An Empirical Study on Anomaly Detection Algorithms for Extremely Imbalanced Datasets

Gonçalo Fontes, Luís Miguel Matos, Arthur Matta, André Pilastri, Paulo Cortez

An Empirical Study on Anomaly Detection Algorithms for Extremely Imbalanced Datasets

Gonçalo Fontes[2 [0000−0003−0288−9029]], Luís Miguel Matos[1 [0000−0001−5827−9129]], Arthur Matta[2 [0000−0002−4902−9483]], André Pilastri[2 [0000−0002−4380−3220]], and Paulo Cortez[1 [0000−0002−7991−2090]]

[1] ALGORITMI R&D Centre, Department of Information Systems,University of Minho, 4804-533 Guimarães, Portugal
luis.matos@dsi.uminho.pt, pcortez@dsi.uminho.pt
[2] EPMQ - IT CCG ZGDV Institute, 4804-533 Guimarães, Portugal
goncalo.fontes@ccg.pt, arthur.matta@ccg.pt, andre.pilastri@ccg.pt

Abstract. Anomaly detection attempts to identify abnormal events that deviate from normality. Since such events are often rare, data related to this domain is usually imbalanced. In this paper, we compare diverse preprocessing and Machine Learning (ML) state-of-the-art algorithms that can be adopted within this anomaly detection context. These include two unsupervised learning algorithms, namely Isolation Forests (IF) and deep dense AutoEncoders (AE), and two supervised learning approaches, namely Random Forest and an Automated ML (AutoML) method. Several empirical experiments were conducted by adopting seven extremely imbalanced public domain datasets. Overall, the IF and AE unsupervised methods obtained competitive anomaly detection results, which also have the advantage of not requiring labeled data.

Keywords: Autoencoder · Deep Learning · Isolation Forest · One-class classification · Random Forest · AutoML · Supervised Learning · Unsupervised learning.

1 Introduction

Anomaly detection, also known as outlier detection or novelty detection, has been an area of study for several years due to its value in diverse real-world application domains, such as: fraud detection [1], network intrusion [18] and predictive maintenance [14]. Anomaly detection can be defined as the identification of abnormal or anomalous events that deviate from the perceived normal ones [30]. Usually, anomaly detection involves dealing with highly imbalanced data, since anomalous events are often rare [21].

Within this context, Machine Learning (ML) algorithms have been widely applied to anomaly detection. A common approach is to employ supervised learning methods, such as Logistic Regression [29], Decision Trees (DT) [33] and Random Forests [34]. These supervised ML algorithms are often coupled with resampling techniques to balance the training data, such as SMOTE [9] or Gaussian Copula (GC) [32]. Within the supervised learning domain, there has been a stronger focus on the usage of Automated ML (AutoML) [13], which alleviates the modeling effort by automating the ML algorithm and hyperparameter search. Indeed, in a recent study, an AutoML method

was compared favorably with other ML algorithms (e.g., DT, RF) when performing an anomaly detection industrial quality inspection task [26]. One disadvantage of the supervised learning approach is that it requires labeled data, which often requires a huge manual effort to create/obtain correct labeled data. An alternative is to employ an unsupervised learning, in particular via a one-class learning approach, where the ML algorithms are only fed with normal examples (the majority class). This includes algorithms such as Isolation Forest (IF) [20] and deep AutoEncoders (AE) [26,34].

In this paper, we attempt to empirically measure the effect of several state-of-the-art ML methods when applied to anomaly detection tasks. The compared methods include: supervised – RF, AutoML; and unsupervised – IF and AE. The four methods were compared by adopting seven extremely imbalanced public domain datasets. The paper is structured as follows. Section 2 describes the related work. Section 3 presents the public domain datasets, preprocessing methods, the compared Machine Learning (ML) algorithms and the evaluation methodology. Then, Section 4 presents the obtained results. Lastly, the main conclusions are discussed in Section 5.

2 Related Work

Anomaly detection is a key ML task that impacts in several application domains (e.g., Finance, Fraud, Industry, Security). In effect, the early studies addressing this task dates back to the 1960s [15]. In more recent years, a diverse range of algorithms have been proposed for anomaly detection, including based on statistics [17], clustering [8,1], classification [8,33,29,34,26] and graph mining [2]. In particular, supervised classification approaches require labeled data that is often difficult to obtain (e.g., requiring human effort). When labeled data is available, it is often extremely unbalanced, since anomalous events tend to be rare. Thus, some supervised learning studies employ balancing training methods, such as SMOTE [9] or GC [32].

One of the challenges that anomaly detection has to address is that the boundaries between normal and abnormal data are often not clearly defined, typically addressed by using an unsupervised or one-class learning methods [6,36]. Under the one-class learning approach, the training datasets only contain “normal" examples. The assumption is that any anomaly should be more distanced from the training learning space. Examples of one-class ML algorithms include [5,3]: Local Outlier Factor (LOF), One-Class Support Vector Machine (OC-SVM) and Isolation Forest (IF). More recently, Deep learning have been proposed for anomaly detection in diverse applications [7,30]. One popular deep learning model is the AutoEncoder (AE), which when compared with other one-class methods (e.g., LOF, OC-SVM and IF), tends to provide faster training times, thus are capable of handling a larger amount of training data. Another advantage of the AE algorithm is that it can be easily adapted to an online (or continual) learning, thus tackling better the concept drift phenomenon [24,25].

In anomaly detection, there is a recurrent problem, which is the sparseness of anomalous data when applied to a supervised approach. This sparseness makes it quite challenging to model machine learning models, as there are not many examples to feed into the model. In some studies, balancing techniques are used to generate enough minority examples for more robust models to learn (oversampling) or even to reduce the major-

ity class so that the model learns both classes in the same proportion (Undersampling) [23].

Within our knowledge, there is a lack of studies that perform a comparison of both supervised learning and one-class learning ML methods over several anomaly detection tasks, particularly extremely imbalanced ones. This paper fulfills this research gap by comparing the performance of state-of-the-art methods, namely RF, AutoML, IF and AE.

3 Materials and Methods

3.1 Datasets

This work experimented with seven public domain imbalanced datasets that can be tested for anomaly detection, namely:

- The **Predictive Maintenance Modeling Guide Collection (PMMGC)** consists of real-time telemetry readings and failure history acquired over the year of 2015 for 100 machines [28]. The full datasets are related with 876,142 hourly telemetry records (roughly 8,761 records per machine) captured by four sensors installed in each machine that measure tension (voltage), pressure, vibration, and rotation. The measurements were averaged using an hourly time period. In this paper, we selected five datasets from this collection and that are related with machines that contain more failures (the machine identification numbers are shown in Table 1). Each failure indicates the occurrence of a machine component replacement.
- The **PMAI4I** is another Predictive Maintenance dataset that was presented at the Artificial Intelligence for Industries conference (PMAI4I) [27]. It consists of a synthetic dataset that reflects a real predictive industrial maintenance task. The dataset contains 10,000 data points with one input attribute representing the type of product quality (categorical $\in$\{"low", "medium", "high"\}), five numerical inputs (air temperature, process temperature, rotational speed, torque and tool wear) and a target label indicating the occurrence of a machine failure.
- The **Credit Card Fraud Detection (CCFD)** data is related with transactions made by credit cards by European cardholders in September 2013 over two days [10]. The dataset contains a total of 283,726 transactions and 473 frauds. Due to confidentiality issues, the dataset contains only numerical features resulting from a Principal Component Analysis (PCA) transformation. The only input attributes that have not been transformed with PCA are: *Amount*, which indicates the transaction amount; and *Class*, which indicates whether the transaction was fraudulent or not.

Table 1 characterizes the adopted datasets. The last column (**Failures %**) shows that all datasets are highly imbalanced. In effect, the failure frequency that is lower than 1%, except for dataset PMAI4I (3.39%).

3.2 Data Preprocessing

For the PMMGC datasets, we have first removed all duplicate entries, since several of the selected machine datasets included repetitions of rows for the same anomalies.

Table 1. Summary of the adopted anomaly detection datasets.

Source	Dataset	Records	Input Features	# Failures	Failures(%)
PMMGC [28]	Machine 17	≈8,761	4	15	0.17
	Machine 22			15	0.17
	Machine 83			14	0.16
	Machine 98			16	0.18
	Machine 99			19	0.22
[27]	PMAI4I	10,000	6	339	3.39
[10]	CCFD	283,726	29	482	0.17

Moreover, the product quality type attribute of the PMAI4I dataset (the only categorical input feature of all the analyzed datasets) was transformed into a numeric one by using one-hot encoding, as implemented by the the `cane`[3] Python module[22,24]. This transformation assumes one binary input per categorical level, namely: "low" $\rightarrow$ (1,0,0), "medium" $\rightarrow$ (0,1,0) and "high" $\rightarrow$ (0,0,1). Regarding the same dataset, in order to increase the imbalanced ratio of the target class to a value that is closer to the percentage of failures of the other datasets, we have performed a random undersampling of the anomaly cases, leading to just 100 anomalies (of a total of 339), thus resulting in a failure rate of 1%.

3.3 Anomaly Detection Methods

All ML methods were implemented by using the Python language and the following modules: `scikit-learn`[4] – for IF and RF; `TensorFlow`[5] – for AE; and `H2O`[6] module for the AutoML.

The IF is a one-class ML algorithm that takes advantage of two significant characteristics of abnormal instances [20]: they are present in fewer quantities and are also numerically different to normal instances. The IF adopts this principle, constructing an ensemble of several isolation trees, each containing the abnormal instances closer to the root of the tree (Fig. 1). The `scikit-learn` IF implementation provides a decision score that ranges from $\hat{y}_i$ =-1 (highest abnormal score) to $\hat{y}_i$ =1 (highest normal score). In order to obtain an anomaly probability score ($d_i \in [0,1]$, for an input example i), we rescale the IF scores by computing $d_i = (1-\hat{y}_i)/2$.

Autoencoders (AE) are unsupervised learning techniques that efficiently compress and encode data into a lower-dimensional representation by assuming a bottleneck layer (with L_b hidden units) [16]. Let $(L_I, L_1, ..., L_H, L_O)$ denote the structure of a dense (fully connected) Deep FeedForward Network (DFFN) with the layer node sizes, where L_I and L_O represent the input and output layer sizes and H is the number of hidden layers. The

[3] https://pypi.org/project/cane/
[4] https://scikit-learn.org/stable/
[5] https://www.tensorflow.org/
[6] https://docs.h2o.ai/

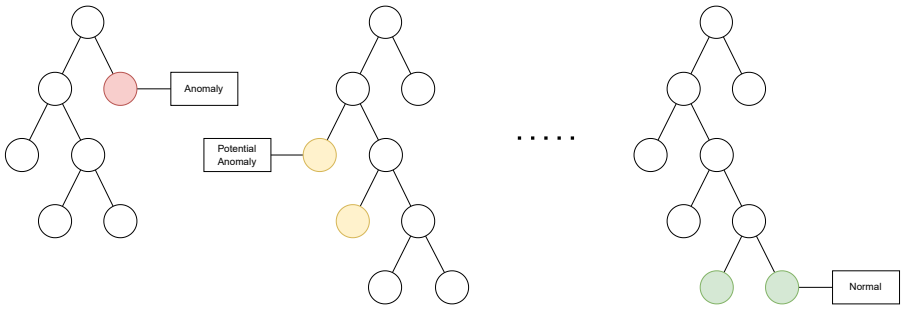

Fig. 1. Exemplification of the IF algorithm.

proposed AE is based on an architecture that previously obtained high quality anomaly detection results in a industrial anomaly detection task [26,34,35]. It assumes $L_I = L_O$, a symmetrical encoder and decoder structure (e.g., $L_1 = L_{O-1}$) and the popular ReLu activation function is used by all hidden neural units with the exception of the output layer, which assumes a linear activation function. In the encoder component, the number of hidden layer units decreases by half in each subsequent hidden layer until the bottleneck size (L_b) is reached: $L_1 = L_I/2$, $L_2 = L_1/2$, and vice-versa. Each hidden layer is also attached with a Batch Normalization (BN) layer. Fig. 2 represent the base structure used for all datasets in this work.

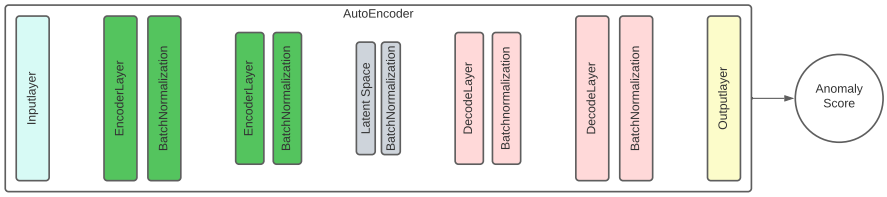

Fig. 2. Exemplification of the adopted base AE structure.

When adapted to anomaly detection, the AE training algorithm is only fed with standard (normal) instances, aiming to generate output values identical to its inputs. In this work, the AE is trained with the Adam optimizer using a batch size of 32, 200 epochs and early stopping (using 10% of the training data as the validation set). The Mean Absolute Error (MAE) is used as the loss function and reconstruction error: $MAE_i = \sum_{k=1}^{n} \frac{|x_{i,k} - \hat{x}_{i,k}|}{n}$,where $x_{i,k}$ and $\hat{x}_{i,k}$ denote the AE input and output value for the i-th data instance and k-th input or output node. The reconstruction MAE error is used as the decision score $d_i = MAE_i$, where higher reconstruction errors should correspond to a higher anomaly probability.

Turning to the supervised learning methods, the RF is a popular method that tends to obtain high quality prediction results when assuming its default hyperparameter values [11]. The algorithm works as an ensemble of decision trees that form a “forest". Each

tree depends on the values of a randomly sampled input vector and a bagging selection of training samples [4]. RF can be used for both regression and classification tasks. In this work, we used RF to output anomaly class probabilities ($\in$[0.0,1.0]), which are used as the decision score (d_i). As for the AutoML method, we adopt the H2O tool, which performs an automatic training and tuning of several ML algorithms within a user-specified time limit. Under the adopted H2O default configuration, the tool trains four distinct algorithms: RF, Generalized Linear Model (GLM), Gradient Boosting Machine and a default DFFN network. Then, it employs a Stacking Ensemble (SE), which uses all previously trained models to generate inputs for another GLM model. To perform the model selection, we set the tool to optimize the Area Under Curve (AUC) of the Receiver Operating Characteristic (ROC) analysis [12]. The tool was run for a maximum of 10 minutes, assuming an internal (applied over the training data) 5-fold cross-validation scheme.

3.4 Evaluation

To evaluate the anomaly detection performance, an external (applied over all available data) stratified 5-fold cross-validation scheme . During each of the five iterations, the training data was used to fit the ML model and the test (unseen) data was used to compute the predicted anomaly decision scores and respective ROC analysis [12]. When a model outputs a decision score d_i, the class can be interpreted as positive if $d_i > K$, where K is a fixed decision threshold, otherwise it is considered negative. The ROC curve shows the performance of a two-class classifier across all $K \in [0,1]$ values, plotting one minus the specificity (x-axis), or False Positive Rate (FPR), versus the sensitivity (y-axis), or True Positive Rate (TPR). The discrimination performance is given by the $AUC = \int_0^1 ROC dK$. It should be noted that the AUC measure is a popular measure that contains two main advantages [34,35]: quality values are not affected by the imbalanced rate of the target class; and quality values are easy to interpret (50% – performance of a random classifier; 70% - good; 80% – very good; 90% - excellent; and 100% - ideal classifier). After executing the 5-fold cross-validation, the AUC results are aggregated by computing the median value (which is less sensitive to outliers when compared withe the average value).

4 Results

In the first set of computational experiments, we executed the external 5-fold cross-validation for the tested seven datasets and four baseline ML algorithms (RF, AutoML, IF) and a generic AE structure. The results are summarized in Table 2. When comparing the supervised learning results (RF and AutoML), the H2O automated method tends to obtain a better anomaly detection performance. For instance, it presents a higher median PMMGC value (66.4% versus 62.5%) and also better median AUC values for the PMAI4I and CCFD tasks. Turning to the one-class methods, both IF and AE obtain the best median AUC values in three cases (IF – Machine 17 and 98, and CCFD; AE – Machine 22 and 99, and PMAI4I). Overall, AE produces a better discrimination accuracy

(in terms of median AUC) for PMMGC and PMAI4I, while IF obtains better CCFD results. More importantly, the comparison of supervised versus one-class methods tends to favor the latter ones. In effect, there is only one case (PMMGC Machine 83) where a supervised learning algorihm (RF) obtains the best anomaly detection performance. For all other cases, the IF and AE algorithms tend to outperform the supervised learning methods. This is an important result, since the one-class methods have the advantage of not requiring labeled data during their training procedure that is often more costly given the manual effort required to inspect and tag the examples.

Table 2. Median AUC values for the first comparison experiment (best results are in **bold**).

		Supervised		**Unsupervised**	
Dataset		RF	AutoML	IF	AE
PMMGC:	Machine 17	0.5000	0.6664	**0.6830**	0.6459
	Machine 22	0.6667	0.6661	0.7069	**0.7173**
	Machine 83	**0.6667**	0.6638	0.6114	0.4768
	Machine 98	0.6250	0.6250	**0.8155**	0.7979
	Machine 99	0.5000	0.4986	0.6818	**0.7047**
Median PMMGC value:		0.6250	0.6638	0.7069	**0.7173**
	PMAI4I	0.5997	0.6732	0.7644	**0.7914**
	CCFD	0.8838	0.8979	**0.9554**	0.8813

To further check if a balancing training technique would improve the supervised learning results, we executed a second set of experiments by adopting the five PMMGC datasets, for which the supervised learning anomaly detection performance is weak (e.g., random classification for RF and Machine 17). In these set of experiments, the RF and AutoML training data was first balanced by employing two oversampling techniques, SMOTE and CG, as implemented by the `imblearn` [19] and `sdv` [31] Python modules, using the baseline configurations. We note that in these experiments, the external 5-fold test data is kept without any changes. The obtained results are shown in Table 3.

Rather than improving the performance of supervised learning methods, both resampling techniques tends to diminish the anomaly detection capability. For instance, the median machine AUC value for RF decreased from 62.5% to 49.49% (SMOTE) and 50.0% (GC). Similarly, the AutoML median machine performance decreased from 66.4% to 50% (for both SMOTE and CG). The oversampling poor performance behavior might be explained by the extremely imbalanced nature of the analyzed datasets. Since the percentage of failures is lower than 1%, the SMOTE and GC methods have to generate a substantially high number of false synthetic cases, which creates records of the least representative class with the possibly of containing incorrect information that in turn prejudices the supervised learning.

This further enhances the quality of applying one-class learning instead of balancing the dataset. The one-class learning approach uses the class itself to establish the

Table 3. Median AUC values for the second comparison experiment using the PMMGC data source (best results are in **bold**).

Machine	SMOTE		Gaussian Copula	
	RF	AutoML	RF	AutoML
17	0.4949	0.4991	**0.6667**	0.6641
22	0.6584	**0.6624**	0.5000	0.4991
83	0.4949	0.4986	**0.5000**	**0.5000**
98	0.6561	**0.6621**	0.6250	0.4997
99	0.4874	0.4951	**0.5000**	0.4977
Median machine value:	0.4949	0.4991	**0.5000**	0.4997

boundaries that distinguish it from other classes, thus proving to be more useful than balancing extremely unbalanced datasets.

5 Conclusions

Due to advances in Information Technology (e.g., Smart Cities, Industry 4.0), anomaly detection is becoming an increasingly relevant task in diverse domains where digital data is in abundance (e.g., industry, finance, security). In this work, we perform an empirical comparison study that considers seven extremely imbalanced public domain datasets and four state-of-the-art Machine Learning (ML) algorithms: one-class – Isolation Forest (IF) and deep AutoEncoders (AE); and supervised – Random Forest (RF) and an Automated ML (AutoML) tool. Overall, the best anomaly detection results were obtained by the one-class learners (IF and AE), which have the advantage of not requiring labeled data during their training procedure thus proving to be more effective in detecting anomalies than applying dataset augmentation techniques (e.g., SMOTE, GC). In future work, we wish to extend the empirical study by implementing more robust data augmentation techniques, such as using deep Generative Adversarial Networks (GANs), to check if this produces benefits compared with other synthetic data generators (e.g., SMOTE and GC).

Acknowledgments

This work has been supported by the European Regional Development Fund (FEDER) through a grant of the Operational Programme for Competitivity and Internationalization of Portugal 2020 Partnership Agreement (PRODUTECH4S&C, POCI-01-0247-FEDER-046102).

References

1. Ahmed, M., Mahmood, A.N., Islam, M.R.: A survey of anomaly detection techniques in financial domain. Future Generation Computer Systems **55**, 278–288 (2016)

2. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data mining and knowledge discovery **29**(3), 626–688 (2015)
3. Alla, S., Adari, S.K.: Beginning Anomaly Detection Using Python-Based Deep Learning. Apress, Berkeley, CA (2019). https://doi.org/10.1007/978-1-4842-5177-5
4. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
5. Breunig, M.M., Kriegel, H., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. In: Chen, W., Naughton, J.F., Bernstein, P.A. (eds.) Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA. pp. 93–104. ACM (2000). https://doi.org/10.1145/342009.335388
6. Cao, N., Lin, Y.R., Gotz, D., Du, F.: Z-glyph: Visualizing outliers in multivariate data. Information Visualization **17**(1), 22–40 (2018). https://doi.org/10.1177/1473871616686635
7. Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407 (2019)
8. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 1–58 (2009)
9. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002). https://doi.org/10.1613/jair.953, https://doi.org/10.1613/jair.953
10. Credit Card Fraud - Kaggle: Anonymized credit card transactions labeled as fraudulent or genuine (2018), https://www.kaggle.com/mlg-ulb/creditcardfraud
11. Delgado, M.F., Cernadas, E., Barro, S., Amorim, D.G.: Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. **15**(1), 3133–3181 (2014), http://dl.acm.org/citation.cfm?id=2697065
12. Fawcett, T.: An introduction to ROC analysis. Patt. Recognition Letters **27**, 861–874 (2006)
13. Ferreira, L., Pilastri, A.L., Martins, C.M., Pires, P.M., Cortez, P.: A comparison of automl tools for machine learning, deep learning and xgboost. In: International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021. pp. 1–8. IEEE (2021). https://doi.org/10.1109/IJCNN52387.2021.9534091
14. Ferreira, L., Pilastri, A.L., Sousa, V., Romano, F., Cortez, P.: Prediction of maintenance equipment failures using automated machine learning. In: Yin, H., Camacho, D., Tiño, P., Allmendinger, R., Tallón-Ballesteros, A.J., Tang, K., Cho, S., Novais, P., Nascimento, S. (eds.) Intelligent Data Engineering and Automated Learning - IDEAL 2021 - 22nd International Conference, IDEAL 2021, Manchester, UK, November 25-27, 2021, Proceedings. Lecture Notes in Computer Science, vol. 13113, pp. 259–267. Springer (2021). https://doi.org/10.1007/978-3-030-91608-4_26
15. Grubbs, F.E.: Procedures for detecting outlying observations in samples. Technometrics **11**(1), 1–21 (1969)
16. Hinton, G., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006). https://doi.org/10.1126/science.1127647, cited By 9376
17. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial intelligence review **22**(2), 85–126 (2004)
18. Kumar, V.: Parallel and distributed computing for cybersecurity. IEEE Distributed Systems Online **6**(10) (2005)
19. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research **18**(17), 1–5 (2017), http://jmlr.org/papers/v18/16-365
20. Liu, F.T., Ting, K.M., Zhou, Z.: Isolation forest. In: Proc. of the 8th IEEE Int. Conf. on Data Mining (ICDM), Pisa, Italy. pp. 413–422. IEEE (2008)
21. Longadge, R., Dongre, S.: Class imbalance problem in data mining review. arXiv preprint arXiv:1305.1707 (2013)

22. Matos, L.M., Cortez, P., Mendes, R.: Cane - Categorical Attribute traNsformation Environment (2020), https://pypi.org/project/cane/
23. Matos, L.M., Cortez, P., Mendes, R., Moreau, A.: A comparison of data-driven approaches for mobile marketing user conversion prediction. In: Jardim-Gonçalves, R., Mendonça, J.P., Jotsov, V., Marques, M., Martins, J., Bierwolf, R.E. (eds.) 9th IEEE International Conference on Intelligent Systems, IS 2018, Funchal, Madeira, Portugal, September 25-27, 2018. pp. 140–146. IEEE (2018). https://doi.org/10.1109/IS.2018.8710472
24. Matos, L.M., Cortez, P., Mendes, R., Moreau, A.: Using deep learning for mobile marketing user conversion prediction. In: International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019. pp. 1–8. IEEE (2019). https://doi.org/10.1109/IJCNN.2019.8851888
25. Matos, L.M., Cortez, P., Mendes, R.C., Moreau, A.: Using deep learning for ordinal classification of mobile marketing user conversion. In: Yin, H., Camacho, D., Tiño, P., Tallón-Ballesteros, A.J., Menezes, R., Allmendinger, R. (eds.) Intelligent Data Engineering and Automated Learning - IDEAL 2019 - 20th International Conference, Manchester, UK, November 14-16, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11871, pp. 60–67. Springer (2019). https://doi.org/10.1007/978-3-030-33607-3_7
26. Matos, L.M., Domingues, A., Moreira, G., Cortez, P., Pilastri, A.L.: A comparison of machine learning approaches for predicting in-car display production quality. In: Yin, H., Camacho, D., Tiño, P., Allmendinger, R., Tallón-Ballesteros, A.J., Tang, K., Cho, S., Novais, P., Nascimento, S. (eds.) Intelligent Data Engineering and Automated Learning - IDEAL 2021 - 22nd International Conference, IDEAL 2021, Manchester, UK, November 25-27, 2021, Proceedings. Lecture Notes in Computer Science, vol. 13113, pp. 3–11. Springer (2021). https://doi.org/10.1007/978-3-030-91608-4_1
27. Matzka, S.: Explainable artificial intelligence for predictive maintenance applications. Proceedings - 2020 3rd International Conference on Artificial Intelligence for Industries, AI4I 2020 pp. 69–74 (2020). https://doi.org/10.1109/AI4I49448.2020.00023
28. Microsoft: Predictive maintenance modelling guide (2016), https://gallery.azure.ai/Collection/Predictive-Maintenance-Implementation-Guide-1
29. Muharemi, F., Logofătu, D., Leon, F.: Machine learning approaches for anomaly detection of water quality on a real-world data set. Journal of Information and Telecommunication **3**(3), 294–307 (2019)
30. Pang, G., Shen, C., Cao, L., Hengel, A.V.D.: Deep learning for anomaly detection: A review. ACM Comput. Surv. **54**(2) (mar 2021). https://doi.org/10.1145/3439950
31. Patki, N., Wedge, R., Veeramachaneni, K.: The synthetic data vault. In: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA). pp. 399–410 (Oct 2016). https://doi.org/10.1109/DSAA.2016.49
32. Pereira, P.J., Pereira, A., Cortez, P., Pilastri, A.L.: A comparison of machine learning methods for extremely unbalanced industrial quality data. In: Marreiros, G., Melo, F.S., Lau, N., Cardoso, H.L., Reis, L.P. (eds.) Progress in Artificial Intelligence - 20th EPIA Conference on Artificial Intelligence, EPIA 2021, Virtual Event, September 7-9, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12981, pp. 561–572. Springer (2021). https://doi.org/10.1007/978-3-030-86230-5_44
33. Rai, K., Devi, M.S., Guleria, A.: Decision tree based algorithm for intrusion detection. International Journal of Advanced Networking and Applications **7**(4), 2828 (2016)
34. Ribeiro, D., Matos, L.M., Cortez, P., Moreira, G., Pilastri, A.L.: A comparison of anomaly detection methods for industrial screw tightening. In: Gervasi, O., Murgante, B., Misra, S., Garau, C., Blecic, I., Taniar, D., Apduhan, B.O., Rocha, A.M.A.C., Tarantino, E., Torre, C.M. (eds.) Computational Science and Its Applications - ICCSA 2021 - 21st International Conference, Cagliari, Italy, September 13-16, 2021, Proceedings, Part II. Lecture Notes

in Computer Science, vol. 12950, pp. 485–500. Springer (2021). https://doi.org/10.1007/978-3-030-86960-1_34

35. Ribeiro, D., Matos, L.M., Moreira, G., Pilastri, A., Cortez, P.: Isolation forests and deep autoencoders for industrial screw tightening anomaly detection. Computers **11**(4) (2022). https://doi.org/10.3390/computers11040054
36. Ruff, L., Görnitz, N., Deecke, L., Siddiqui, S.A., Vandermeulen, R.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 4390–4399. PMLR (2018)