown that the selection of an anomaly detection algorithm should be a considered decision, taking into account the specific characteristics of the data and the operational context of the application. By understanding the advantages and disadvantages of each method, practitioners can better tailor their approach to the unique challenges of anomaly detection, optimizing performance and minimizing the risk of critical oversights. Future work should explore parameter optimization, the impact of dataset characteristics on model performance, and the application of these models to real-world datasets to validate their efficacy in practical anomaly detection scenarios. The study thus recommended the application of mathematical numerical simulation techniques such as those used by [35–38].

Limitations of the study

The synthetic simulated dataset employed for the anomaly detection assumes homogeneity of features (in this case, continuous numerical data), specific data distributions (normal for 'normal' data and uniform for outliers), and a clear distinction between normal data points and outliers. These assumptions simplify the complex and varied nature of real-world data, which often involves heterogeneous feature types, diverse distributions, and subtler distinctions between normal and anomalous data. The primary limitations of the synthetic dataset include the absence of feature interactions, noise, and correlations, as well as the lack of scale and density variations. It consists of only two features, which fails to capture the complexities of multi-dimensional real-world datasets, potentially affecting the evaluation of algorithms designed to handle more intricate data structures. These assumptions and limitations can significantly affect the results, potentially leading to an overestimation of an algorithm's performance when generalized to real-world data. The dataset's simplicity and controlled environment might favor certain models that perform well under these specific conditions but might not necessarily offer the same performance against more complex, noisy, and subtly anomalous real-world data, impacting both the sensitivity and specificity of the algorithms.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The author acknowledges the enormous support of the University of Texas Rio Grande Valley (UTRGV) Presidential Research Fellowship fund.

References

[1] Srikanth Thudumu, Philip Branch, Jiong Jin, Jugdutt Singh, A comprehensive survey of anomaly detection techniques for high dimensional big data, J. Big Data 7 (2020) 1–30.
[2] ANM Bazlur Rashid, Mohiuddin Ahmed, Leslie F Sikos, Paul Haskell-Dowland, Anomaly detection in cybersecurity datasets via cooperative co-evolution-based feature selection, ACM Trans. Manage. Inf. Syst. (TMIS) 13 (3) (2022) 1–39.
[3] Yan-Feng Zhang, Hong-Liang Lu, Hong-Fan Lin, Xue-Chen Qiao, Hao Zheng, et al., The optimized anomaly detection models based on an approach of dealing with imbalanced dataset for credit card fraud detection, Mob. Inf. Syst. 2022 (2022).
[4] Ezekiel Nii Noi Nortey, Edmund Fosu Agyemang, Richard Minkah, Kwame Asah-Asante, Bayesian estimation of presidential elections in ghana: A validation approach, Afr. J. Appl. Stat. 9 (1) (2022) 1297–1317.
[5] Edmund Fosu Agyemang, Ezekiel NN Nortey, Richard Minkah, Kwame Asah-Asante, The unfolding mystery of the numbers: First and second digits based comparative tests and its application to Ghana's elections, Model Assist. Stat. Appl. 18 (2) (2023) 183–192.
[6] Edmund F Agyemang, Ezekiel NN Nortey, Richard Minkah, Kwame Asah-Asante, Baseline comparative analysis and review of election forensics: Application to Ghana's 2012 and 2020 presidential elections, Heliyon (2023).
[7] Soumya Ranjan Mishra, Hitesh Mohapatra, et al., Enhancing money laundering detection through machine learning: A comparative study of algorithms and feature selection techniques, in: AI and Blockchain Applications in Industrial Robotics, IGI Global, 2024, pp. 300–321.
[8] Sohrab Mokhtari, Alireza Abbaspour, Kang K Yen, Arman Sargolzaei, A machine learning approach for anomaly detection in industrial control systems based on measurement data, Electronics 10 (4) (2021) 407.
[9] Jiawei Yang, Susanto Rahardja, Pasi Fränti, Outlier detection: how to threshold outlier scores? in: Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing, 2019, pp. 1–6.
[10] Peter J. Rousseeuw, Mia Hubert, Anomaly detection by robust statistics, Wiley Interdiscip. Rev.: Data Min. Knowl. Discov. 8 (2) (2018) e1236.

[11] Nico Görnitz, Marius Kloft, Konrad Rieck, Ulf Brefeld, Toward supervised anomaly detection, J. Artificial Intelligence Res. 46 (2013) 235–262.
[12] Watson Jia, Raj Mani Shukla, Shamik Sengupta, Anomaly detection using supervised learning and multiple statistical methods, in: 2019 18th IEEE International Conference on Machine Learning and Applications, ICMLA, IEEE, 2019, pp. 1291–1297.
[13] Mahdi Rezapour, Anomaly detection using unsupervised methods: credit card fraud case study, Int. J. Adv. Comput. Sci. Appl. 10 (11) (2019).
[14] Fei Tony Liu, Kai Ming Ting, Zhi-Hua Zhou, Isolation-based anomaly detection, ACM Trans. Knowl. Discov. Data (TKDD) 6 (1) (2012) 1–39.
[15] Ankit Kumar, Abhishek Kumar, Ali Kashif Bashir, Mamoon Rashid, VD Ambeth Kumar, Rupak Kharel, Distance based pattern driven mining for outlier detection in high dimensional big dataset, ACM Trans. Manage. Inf. Syst. (TMIS) 13 (1) (2021) 1–17.
[16] Rui Zhang, Shaoyan Zhang, Yang Lan, Jianmin Jiang, Network anomaly detection using one class support vector machine, in: Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol. 1, 2008.
[17] Chin-Shiuh Shieh, Thanh-Tuan Nguyen, Chun-Yueh Chen, Mong-Fong Horng, Detection of unknown DDoS attack using reconstruct error and one-class SVM featuring stochastic gradient descent, Mathematics 11 (1) (2022) 108.
[18] Julien Lesouple, Cédric Baudoin, Marc Spigai, Jean-Yves Tourneret, Generalized isolation forest for anomaly detection, Pattern Recognit. Lett. 149 (2021) 109–119.
[19] Marwan Omar, Malware anomaly detection using local outlier factor technique, in: Machine Learning for Cybersecurity: Innovative Deep Learning Solutions, Springer, 2022, pp. 37–48.
[20] Gregorius Airlangga, Analysis of machine learning algorithms for seismic anomaly detection in indonesia: Unveiling patterns in the pacific ring of fire, J. Lebesgue: J. Ilmiah Pendidikan Mat. Mat. Stat. 5 (1) (2024) 37–48.
[21] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, Sal Stolfo, A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data, in: Applications of Data Mining in Computer Security, Springer, 2002, pp. 77–101.
[22] Siqi Wang, Qiang Liu, En Zhu, Fatih Porikli, Jianping Yin, Hyperparameter selection of one-class support vector machine by self-adaptive data shifting, Pattern Recognit. 74 (2018) 198–211.
[23] Yan Qiao, Kui Wu, Peng Jin, Efficient anomaly detection for high-dimensional sensing data with one-class support vector machine, IEEE Trans. Knowl. Data Eng. 35 (1) (2021) 404–417.
[24] Naeem Seliya, Azadeh Abdollah Zadeh, Taghi M. Khoshgoftaar, A literature review on one-class classification and its potential applications in big data, J. Big Data 8 (1) (2021) 1–31.
[25] Yi Liu, Sahil Garg, Jiangtian Nie, Yang Zhang, Zehui Xiong, Jiawen Kang, M Shamim Hossain, Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach, IEEE Internet Things J. 8 (8) (2020) 6348–6358.
[26] Peter Filzmoser, Valentin Todorov, Review of robust multivariate statistical methods in high dimension, Anal. Chim. Acta 705 (1–2) (2011) 2–14.
[27] Mia Hubert, Michiel Debruyne, Minimum covariance determinant, Wiley Interdiscip. Rev.: Comput. Stat. 2 (1) (2010) 36–43.
[28] Omar Alghushairy, Raed Alsini, Terence Soule, Xiaogang Ma, A review of local outlier factor algorithms for outlier detection in big data streams, Big Data Cogn. Comput. 5 (1) (2020) 1.
[29] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, Jörg Sander, LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104.
[30] Hui Liu, Bo Zhao, Jiabao Guo, Kehuan Zhang, Peng Liu, A lightweight unsupervised adversarial detector based on autoencoder and isolation forest, Pattern Recognit. 147 (2024) 110127.
[31] Chaman Verma, Zoltán Illés, Deepak Kumar, An investigation of novel features for predicting student happiness in hybrid learning platforms–An exploration using experiments on trace data, Int. J. Inf. Manag. Data Insights 4 (1) (2024) 100219.
[32] Dongqi Han, Zhiliang Wang, Wenqi Chen, Ying Zhong, Su Wang, Han Zhang, Jiahai Yang, Xingang Shi, Xia Yin, Deepaid: Interpreting and improving deep learning-based anomaly detection in security applications, in: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 3197–3217.
[33] Michael Heigl, Kumar Ashutosh Anand, Andreas Urmann, Dalibor Fiala, Martin Schramm, Robert Hable, On the improvement of the isolation forest algorithm for outlier detection with streaming data, Electronics 10 (13) (2021) 1534.
[34] Sujeong Kim, Chanwoong Hwang, Taejin Lee, Anomaly based unknown intrusion detection in endpoint environments, Electronics 9 (6) (2020) 1022.
[35] Omar Abu Arqub, Zaer Abo-Hammour, Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm, Inf. Sci. 279 (2014) 396–415.
[36] Haneen Badawi, Omar Abu Arqub, Nabil Shawagfeh, Stochastic integrodifferential models of fractional orders and Leffler nonsingular kernels: well-posedness theoretical results and Legendre Gauss spectral collocation approximations, Chaos Solitons Fractals 10 (2023) 100091.
[37] Haneen Badawi, Omar Abu Arqub, Nabil Shawagfeh, Well-posedness and numerical simulations employing Legendre-shifted spectral approach for Caputo–Fabrizio fractional stochastic integrodifferential equations, Internat. J. Modern Phys. C 34 (06) (2023) 2350070.
[38] Haneen Badawi, Nabil Shawagfeh, Omar Abu Arqub, Fractional conformable stochastic integrodifferential equations: existence, uniqueness, and numerical simulations utilizing the shifted Legendre spectral collocation algorithm, Math. Probl. Eng. 2022 (1) (2022) 5104350.