ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2023.2.237

Detecting Outliers Using Modified Recursive PCA Algorithm For Dynamic Streaming Data

Yasi Dani[1,✉], **Agus Yodi Gunawan**[1,✉], **Masayu Leylia Khodra**[2,4], **Sapto Wahyu Indratno**[3,4]

[1]Industrial and Financial Mathematics Research Group, Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, Jl. Ganesha 10 Bandung, Indonesia
[2]School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganesha 10 Bandung, Indonesia
[3]Department Statistics Research Division, Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, Jl. Ganesha 10 Bandung, Indonesia
[4]Natural Language Processing and Big Data Analytics (U-CoE AI-VLB), University Center of Excellence on Artificial Intelligence for Vision, Institut Teknologi Bandung, Jl. Ganesha 10 Bandung, Indonesia
30119012@mahasiswa.itb.ac.id✉, ayodi@itb.ac.id✉, masayu@staff.stei.itb.ac.id, sapto@itb.ac.id

Abstract

Outlier analysis has been widely studied and has produced many methods. However, there is still rare a method to detect outliers for dynamically streaming batch data (online learning). In the present research, a novel online algorithm to detect outliers in such dataset is proposed. Data points are proceeded by applying a modified recursive PCA to predict sequentially parameters of the model; eigenvalues and eigenvectors of the statistical detection model are recursively updated using approximate values by perturbation methods. More specifically, the recursive eigenstructure is obtained from the derivation of the covariance matrix using the first-order perturbation technique. The Mahalanobis distance is then used as an outlier score. Our algorithm performances are evaluated using some metrics, namely accuration, precision, recall, F1-score, AUC-PR, and the execution time. Results show that the proposed online outlier detection is computationally efficient in time and the algorithm's performance effectiveness is comparable to that of the offline outlier detection algorithm via classical PCA.

Keywords: *Outlier, Online Learning, Recursive PCA, Eigendecomposition, Perturbation Method.*

Received: 27 October 2023
Accepted: 30 November 2023
Online: 07 December 2023
Published: 20 December 2023

1 Introduction

An outlier is an observation that deviates so much from other observations that it raises the suspicion that it is produced by a different mechanism [15]. In detecting outliers, there are two outlier detection techniques, supervised outlier detection based on classification or regression and unsupervised outlier detection based on clustering. In the digital era, the introduction of a technology system is very important since we have entered an era, namely the era of big data. Big data is data on a large scale in terms of volume, intensity, and complexity that exceeds the capacity of standard analytical tools [27]. Emerson et al. [11] proposed that a data set will be considered large if it exceeds 20% of the RAM on a given machine and very large if it exceeds 50%, in which case even the simplest calculations will consume all the remaining RAM. Consequently, it would be time consuming to process the data using traditional methods. It is better to construct an appropriate algorithm so that the data can be managed properly and efficiently.

Online learning, also known as incremental learning, is a machine learning method that builds a learnable model for effective classification in real-time detection [18]. While, offline learning is a traditional machine learning technique that requires large computational time and time to process all data. The model uses only the previously provided data (a set of historical data). It will then require manually updating the model on more recent emerging data and apply the resulting model whenever the normal system behavior changes. The online algorithm could use all available information without storing or revisiting individual data points [8, 23]. Other previous studies developed several techniques in identifying outliers or anomalies. Bosman et al.[5] identified anomalies in sensor systems with parameter estimation using the recursive least square (RLS) method, Zangeneh-Nejad et al. [31] studied anomalies with the DIA (Detection, Identification, and Adaptation) algorithm after estimating parameters using the RLS method. Later, Schifano et al. [25] and Wang et al. [29] detected outliers with standardized predictive residuals and to test for outliers in the n-th data where after estimating the parameters with the Bayesian framework method. Hoeltgebaum et al. [17] identified anomalies using The Hall-Buckley Eggleson (HBE) method after estimating parameters using the LASSO method. Then, Ippel et al. [19] estimated the parameters recursively with stochastic gradient descent. And Majdoubi et al. [22] recursively estimated the parameters based on the recursive least squares method. Thuy et al. [28] provided a method consisting a deep neural network and heuristic algorithms combined with LR to boost the accuracy of attack detections in an intrusion detection system (IDS). Fieri and Suhartono [12] studied two types of soft voting models, namely machine learning-based and deep learning-based to detect offensive language a Twitter dataset and to improve the performance of the soft voting classifier method.

According to Jolliffe [20], Principal Component Analysis (PCA), is one of the oldest and most well-known multivariate analysis techniques. This PCA method is applied primarily to reduce the dimensions of the dataset by projecting each data point into the first few principal components to obtain data of less dimension while retaining as much as possible of the variation present in the dataset [3, 2]. The Mahalanobis distance is equal to the Euclidean distance between data point and in such a transformed (axis-rotated) dataset after dividing each of transformed coordinate values by the standard deviation of that direction. Thus, PCA can also be used to calculate the Mahalanobis distance [1].

In this research, we propose a novel online outlier detection algorithm to identify outliers with Mahalanobis distance using modified recursive PCA where outliers are identified as soon as a new data record appears in dataset. To be explicit, recusive eigenstructures are calculated from the covariance matrix using the first-order perturbation technique. For outlier detection score the Mahalanobis distance is applied. We then evaluate the effectiveness and efficiency of the algorithm performance evaluation using some metrics, i.e. accuration, precision, recall, F1-score, AUC-PR, and the execution time. We apply this online outlier detection algorithm on synthetic dataset.

The remainder of the paper is organized as follows: Section 2 gives recursive formulas for the sample mean and covariance matrix. It also presents approximate perturbation methods for recursive PCA, our proposed online detection algorithm, and algorithm performance evaluation using some metrics. In Section 3, we apply online and offline algorithm approaches to synthetic dataset and discuss the Mahalanobis score, and performance analysis for both approaches. Conclusions are written in Section 4.

2 Materials and Methods

In this section, we build an online outlier detection algorithm consisting of the online parameter estimation formula via modified recursive PCA and identifying the outliers in each new data arrive using the Mahalanobis distance as the outlier score.

2.1 Parameter Estimation via Modified Recursive PCA

We first derive a recursive mean formula. Given vector data $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{n+1} \in \Re^p$. A recursive mean formula is calculated as

$$\begin{aligned}\boldsymbol{\mu}_{n+1} &= \frac{1}{n+1}\sum_{i=1}^{n+1}\boldsymbol{x}_i \\ &= \frac{1}{n+1}\sum_{i=1}^{n}\boldsymbol{x}_i + \frac{1}{n+1}\boldsymbol{x}_{n+1} \\ &= \frac{n}{n+1}\boldsymbol{\mu}_n + \frac{1}{n+1}\boldsymbol{x}_{n+1}. \qquad (1)\end{aligned}$$

For the covariance matrix, a recursive formula is given by the following

$$\begin{aligned}\boldsymbol{C}_{n+1} &= \frac{1}{n+1}\sum_{i=1}^{n+1}(\boldsymbol{x}_i - \boldsymbol{\mu}_{n+1})(\boldsymbol{x}_i - \boldsymbol{\mu}_{n+1})^T \\ &= \frac{1}{n+1}\sum_{i=1}^{n+1}\boldsymbol{x}_i\boldsymbol{x}_i^T - \boldsymbol{\mu}_{n+1}\boldsymbol{\mu}_{n+1}^T \\ &= \frac{n}{n+1}(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\boldsymbol{x}_i^T - \boldsymbol{\mu}_n\boldsymbol{\mu}_n^T) \\ &+ \frac{n}{(n+1)^2}(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n)(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n)^T \\ &= \frac{n}{n+1}\boldsymbol{C}_n + \frac{n}{(n+1)^2}(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n)(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n)^T \\ &= \boldsymbol{C}_n + \frac{1}{n+1}[\frac{n}{n+1}(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n)(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n)^T \\ &- \boldsymbol{C}_n]. \qquad (2)\end{aligned}$$

In Eq. (2) we have substituted $\boldsymbol{\mu}_{n+1}$ by Eq. (1). Let $\boldsymbol{C}_n$ be the covariance matrix at step n-th whose size is $p \times p$. Assume that at $(n+1)$-th, $\boldsymbol{C}_n$ changes slightly by matrix $\boldsymbol{B}$. This can be written as

$$\boldsymbol{C}_{n+1} = \boldsymbol{C}_n + \varepsilon\boldsymbol{B}, \qquad (3)$$

with small parameter $\varepsilon = \frac{1}{n+1}$. In other words, $\boldsymbol{C}_n$ is perturbed by $\boldsymbol{B}$. Now, an eigen equation for $\boldsymbol{C}_{n+1}$ is given by

$$\boldsymbol{C}_{n+1}\boldsymbol{v}_{n+1} = \lambda_{n+1}\boldsymbol{v}_{n+1} \Rightarrow (\boldsymbol{C}_n + \varepsilon\boldsymbol{B})\boldsymbol{v}_{n+1} = \lambda_{n+1}\boldsymbol{v}_{n+1}, \qquad (4)$$

where $(\lambda_{n+1}, \boldsymbol{v}_{n+1})$ is the eigenpair of $\boldsymbol{C}_{n+1}$. Assuming that there are p distinct eigenvalues. We denote the j-th eigenpair of $\boldsymbol{C}_n$ by $(\lambda_{j,n}, \boldsymbol{v}_{j,n})$ with $j = 1, 2, 3, ..., p$. The first order asymptotic approximation for $(\lambda_{j,n+1}, \boldsymbol{v}_{j,n+1})$ can be written as

$$\lambda_{j,n+1}(\varepsilon) \approx \lambda_{j,n} + \varepsilon\lambda_{1j,n}, \qquad (5)$$

$$\boldsymbol{v}_{j,n+1}(\varepsilon) \approx \boldsymbol{v}_{j,n} + \varepsilon\boldsymbol{v}_{1j,n}, \qquad (6)$$

with

$$\lambda_{1j,n} = \frac{< \boldsymbol{B}\boldsymbol{v}_{j,n}, \boldsymbol{v}_{j,n} >}{< \boldsymbol{v}_{j,n}, \boldsymbol{v}_{j,n} >} \qquad (7)$$

$$\boldsymbol{v}_{1j,n} = \sum_{k \neq j}^{p} \frac{< \boldsymbol{v}_{k,n}, \boldsymbol{B}\boldsymbol{v}_{j,n} >}{(\lambda_{j,n} - \lambda_{k,n})} \boldsymbol{v}_{k,n} + \beta \boldsymbol{v}_{j,n}, \quad (8)$$

where β is an arbitrary constant and $\boldsymbol{v}_{j,n}$ is an orthonormal vector. Detailed derivation of Eq. (7) and (8) can be found in [16].

We now apply Eq. (4) to Eq. (2) with $\boldsymbol{B} = \frac{n}{n+1}(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n)(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n)^T - \boldsymbol{C}_n$ and $\varepsilon = \frac{1}{n+1}$. The recursive eigenvalue and eigenvector estimation formula can now be obtained respectively as

$$\lambda_{j,n+1} \approx \lambda_{j,n} + \frac{1}{n+1}(\frac{n}{n+1}\phi_{j,n}^2 - \lambda_{j,n}), \quad (9)$$

$$\boldsymbol{v}_{j,n+1} \approx \boldsymbol{v}_{j,n} + \frac{1}{(n+1)}(\sum_{k \neq j} \frac{\frac{n}{n+1}\phi_{k,n}\phi_{j,n}}{\lambda_{j,n} - \lambda_{k,n}} \boldsymbol{v}_{k,n} + \beta \boldsymbol{v}_{j,n}), \quad (10)$$

where $\phi_{j,n} = (\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n)^T \boldsymbol{v}_{j,n}$, $\phi_{k,n} = \boldsymbol{v}_{k,n}^T(\boldsymbol{x}_{n+1} - \boldsymbol{\mu}_n)$, and β is an arbitrary constant.

2.2 Outlier Detection Method

The Mahalanobis distance measures the number of standard deviations that an observation is from the mean of a distribution, introduced by Prasanta Chandra Mahalanobis in 1930 [21]. Then the Mahalanobis distance as the outlier score of a data point $\boldsymbol{x}$ can be defined by

$$score(\boldsymbol{x}) = \sum_{j=1}^{p} \frac{|(\boldsymbol{x} - \boldsymbol{\mu}) \cdot \boldsymbol{v}_j|^2}{\lambda_j} \quad (11)$$

where $\boldsymbol{\mu}$ is the centroid of the data, λ is the eigenvalue, and $\boldsymbol{v}$ is the eigenvector. For our problem at hand the eigenstructure is given by Eq. (9) and (10). An outlier has to satisfy the following condition

$$score(\boldsymbol{x}) > \sqrt{\chi^2_{d,1-\alpha}} \quad (12)$$

where $\chi^2_{d,1-\alpha}$ is chi-square distribution with degrees of freedom d and $(1-\alpha)\%$ the significance level (For more information concerning Eq. (12), see Aggarwal [1])

2.3 The Proposed Online Outlier Detection Algorithm

Algorithm 1 describes the procedure of detecting outliers using the Mahalanobis distance via modified recursive PCA. We describe how these parameters are updated and how outliers are identified with every step of new data.

2.4 Performance Evaluation

In the field of machine learning and computing, evaluating performance of a classification algorithm is very important [24, 14]. In binary classification, the input

Algorithm 1: Mini-batch Outlier Detection Algorithm via Modified Recursive PCA.

```
Input: a data matrix X ∈ ℜ^{n×p}
Initialization:
μ_n ← (1/n) Σ_{i=1}^{n} x_i
C_n ← (1/n) Σ_{i=1}^{n} (x_i − μ_n)(x_i − μ_n)^T
λ_{j,n} and v_{j,n} of C_n with j = 1, 2, 3, ..., p
for n + 1 to final do
    φ_{j,n} ← (x_{n+1} − μ_n)^T v_{j,n}
    λ_{j,n+1} ← λ_{j,n} + (1/(n+1)) ((n/(n+1)) φ²_{j,n} + −λ_{j,n})
    φ_{k,n} ← (x_{n+1} − μ_n)^T v_{k,n}
    v_{j,n+1} ← v_{j,n} + (1/(n+1)) Σ_{k≠j} ((n/(n+1)) φ_{k,n} φ_{j,n} / (λ_{j,n} − λ_{k,n})) v_{k,n}
    v_{j,n+1} ← v_{j,n+1} / ||v_{j,n+1}||
    μ_{n+1} ← (n/(n+1)) μ_n + (1/(n+1)) x_{n+1}
    score(x_{n+1}) ← Σ_{j=1}^{p} |(x_{n+1} − μ_{k+1}) · v_{j,n+1}|² / λ_{j,n+1}
    if score(x_{n+1}) > sqrt(χ²_{d,1−α}) then
        outlier detected
        record outlier with label 1
        else
            record inlier with label 0
        end if
    end if
end for
Output: binary classification
```

data is grouped into one of two classes. In measuring the performance of an algorithm that is often used in machine learning, especially the classification model, it creates a confusion matrix [4]. This research performs a binary classification, so the results of the confusion matrix are two classes. The confusion matrix aims to compare the classification results of an algorithm with the truth classification results [13, 9].The representation of the confusion matrix is a matrix table with four combinations of predicted values and the actual value where the table can be seen in Table 1.

We define the following loss function $I : Y \times Y \rightarrow \{TP, TN, FP, FN\}$. Let $y \in \{i_0, i_1\}$ be the prediction where i_0 = inlier and i_1 = outlier. The mapping of the I function as follows [7]

- If $y = i_1$ and $\hat{y} = i_1$, then $I(y, \hat{y}) = TP$;
- If $y = i_0$ and $\hat{y} = i_0$, then $I(y, \hat{y}) = TN$;
- If $y = i_0$ and $\hat{y} = i_1$, then $I(y, \hat{y}) = FP$;
- If $y = i_1$ and $\hat{y} = i_0$, then $I(y, \hat{y}) = FN$.

Next, when the prediction result is a real number, a threshold value of t is needed to distinguish positive and negative classes [10].

Furthermore, we use the results of the confusion matrix table to evaluate the performance of the machine learning algorithm for making predictions, namely by calculating the values of accuracy, precision, recall/sensitivity, F1-score. To add a measure to evaluate

Table 1: Confusion Matrix.

		Predicted Values	
		Positive	Negative
Actual Values	Positive	TP	FN
	Negative	FP	TN

the performance of the algorithm, we also use AUC-PR that stands for area under the precision-recall curve [32, 30]. The range of AUC-PR values is between 0 and 1. Generally, the higher the AUC-PR score, the better a classifier performs for the given task. Next, we also calculate the execution time of the algorithm to evaluate the efficiency of the algorithm.

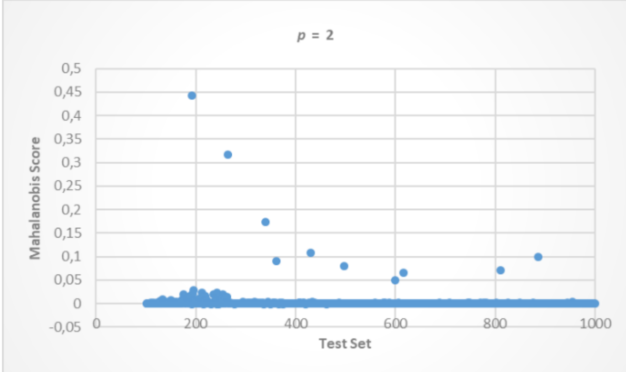

Figure 1: Mahalanobis Score Difference for 2 Features.

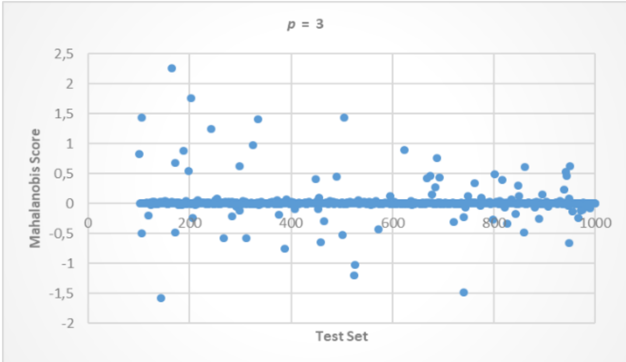

Figure 2: Mahalanobis Score Difference for 3 Features.

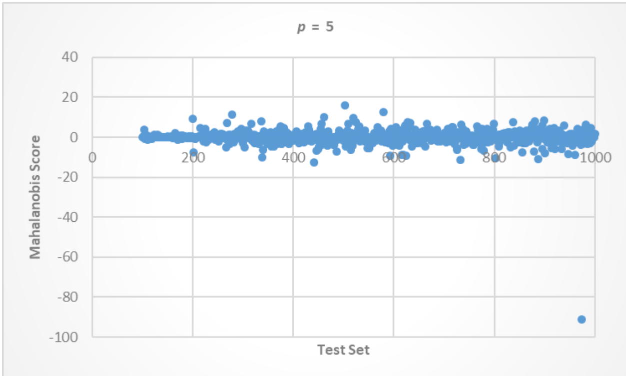

Figure 3: Mahalanobis Score Difference for 5 Features.

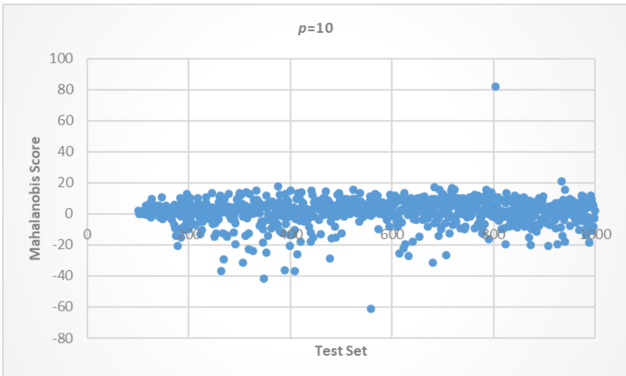

Figure 4: Mahalanobis Score Difference for 10 Features.

3 Results and Discussion

Our experiments are performed on several synthetic datasets. Our synthetic datasets have same number of data points that are created with the scikit-learn module "make_classification" algorithm [26, 6].We generate synthetic datasets to simulate a two-class classification problem with 1000 samples (100 samples of the train set and 900 samples of the test set) with different number of features ($p = 2, 3, 5, 10$) randomly drawn following a standard Normal distribution, i.e $\mathcal{N}(0, 1)$ and one binary dependent variable $y \in Y = \{0, 1\}$ being the class of the data points, the distribution of the two classes defined by Y is imbalanced, i.e. the proportion of observations for which $y = 0$ was 99% and $y = 1$ was 1% with no redundant variables in the dataset. In the following subsections, we compare the Mahalanobis distance results as outlier scores and evaluate the algorithm performance via modified recursive PCA (online algorithm) and classical PCA (offline algorithm) with respect to ground-truth outlier information. In simulating each test data point in the offline outlier detection algorithm, all training data and previous test data are still used.

3.1 Mahalanobis Distance Analysis as Outlier Score

In this section, we show plots of the Mahalanobis score difference on the y-axis obtained from the results of the online and offline algorithm against the test dataset on the x-axis. The scores were obtained from the results of subtracting the Mahalanobis score using the modified recursive PCA method with the classical PCA method.

All plots display the difference in Mahalanobis scores converging to zero for some features although there is a slight spread across some test points. In particular, at 2, 3, and 5 features (Fig. 1 to 3) that the difference in Mahalanobis score values is close to zero. Meanwhile, the 10 features in Fig. 4 show that the Mahalanobis score is slightly different from zero, but still converges around zero.

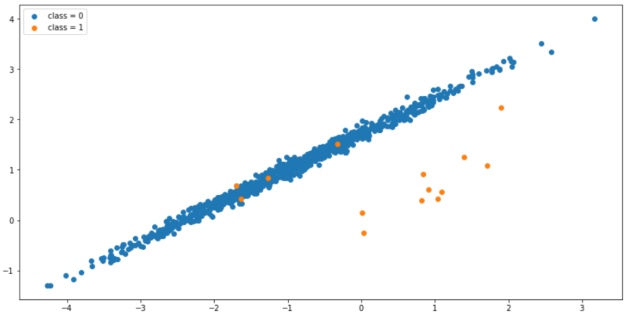

Figure 5: Plot Test Data with Two Features on Ground-Truth.

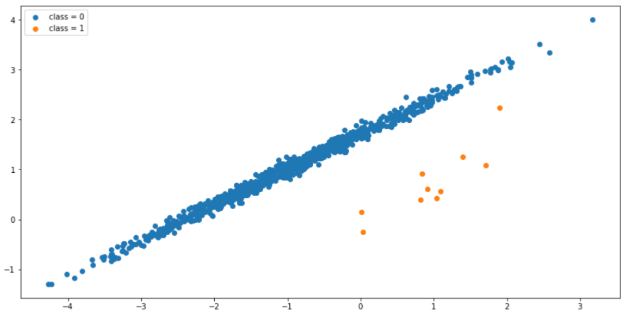

Figure 6: Plot Test Data with Two Features on Offline Algorithm Prediction Values.

3.2 Performance Analysis of Online and Offline Algorithm

In this section, we show the representation of outlier detection results on test data with two features using online and offline algorithms compared to ground-truth. The outliers are shown in yellow color and the inliers are shown in blue color. The ground-truth of the test set used is shown in Fig 5 and the predicted results of the two algorithms on the test data are shown in Fig. 6 and 7. The number of outliers on the ground-truth data is 14 as shown in Table 2. Meanwhile, the number of outliers predicted by the two algorithms is 10 as shown in Table 2. For more details about the outlier detection results with two features, it can be seen in Table 2.

And in this section, we also show the confusion matrix of the results of the online and offline outlier detection algorithms respect to ground-truth of all test data (Table 3 to 6). The results of the confusion matrix of the two algorithms are exactly the same for 2 and 3 features where both algorithms correctly predict a total of 10 outliers and correctly predict a total of 886 inliers. While the results of the confusion matrix are different for 5 and 10 features. For the 5 features, the offline algorithm correctly predicts a total of 9 outliers and correctly predicts a total of 887 inliers, while the online algorithm correctly predicts a total of 10 outliers and correctly predicts a total of 886 inliers. For 10 features, the offline algorithm correctly predicts a total of 9 outliers and correctly predicts a total of 886 inliers, while the online algorithm correctly predicts a total of 7 outliers and correctly predicts a total of 886 inliers.

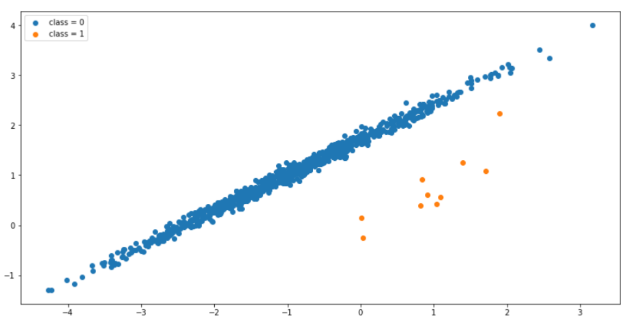

Figure 7: Plot Test Data with Two Features on Online Algorithm Prediction Values.

Table 7 represents the comparison of the performance results of the online and offline outlier detection algorithms on the given test data. The accuracy and precision values of the two algorithms are very satisfying since the values are more than 0.90, then the accuracy of the two algorithms is exactly the same, only the precision differs slightly on 5 features where the offline algorithm's precision is slightly better. The recall and F1-score of the two algorithms are good enough where the values are in the range [0.50,0.83], there are only differences in 5 and 10 features where the recall and F1-score of the online algorithm are comparable to the offline algorithm. The AUC-PR of both algorithms is also satisfactory since the value more than 0.75, there are only differences in 5 and 10 features where the offline algorithm's AUC-PR is slightly better. Then the execution time of the online algorithm is always faster than the offline algorithm for each feature. Overall, it can be seen that the effectiveness of the two algorithms decreases slightly as the features increase, the effectiveness of the online algorithm is comparable to the offline algorithm, and the efficiency of the online algorithm is the most outperforming.

4 Conclusions and Remarks

A way to construct an online algorithm to identify outliers for data streams was discussed in this paper. This research applied a recursive schema strategy that predicts the iterative model to update the parameters in the model when new data appears and detects outliers. In other words, an iterative schema plays an important role for data streams. We constructed an online algorithm to identify outliers with Mahalanobis distance using modified recursive PCA. In conclusion, this work showed that in terms of effectiveness the performance of the online algorithm is comparable to that of the offline algorithm and in terms of efficiency the performance of the online algorithm outperforms the offline algorithm.

We remark here that our proposed algorithm is still limited, it is quite appropriate only for the incoming data whose changes are not too large since we apply the perturbation method in our algorithm. Other things,

Table 2: The Results of Outliers Detection.

2 Features	Ground-Truth Outlier (nth test data)	Predicted Values (Offline Algorithm)	Predicted Values (Online Algorithm)
	92th	√	√
	114th	-	-
	164th	√	√
	238th	√	√
	260th	√	√
	326th	-	-
	330th	√	√
	397th	√	√
	498th	√	√
	516th	√	√
	644th	-	-
	710th	√	√
	723th	-	-
	785th	√	√
Total Outliers	14	10	10

Table 3: Confusion Matrices Results of The Test Set for 2 Features.

		$p = 2$			
		Offline Algorithm		Online Algorithm	
		Predicted Values		Predicted Values	
		Positive (1)	Negative (0)	Positive (1)	Negative (0)
Actual Values	Positive (1)	10	4	10	4
	Negative (0)	0	886	0	886

Table 4: Confusion Matrices Results of The Test Set for 3 Features.

		$p = 3$			
		Offline Algorithm		Online Algorithm	
		Predicted Values		Predicted Values	
		Positive (1)	Negative (0)	Positive (1)	Negative (0)
Actual Values	Positive (1)	10	4	10	4
	Negative (0)	0	886	0	886

Table 5: Confusion Matrices Results of The Test Set for 5 Features.

		$p = 5$			
		Offline Algorithm		Online Algorithm	
		Predicted Values		Predicted Values	
		Positive (1)	Negative (0)	Positive (1)	Negative (0)
Actual Values	Positive (1)	9	4	10	3
	Negative (0)	0	887	1	886

Table 6: Confusion Matrices Results of The Test Set for 10 Features.

		$p = 10$			
		Offline Algorithm		Online Algorithm	
		Predicted Values		Predicted Values	
		Positive (1)	Negative (0)	Positive (1)	Negative (0)
Actual Values	Positive (1)	9	5	7	7
	Negative (0)	0	886	0	886

Table 7: Performance Comparison of Online and Offline Outlier Detection Algorithms.

The number of Features (p):		2	3	5	10
Accuracy (100%)	Offline Algorithm	1.000	1.000	1.000	0.990
	Online Algorithm	1.000	1.000	1.000	0.990
Precision	Offline Algorithm	1.000	1.000	1.000	1.000
	Online Algorithm	1.000	1.000	0.910	1.000
Recall	Offline Algorithm	0.710	0.710	0.690	0.640
	Online Algorithm	0.710	0.710	0.770	0.500
F1-score	Offline Algorithm	0.830	0.830	0.820	0.780
	Online Algorithm	0.830	0.830	0.830	0.670
AUC-PR	Offline Algorithm	0.859	0.859	0.848	0.824
	Online Algorithm	0.859	0.859	0.841	0.754
Time (second)	Offline Algorithm	0.461	0.480	1.082	1.574
	Online Algorithm	0.300	0.414	0.726	1.286

the present algorithm only applies for the data streams arriving one by one. Therefore, for the future work, an outlier detection algorithm involving the incoming data with mini-batch size issue can be extended.

Acknowledgement: The authors sincerely wish to thank the journal editors and reviewers who critically reviewed the manuscript and made valuable suggestions for its improvement. All funding is also supported by P2MI-Institut Teknologi Bandung, Indonesia 2022.

References

[1] Aggarwal, C. C. An introduction to outlier analysis. In *Outlier analysis*. Springer, 2017, pp. 1–34.

[2] Ahmadi, M., Sharifi, A., Jafarian Fard, M., and Soleimani, N. Detection of brain lesion location in mri images using convolutional neural network and robust pca. *International journal of neuroscience* (2021), 1–12.

[3] Al-Fawa'reh, M., Al-Fayoumi, M., Nashwan, S., and Fraihat, S. Cyber threat intelligence using pca-dnn model to detect abnormal network behavior. *Egyptian Informatics Journal 23*, 2 (2022), 173–185.

[4] Alimohammadi, H., and Chen, S. N. Performance evaluation of outlier detection techniques in production timeseries: A systematic review and meta-analysis. *Expert Systems with Applications 191* (2022), 116371.

[5] Bosman, H. H., Liotta, A., Iacca, G., and Wörtche, H. J. Anomaly detection in sensor systems using lightweight machine learning. In *2013 IEEE International Conference on Systems, Man, and Cybernetics* (2013), IEEE, pp. 7–13.

[6] Brownlee, J. *Imbalanced classification with Python: better metrics, balance skewed classes, cost-sensitive learning*. Machine Learning Mastery, 2020.

[7] Caelen, O. A bayesian interpretation of the confusion matrix. *Annals of Mathematics and Artificial Intelligence 81*, 3 (2017), 429–450.

[8] Cesa-Bianchi, N., and Orabona, F. Online learning algorithms. *Annual review of statistics and its application* (2021).

[9] Chicco, D., Starovoitov, V., and Jurman, G. The benefits of the matthews correlation coefficient (mcc) over the diagnostic odds ratio (dor) in binary classification assessment. *Ieee Access 9* (2021), 47112–47124.

[10] Chicco, D., Tötsch, N., and Jurman, G. The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData mining 14*, 1 (2021), 1–22.

[11] Emerson, J. W., and Kane, M. J. Don't drown in the data. *Significance 9*, 4 (2012), 38–39.

[12] Fieri, B., and Suhartono, D. Offensive language detection using soft voting ensemble model. *MENDEL Journal 29*, 1 (2023), 1–6.

[13] Fischer, M. E., Cruickshanks, K. J., Dillard, L. K., Nondahl, D. M., Klein, B. E., Klein, R., Pankow, J. S., Tweed, T. S., Schubert, C. R., Dalton, D. S., et al. An epidemiologic study of the association between free recall dichotic digits test performance and vascular health. *Journal of the American Academy of Audiology 30*, 04 (2019), 282–292.

[14] Gunawan, A. Y., Kresnowati, M. T. A. P., et al. Artificial neural network approach for the identification of clove buds origin based on metabolites composition. *arXiv preprint arXiv:2007.05125* (2020).

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

[16] Hinch, E. *Perturbation methods*. Cambridge University Press, 1992.

[17] Hoeltgebaum, H., Adams, N., and Fernandes, C. Estimation, forecasting, and anomaly detection for nonstationary streams using adaptive estimation. *IEEE Transactions on Cybernetics* (2021).

[18] Ifzarne, S., Tabbaa, H., Hafidi, I., and Lamghari, N. Anomaly detection using machine

learning techniques in wireless sensor networks. In *Journal of Physics: Conference Series* (2021), vol. 1743, IOP Publishing, p. 012021.

[19] Ippel, L., Kaptein, M., and Vermunt, J. Dealing with data streams: An online, row-by-row, estimation tutorial. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences 12*, 4 (2016), 124.

[20] Jolliffe, I. T. *Principal component analysis for special types of data*. Springer, 2002.

[21] Mahalanobis, P. C. On test and measures of group divergence: theoretical formulae. *Journal and Proceedings of Asiatic Society of Bengal New series 26* (1930), 541–588.

[22] Majdoubi, R., Masmoudi, L., Bakhti, M., Elharif, A., and Jabri, B. Parameters estimation of bldc motor based on physical approach and weighted recursive least square algorithm. *International Journal of Electrical & Computer Engineering (2088-8708) 11*, 1 (2021).

[23] Pokrajac, D., Lazarevic, A., and Latecki, L. J. Incremental local outlier detection for data streams. In *2007 IEEE symposium on computational intelligence and data mining* (2007), IEEE, pp. 504–515.

[24] Saberioon, M., Císař, P., Labbé, L., Souček, P., Pelissier, P., and Kerneis, T. Comparative performance analysis of support vector machine, random forest, logistic regression and k-nearest neighbours in rainbow trout (oncorhynchus mykiss) classification using image-based features. *Sensors 18*, 4 (2018), 1027.

[25] Schifano, E. D., Wu, J., Wang, C., Yan, J., and Chen, M.-H. Online updating of statistical inference in the big data setting. *Technometrics 58*, 3 (2016), 393–403.

[26] Sippola, V., and Mercer, R. E. An experimental comparison of the geometry of models trained on natural language and synthetic data. In *Canadian Conference on AI* (2021).

[27] Snijders, C., Matzat, U., and Reips, U.-D. " big data": big gaps of knowledge in the field of internet science. *International journal of internet science 7*, 1 (2012), 1–5.

[28] Thuy, T. T. T., Thuan, L. D., Duc, N. H., and Minh, H. T. A study on heuristic algorithms combined with lr on a dnn-based ids model to detect iot attacks. *MENDEL Journal 29*, 1 (2023), 62–70.

[29] Wang, C., Chen, M.-H., Wu, J., Yan, J., Zhang, Y., and Schifano, E. Online updating method with new variables for big data streams. *Canadian Journal of Statistics 46*, 1 (2018), 123–146.

[30] Wissel, B. D., Greiner, H. M., Glauser, T. A., Pestian, J. P., Kemme, A. J., Santel, D., Ficker, D. M., Mangano, F. T., Szczesniak, R. D., and Dexheimer, J. W. Early identification of epilepsy surgery candidates: A multicenter, machine learning study. *Acta Neurologica Scandinavica 144*, 1 (2021), 41–50.

[31] Zangeneh-Nejad, F., Amiri-Simkooei, A., Sharifi, M., and Asgari, J. Recursive least squares with additive parameters: Application to precise point positioning. *Journal of Surveying Engineering 144*, 4 (2018), 04018006.

[32] Zea-Vera, R., Ryan, C. T., Havelka, J., Corr, S. J., Nguyen, T. C., Chatterjee, S., Wall Jr, M. J., Coselli, J. S., Rosengart, T. K., and Ghanta, R. K. Machine learning to predict outcomes and cost by phase of care after coronary artery bypass grafting. *The Annals of Thoracic Surgery 114*, 3 (2022), 711–719.