Unsupervised anomaly detection of runners performance

Bota Duisenbay

"La Sapienza" University of Rome, via Ariosto 25, Rome, Italy

Abstract

Unsupervised anomaly detection is a process of identification of deviant portions of a data without any label given. In this project analyzed data is readings of heart beat rate and speed of an athlete during a training. The data was used as a collection of points and as a timeseries to capture temporal sequence. Three most common machine learning algorithm for anomaly detection (kNN, OC SVM and KDE) for points as well as GANs for timeseries (TadGAN) were used and compared for this task. It has been revealed that kNN and OCSVM performed well at detecting point anomalies only, while KDE can detect group anomalies as well. In contrast, contextual and group anomalies has been clearly highlighted in TadGANs. Therefore, with improvements on point anomalies, TadGANs is suggested to be studies further for anomaly detection in athletes performance readings.

Keywords

Unsupervised Anomaly Detection, KNN, OC SVM, KDE, TadGANs for multivariate timeseries,

1. Introduction

A variety of training load monitoring technologies has become widely accessible to general public via lightweight wearable devices and cell phones. The data collected by these devices reflects individual physiological responses and physical conditions. It is useful to analyze a performance, track fitness of an athlete and improve effectiveness and efficiency of the training. Depending on a goal and current physical condition of an athlete, adjustments during the training help to prevent injuries, to avoid over strain, while not under strain, and even detect risk factor for cardiovascular diseases. For optimal training, these adjustments are based on useful metrics depending of type of physical activity. For runners, there are athlete's speed and heart beat rate during a training as an athlete can adjust his/her speed to perform at particular intensity zone that is based in heart beat rate [1].

Both speed and heart beat rate measurements are indirect, and resulted from approximation based on relations with other more accessible non-invasive readings. Estimation of speed is done using GPS by either deriving speed from change in distance (which comes from positional differentiation) or calculating with Doppler's shift method. It has been proven that Doppler's shift method provides more accurate than distance based method [2]. Nevertheless, not only a chosen method affect accuracy of calculated speed, but also manufacturer's other settings like sampling rate. Apart from that, speed of a runner itself may account for from 3 to 8% of error as speed increases [3]. As well as speed, heart beat rate measurements are made easily accessible thanks to optical heart rate sensors put on wrist. It flashes green LED lights and measures absorption of it by blood, then converts to an approximate number of times the heart beats per minute, bpm [4]. This widely used method is called photoplethysmography. Due to different internal and external factors discussed in [4] after testing validity of heart rate readings of various devices, these indirect measurements from wearable devices vary from device to device, and subject to uncertainties and anomalous readings, that are needed to be adjusted and compensated too [5].

Apart from technical issues, anomaly may also come from an athlete itself as a body structure deviation or health condition specifics during rehabilitation period. Given these two potential sources of anomaly and uncertainty: from an athlete physical condition and from device readings (both speed and bpm), to notify about possible health risk for the user and improve sensors themselves, it is crucial to detect anomaly of readings.

The aim of the work is detect portions of speed and heart beat rate measurements, where values are abnormal, generally called as anomaly detection. Anomalies can be categorized as point, contextual and collective (or group) anomaly. Point anomaly, also referred as an outlier, is a single data point that is different from other. If data is identified as a deviation or taken as a usual depending on context of a scenario as a condition, then it is contextual. If abnormality of concurrence in a sequence is detected not based on condition, but with regard to the entire data set, then it is classified as a collective anomaly [6] [7]. These differences are clearly represented in a figure 1 taken from the review by Ruff et al. [8].

As in a standard setting in anomaly detection there is no labelling present for these readings recorded during the training. Therefore only unsupervised the methods were considered for this task. It is aimed to detect point, contextual and collective anomalies by analysing data

Final project for Elective in AI and Seminar in AI

duisenbay.1849680@studenti.uniroma1.it (B. Duisenbay)

CEUR Workshop Proceedings (CEUR-WS.org)

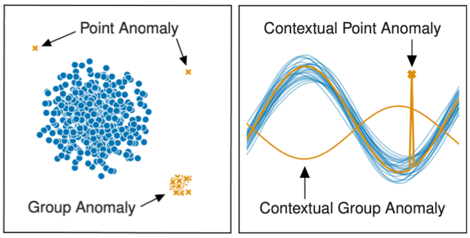

Figure 1: Anomaly types

point-wise and as time series. Speed and heart beat rate at each timestamp considered separately as a collection of data point, disregarding temporal relations between points. Then, conversely, temporal change in data points is analyzed as a time series to detect anomaly, taking into account preceding and following values.

Detection anomalies in speed/heart beat rate readings is done in this project by comparing different existing techniques for anomaly detection. For KNN, OCSVM, KDE sklearn implementation and for GAN TadGAN implementation were adapted for this task.

The main questions to be addressed in this work are the following:

- Importance of time for speed/heart beat rate anomaly detection to be studied by keeping only sequential relations (2 variable time series) and removing time frame (2d point-wise).
- Effectiveness of classic unsupervised methods KNN, OCSVM and KDE, and GANs based methods for a given data set.

The rest of this report is structured as follows. It starts from defining problem of detection and a brief overview of approaches available in literature in Section Related works. Section Implementation contains theoretical details for each chosen unsupervised approaches and implementation details, followed by results and discussion of findings in Section Results. Finally, Section Conclusion summaries the project

2. Related Works

Anomaly detection has been studied for centuries and has wide range of applications. Various reviews have offered different ways to classify methods. In the review by Nassif et al (2021) [9], that studies 290 research articles published on anomaly detection from 2000-2020, 29 distinct machine learning techniques were grouped as following: Classification, with subcategories like Support Vector Machine (SVM), Decision Tree, Bayesian Network, Neural Networks (NN) and K-Nearest Neighbors (kNN), Optimization, Ensemble, Rule System, Clustering, including k means and hierarchical clustering, and Regression. These techniques are used as standalone or, commonly, as hybrid, obtained by combination of two or more techniques.

Another review by Ruff et al [8] declares that the main idea in unsupervised anomaly detection is to find a model of a "normal" data in order to detect deviations from the model as anomaly. Therefore, unsupervised anomaly detection techniques are divided based on models into three categories: **probabilistic, classification and reconstruction models**. Purely **distance based** models has been taken separately.

Probabilistic models' detection is based on estimating probability distribution of normal data. In classic density estimation approaches it is done by either computing a distance like Mahalanobis to train data mean or fitting a Gaussian distribution and evaluating log-likelihood [10]. More sophisticated versions include kernel density estimation[11], Gaussian Mixture Models (GMM)[12] and histogram estimators[13]. Most prominent probabilistic approaches are neural generative models that learns NN, which maps sampled vector to a training distribution. One of the promising example for this approach could be Variational Autoencoders (VAEs)[14] and Generative Adversarial Networks (D-based GANs)[15]. These methods show good performance on low dimensional feature space, but accuracy suffers with an increase of dimensionality and it requires more samples to attain the same accuracy[8]. Besides, there is an excessive full estimation of the density as an intermediate step. This problem is addressed in one class classification approach.

In **classification** a decision boundary between anomaly and normal data is learned explicitly. Most known kernel based One Class SVM (OC-SVM) [16] [17] and Support Vector Data Description (SVDD) [18] learn a hyper-sphere with minimum volume that would fit the training data. With application of neural network to learn/ transfer feature maps, relevant feature and kernel selection challenges has been solved in modifications as Deep SVDD [19].

Like the previous approaches, **reconstruction** methods assume that normal data can be grouped as similar prototypes. In case of reconstruction methods, data is structured and can be characterised using finite number of properties (manifold assumption) and grouped into a finite number of collections(prototype assumption). As name suggests, these methods score anomalous data based on how accurate it can be reconstructed by learned model. Higher reconstruction error suggests higher anomaly score as we assume fewer anomaly in the training data. It consists of encoding and decoding that maps to latent representation and back. Depending on technique used for encoding and decoding there are variations of Principal Component Analysis (PCA) [20], Autoencoders with RNN and LSTM [21], and GANS(G based)[22][23].

Apart from the methods discussed above, there is also **distance-based** models that does not have a training for learning a model, as it evaluates new points with respect to the training data. In a review by Goldstein et al. [24] that compared distance based algorithms on 19 various data sets. Generally nearest neighbor based methods reveal better performance than clustering based ones. However, it needs to be noted that Nearest Neighbor based methods have higher computational time. Also it has been concluded that global anomaly detection algorithms perform better in cases when nature of anomaly in a data set is not identified beforehand. As local anomaly detection methods like LOF, COF and INFLO that show poor performance when data set contains both local and global anomalies, while global anomaly detection algorithms like kNN, perform at least average on local anomalies[25]. Hence, kNN has been chosen for non timeseries part of the project. It's been used in hybrid approaches to rank outliers based on distance.

All above mentioned approaches have been developed for different applications and hence has adaptation on different types of dataset as the one to be studied in the case of this work, collection of independent data point and as time-series. The main difference between those types is that the first one will produce the same result even if it would be shuffled, whereas for the second sequential order matters [26]. For this tabular data is taken as a time series and anomaly detection algorithm's task is to identify anomalous portion - sub-sequence of a varied length. After cutting sequence into overlapping or non overlapping sub sequences and most of papers apply clustering to get outlier sub sequence, learn to predict or reconstruct a signal and high difference from a real one suggests presence of an anomaly [27].

Although all relies on the assumption that anomalies are rare, there is not a single benchmark dataset to compare those algorithms. Either synthetic or real world data is used as the dataset for anomaly detection. Synthetic anomalies is generated given normal data, and suffers from not being able to replicate all possible real anomalies. Instead data collected from a real application requires human expert annotation, which is not always possible due to large volume of training data required and expenses. Hence, most of the work are trained on unsupervised methods, leaving annotated data for evaluation.

As different methods are trained and tested for data of different domains, there is not possible to define a single state-of-art method. The most compared as showing high performance there are the following: [28], [29], [30], [31], [15], [23] and [32]. Among those fewer has been adapted for anomaly detection to multivariate time series. Most of the methods for multivariate timeseries have been proposed for a data with high dimensionality, while ours has only 2. Hence in terms of efficiency for this task TadGAN[27] version for multivariate has been chosen for implementation of time-series part of this work.

Table 1
Data set description

	Train	Test	Total
# runs	34	9	43
# points	137515	43263	180876
# inf speed points	1	1	2
# inf bpm points	0	0	0
# NaN speed points	3218	101	3319
# NaN bpm points	108993	35026	144019

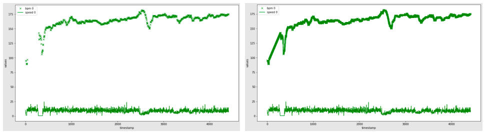

Figure 2: Before (left) and After (right) preprocessing

3. Unsupervised Anomaly Detection: Implementation

3.1. Data preparation

The given dataset consists of 43 recorded sessions of bpm and speed readings with timestamps, and duration for speed. These runs were divided into train and test sets as 34 and 9. Corresponding readings of bpm and speed are not correlated and recorded separately. Hence, the first step was to unify the readings of each session converting timestamps second-wise as time passed since the start, later referred as time index, extending speed points with respect to indicated duration. The resulting table with bpm, speed and time index, is found to be unevenly spaced timeseries, with missing values as NaN and infinite values, see table 1. Inf values were replaced as 10 times higher than median in a column, while for missing data linear interpolation was performed as the most common approach to fill irregularly spaced timeseries. The results of the pre-processing can be viewed in an example of first run's graph 2

For point-wise methods it involves scaling of features to equalize influences of each variable, so features are standardized by removing the mean and scaling to unit variance. Mean and standard deviation of the training data applied for both training and testing points.

For timeseries technique, heart beat rate and speed data are combined as a multivariate time series without time index eq1:

$$X = (x^1, x^2, ..., x^T) \tag{1}$$

where $x^i \in R^{2\times 1}$ is a vector of heart rate and speed measurement at each time step (second) i, and T is total number of data points. To capture temporal correlations, X is cut further into overlapping N sub-sequences using sliding window with window size t = 100 and step size s = 5, resulting in collection of samples X in eq(2) and N calculated as in eq(3)

$$X = \{(x_i^{1...t})\}_{i=1}^{N} \tag{2}$$

$$N = (T - t)/s \tag{3}$$

3.2. Distance based model:k Nearest Neighbors

All the data points from each run were collected together into a single set $X = (x_1, x_2, ..., x_n)$ with each point represents speed and bpm $x_i \in R^2$. Euclidean distance between pairs of points in a dataset is calculated as in eq 4.

$$dist = \sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2} \tag{4}$$

Based on smallest distance, belonging to one of k groups is set for each point. For kNN average distance to all k nearest neighbor is used. This step done in training data is considered as a preprocessing and generally takes most of the computational time.

Under an assumption that outlier are far away from normal points, to classify a point as a normal or an anomalous instance, m points with the largest distances to rank or specific distance threshold were used.

It is commonly used in practice, but the most complication arises from high dependence of choice of k and threshold value m on data set itself and no possibility to carry out cross validation due to absence of labels [24]. For this task, after experimenting with different k, k = 50 has been chosen for a comparison with other methods.

3.3. Classification Model: One Class Support Vector Machine

OC SVM can be viewed as a process with following steps and associated parameters to tune:

1. Mapping from input to higher dimensional feature space using a kernel function like linear, sigmoid, radial basis function.
2. Finding an optimal decision boundary (hyperplane) in feature space, that separate majority of data as inliers leaving n percent as outliers.

Instead of tuning margin parameter C, there is nu parameter that approximately correlates with n percent. It is an upper bound on the fraction of training errors and a lower bound of the fraction of support vectors[33]. If nu=0.05, then at most 5% of training data can be classified and at least 5% of training data will become support vector. During the experiments, kernel rbf with gamma = 0.1 and nu = 0.01 has shown better results and used for further comparison.

3.4. Probabilistic Model: Kernel Density Estimator

It is probabilistic non-parametric approach that estimates probability density of training data, using kernel function K and bandwidth h to get height curve for each point and summing with kernel estimator function after normalizing each [34]. General equation for multivariate pdf used for d-dimensional n data points X is expressed as in eq 5:

$$pdf(x) = \frac{1}{Nh^d} \sum_{i=1}^{N} K\{\frac{1}{h}(x - X_i)\} \tag{5}$$

Although there is a wide variation of choice for Kernel function, usually Gaussian is taken as there is no significant difference shown. Alike kernel, bandwidth selection crucially affects the performance [35]. The optimal fixed bandwidth found experimentally for the given data set and the final parameters are the following: euclidean distance metric with gaussian kernel and bandwidth=0.5.

As long as probability density function is obtained, comparing density of a sample with a threshold needed to separate outliers[36].

3.5. Restoration Model:TadGAN

Reconstruction is a model that maps a segment of time series from input data domain X into custom defined latent domain Z, and maps a segment back from Z to X domain. The processes are both Generators and called encoding ε and decoding ζ respectively. With these notation, reconstruction can be written as in eq 6:

$$x_i \rightarrow \varepsilon(x_i) \rightarrow \zeta(\varepsilon(x_i)) \approx \hat{x_i} \tag{6}$$

To improve quality of reconstruction, there are Critics for each domains: C_z is for encoding improvement, while C_x discriminates real time series x from generated after reconstruction for improvement of both encoding and decoding. The full architecture is depicted in figure 3.

As a model is trained using mostly usual segments, during those mappings usual points are expected to be reconstructed better, thus having lower reconstruction error compared to anomalous portions. Due to the same assumption that all training data points are non-anomalous,

Table 2
GANs Implementation Details

Encoder	LSTM(100, 20, bidirectional) Linear(40, 20, bias)
Decoder	LSTM(20, 64, bidirectional) LSTM(20, 64, bidirectional) Linear(128, 100, bias)
CriticX	Linear(100, 20, bias) Linear(20, 1, bias)
CriticZ	Linear(20, 1, bias)

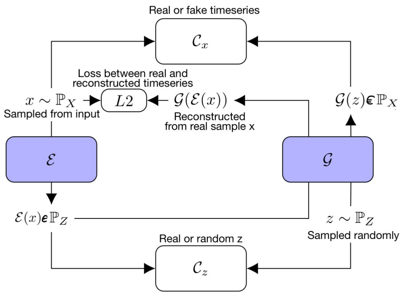

Figure 3: TadGAN architecture

Z is sampled randomly from normal distribution $N(0,1)$ as in eq 7 below, with 2 dimensional vector z_i and k as a dimension of a latent space (k = 20).

$$Z = \{(z_i^{1...k})\}_{i=1}^{N} \tag{7}$$

As Encoding $\varepsilon(x)$ and Decoding $\zeta(z)$ Generators TadGAN uses bidirectional LSTM network with depth 1 and 100 hidden (internal) units and a bidirectional LSTM network with depth 2 and 64 hidden units respectively. For both discriminators C_x and C_z it is 1-D convolutional layer. The GANs model is specified in table 2, with Adam optimizer and learning rate 1e-6. The model is trained with 10 epoch only and a batch size of 64.

Like other GANs methods [32] [37] there anomaly score is calculated from both Discrimination and Residual Losses. So, anomaly detection in TadGAN is done in two levels: distinguish between real and fake timeseries using C_x, while difference between real and restored by Generators captured by L_2. However, alike the mentioned approaches, in TadGAN to measure similarity distance of reconstructed and original timeseries Dynamic Time Wrapping (DTW) technique [38] was used and for losses Wasserstein-1 distance was used as in eq 8. As both reconstruction and critic scores were obtain, their z-scores are calculated to normalize and combined into a single score as average.

$$\underset{\{\epsilon,\zeta\}}{min} \underset{\{C_x \in \mathbf{C_x}, C_z \in \mathbf{C_z}\}}{max} V_X(C_x,\zeta)+V_Z(C_z,\epsilon)+V_L2(\epsilon,\zeta) \tag{8}$$

Instead of detecting anomalies as in the paper, after collecting scores for each segment, scores get scaled between 0 and 1, as they contained both negative and positive scores. To assigned for each point, mean of scores of all 20 segments containing a point is used as a score for the point.

4. Results and Discussion

As no label present for the given dataset, to compare the methods plots shown in table 3 are used to visually investigate the performance on test data. Each method uses different approach resulting in anomaly score of different intervals, therefore anomaly scores for each method get scaled to range from 0 to 1 using MinMaxScaler for comparisons.

The first row shows how scores are distributed across all test data with red horizontal line indicating 99.9 percentile. The 99.9 percentile is later used as a threshold to classify points as outliers. Distribution of speed and bpm is plotted with time index to illustrate that non linearity of bpm and speed with respect to time and how top 1 percent of anomalies are distributed. To illustrate how point and collective anomalies were found, bpm versus speed graph is plotted with anomaly score color-bar. Whereas for contextual anomalies, bpm and speed were plot as timeseries for each method.

In KNN and OCSVM global point outliers are clearly highlighted, but there is a little variation in scores for group outliers and no local outliers captured. As expected, in example run plot as timeseries, anomaly scores are not affected by significant and rapid variations at all and only extreme global point outliers can be detected.

In contrary, anomaly scores from KDE varies discriminating both global point anomalies and group anomalies. In the timeseries, it detects the point anomaly with extreme speed value and rapid bpm dip. However another steady bpm dip has been unnoticed, meaning local outliers are not captured. These local outliers require time information to be embedded as context, which is not a case for KDE.

Expectantly, TadGANs trained on timeseries to detect contextual point and group anomalies shows higher sensitivity to fluctuations. The scores are based on variation in a sequence of values, and it is been observed on all test data that plateaus have higher anomaly scores, while rapid changes have lower score. Opposite to the results previous methods, despite of well captured contextual group anomalies, global and contextual point anomalies were not detected. This could be a result of using slid-

KNN	OCSVM	KDE	TadGAN

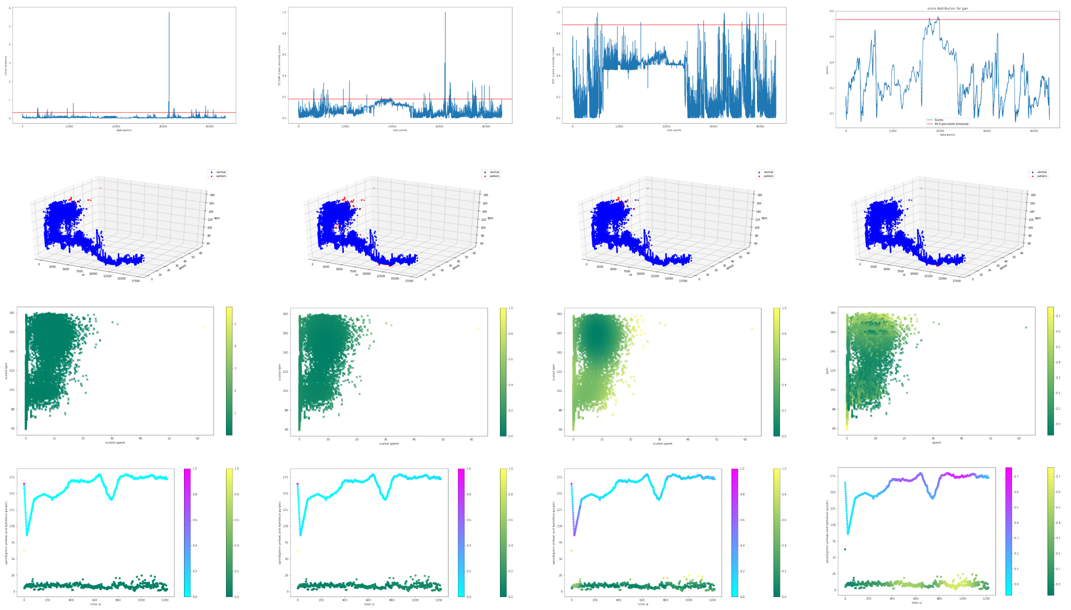

Table 3
Resultant plots of speed, bpm and scores from each method used in columns for test data. Row-wise: distribution of score and 99,9 percentile; time index-speed-bpm of test data with 99.9 percentile as anomalies; speed-bpm correlation with score bar; example test run's speed and bpm as timeseries with score bar

ing mean across window scores to compute point-wise scores. Hence, post-processing to assign scores from intersecting window to points has be studies further. Usage of smaller step size (1 instead of 5) in sliding windows and shorter window size are potential ways to improve detection of point outliers.

5. Conclusion

Runners' heart beat rate and speed data collected during training sessions were analyzed to detect anomalies in sensor readings. There are variety of unsupervised anomaly detection approaches such as distance-based, classification, probabilistic and reconstruction, and kNN, OC SVM, KDE and TadGAN methods from each approach has been tested on the given dataset. The data was treated as collection of point and as timeseries. Point-wise to detect point and group anomalies for first three methods and as timeseries to capture temporal sequence as well for contextual anomalies detection using TadGAN. As expected, point-wise methods performed well on detection of global point outliers only. Instead TadGANs has shown better performance in capturing contextual group anomalies, but failed to detect point anomalies. With improvements addressing indicated issues and further study of post-processing, usage of TadGANs like timeseries based methods is anticipated to show better performance over point-wise methods for anomaly detection in athlete's readings.

References

[1] M. Ludwig, K. Hoffmann, S. Endler, A. Asteroth, J. Wiemeyer, Measurement, prediction, and control of individual heart rate responses to exercise—basics and options for wearable devices, Frontiers in physiology 9 (2018) 778.
[2] J. Zhang, K. Zhang, R. Grenfell, R. Deakin, Short note: On the relativistic doppler effect for precise velocity determination using gps, Journal of Geodesy 80 (2006) 104–110.
[3] M. Cardinale, M. C. Varley, Wearable training-monitoring technology: applications, challenges, and opportunities, International journal of sports physiology and performance 12 (2017) S2–55.
[4] J. Montes, Wearable Technology Devices: Heart

Rate and Step Count Analysis, Ph.D. thesis, University of Nevada, Las Vegas, 2019.

[5] J. W. Navalta, J. Montes, N. G. Bodell, R. W. Salatto, J. W. Manning, M. DeBeliso, Concurrent heart rate validity of wearable technology devices during trail running, Plos one 15 (2020) e0238569.

[6] R. Al-amri, R. K. Murugesan, M. Man, A. F. Abdulateef, M. A. Al-Sharafi, A. A. Alkahtani, A review of machine learning and deep learning techniques for anomaly detection in iot data, Applied Sciences 11 (2021) 5320.

[7] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM computing surveys (CSUR) 41 (2009) 1–58.

[8] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K.-R. Müller, A unifying review of deep and shallow anomaly detection, Proceedings of the IEEE (2021).

[9] A. B. Nassif, M. A. Talib, Q. Nasir, F. M. Dakalbab, Machine learning for anomaly detection: A systematic review, IEEE Access (2021).

[10] S. Roberts, L. Tarassenko, A probabilistic resource allocating network for novelty detection, Neural Computation 6 (1994) 270–284.

[11] J. Kim, C. D. Scott, Robust kernel density estimation, The Journal of Machine Learning Research 13 (2012) 2529–2565.

[12] L. Xiong, B. Póczos, J. Schneider, A. Connolly, J. VanderPlas, Hierarchical probabilistic models for group anomaly detection, in: Proceedings of the fourteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2011, pp. 789–797.

[13] B. Nachman, D. Shih, Anomaly detection with density estimation, Physical Review D 101 (2020) 075042.

[14] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, Special Lecture on IE 2 (2015) 1–18.

[15] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, U. Schmidt-Erfurth, f-anogan: Fast unsupervised anomaly detection with generative adversarial networks, Medical image analysis 54 (2019) 30–44.

[16] I. Steinwart, D. Hush, C. Scovel, A classification framework for anomaly detection., Journal of Machine Learning Research 6 (2005).

[17] K.-L. Li, H.-K. Huang, S.-F. Tian, W. Xu, Improving one-class svm for anomaly detection, in: Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693), volume 5, IEEE, 2003, pp. 3077–3081.

[18] M. Turkoz, S. Kim, Y. Son, M. K. Jeong, E. A. Elsayed, Generalized support vector data description for anomaly detection, Pattern Recognition 100 (2020) 107119.

[19] Z. Zhang, X. Deng, Anomaly detection using improved deep svdd model with data structure preservation, Pattern Recognition Letters 148 (2021) 1–6.

[20] H. Ringberg, A. Soule, J. Rexford, C. Diot, Sensitivity of pca for traffic anomaly detection, in: Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, 2007, pp. 109–120.

[21] S. Maleki, S. Maleki, N. R. Jennings, Unsupervised anomaly detection with lstm autoencoders using statistical data-filtering, Applied Soft Computing 108 (2021) 107443.

[22] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, V. R. Chandrasekhar, Efficient gan-based anomaly detection, arXiv preprint arXiv:1802.06222 (2018).

[23] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, S.-K. Ng, Madgan: Multivariate anomaly detection for time series data with generative adversarial networks, in: International Conference on Artificial Neural Networks, Springer, 2019, pp. 703–716.

[24] M. Goldstein, S. Uchida, A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data, PloS one 11 (2016) e0152173.

[25] M. Amer, M. Goldstein, Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer, in: Proc. of the 3rd RapidMiner Community Meeting and Conference (RCOMM 2012), 2012, pp. 1–12.

[26] A. Blázquez-García, A. Conde, U. Mori, J. A. Lozano, A review on outlier/anomaly detection in time series data, ACM Computing Surveys (CSUR) 54 (2021) 1–33.

[27] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, K. Veeramachaneni, Tadgan: Time series anomaly detection using generative adversarial networks, in: 2020 IEEE International Conference on Big Data (Big Data), IEEE, 2020, pp. 33–43.

[28] M. Munir, S. A. Siddiqui, A. Dengel, S. Ahmed, Deepant: A deep learning approach for unsupervised anomaly detection in time series, Ieee Access 7 (2018) 1991–2005.

[29] P. Zhao, X. Chang, M. Wang, A novel multivariate time-series anomaly detection approach using an unsupervised deep neural network, IEEE Access 9 (2021) 109025–109041.

[30] B. Pincombe, Anomaly detection in time series of graphs using arma processes, Asor Bulletin 24 (2005) 2.

[31] G. S. Chadha, I. Islam, A. Schwung, S. X. Ding, Deep convolutional clustering-based time series anomaly detection, Sensors 21 (2021) 5488.

[32] M. A. Bashar, R. Nayak, Tanogan: time series anomaly detection with generative adversarial networks, in: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2020, pp. 1778–

1785.
[33] P.-H. Chen, C.-J. Lin, B. Schölkopf, A tutorial on ν-support vector machines, Applied Stochastic Models in Business and Industry 21 (2005) 111–136.
[34] B. W. Silverman, Density estimation for statistics and data analysis, Routledge, 2018.
[35] A. Eckert-Gallup, N. Martin, Kernel density estimation (kde) with adaptive bandwidth selection for environmental contours of extreme sea states, in: OCEANS 2016 MTS/IEEE Monterey, IEEE, 2016, pp. 1–5.
[36] R. Laxhammar, G. Falkman, E. Sviestins, Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator, in: 2009 12th International Conference on Information Fusion, IEEE, 2009, pp. 756–763.
[37] Z. Niu, K. Yu, X. Wu, Lstm-based vae-gan for time-series anomaly detection, Sensors 20 (2020) 3738.
[38] M. Müller, Dynamic time warping, Information retrieval for music and motion (2007) 69–84.