Enhanced Credit Prediction using Artificial Data

Peter Mitic[1] and James Cooper[2]

[1] Santander UK, 2 Triton Square, Regent's Place, London NW1 3AN
Dept. Computer Science, UCL, Gower Street, London WC1E 6BT
[2] Santander US, 75 State St, Boston MA, 02109
p.mitic@ucl.ac.uk,jsc42@cantab.net

Abstract. *Analysing credit data using a neural network has hitherto proved to be very resilient to attempts to improve success rates in prediction. We present a technique using simulated data which results in a marginal improvement in success rate. The empirical probability distribution for each feature of the training data is determined, and random samples are drawn from those distributions. The result is termed 'artificial' data. It is then possible to generate equal volumes of data for each of the binary outcomes (default or not), thereby alleviating a class imbalance classification problem. The simulation method uses a copula (to preserve the correlation structure of the original data) and optimal feature weighting to give acceptable results. The results indicate that overall percentage success rates for the more common outcome only are improved, but there is a more significant improvement in the AUC metric. The significance of this result in the context of assessing credit worthiness is discussed*

Keywords: Artificial data; Copula; Importance weight; Neural Network; Lorenz curve

1 Introduction

Recent advances in medical diagnoses using artificial intelligence (AI) have been remarkably successful. See, for example, Chabon, [5], Awan [2], Yala [19], and McKinney [13]. Overall classification success rates (i.e. total number of correct predictions divided by total number of predictions) exceeding 90% are common, with AUC values as high as 0.9.

Attempts to apply neural network technology to credit data have not hitherto proved to be as successful as the widely-used logistic regression methods that most lenders employ. Using the same technology (a neural network implemented in *Tensorflow*) that Google employed for the Chabon [5] study, it was difficult to achieve success rates of more than 74%. An attempt to explain this result was made in [14]. It appears that the indicators used when assessing credit worthiness, or combinations of them, are not strong pointers to future success in repayment.

In this paper we attempt to improve on the results reported in [14] using a variant of the *Probabilistic Novelty Detection* technique (hereinafter referred to as *PND*) developed by Clifton et al [6] to generate *artificial data*. The literature

review below summarises the principal drivers for this paper. Following that, our method of deriving and using *artificial data* is described. Result comparisons are then made, and explanations are offered.

In this paper, the most common outcome (i.e. the one with the most instances) will be referred to as the *major* outcome, and the least common outcome will be referred to as the *minor* outcome.

2 Literature Review: Credit Risk and Artificial data

In this review we concentrate on the application of novelty detection methods and to the assessment of credit worthiness. In doing so we provide some specific details of our previous work which provide a basis on which to improve.

AI with Credit Data: previous research

Earlier application of AI technology to credit data have yielded mediocre results compared to the recent medical successes already mentioned. For example, Louzada [12] quotes mean success rates of 77.7% for German credit data and 88.1% for Australian credit data (see [8]). Those figures mask success rates for the major and minor outcomes. Although 'better' results have been reported ([11]: AUC=0.915 and [1]: AUC=0.975), we suspect that either the data set used contains some behavioural indicator of default, or that loans in the dataset are only for 'select' customers who have a high probability of non-default.

Summary of the Metric Framework for Data Concentration

In [14] the first author explores whether the relative lack of success in using artificial neural networks to model credit risk may arise from inherent structures in the data. Three metrics (*Copula*, *Hypersphere* and *k-Neighbours*) are used to measure the 'shape' of the data. They are combined in a metric $\hat{H}$. It was observed that a high value of $\hat{H}$ implies that either the data are too noisy or that they provide insufficient predictive information to train a neural network.

The richness and complexity of the data comes from having different paths to success (or failure), which implies that there is little room to improve on initial results. Effectively, data corresponding to the major and minor outcomes appear to be almost coincident.

Summary of Probabilistic Novelty Detection

An general overview of Novelty Detection methods is given in [16]. This review concentrates on a specific example from that paper: the *PND* method of Clifton et al [6]. It is designed to cope with situations where the instances of the minor outcome are extremely rare, or even non-existent. Original data is used to define a hypersphere of radius $\boldsymbol{r}$, defined by the centroid of the real data. The data set for the major outcome is assumed to exist within a hypersphere of radius $2\boldsymbol{r}$ and the minor class is assumed to exist outside that radius (i.e. the minor class comprises outliers). The artificial data are used as a training set and the original data are used as the test set for an ensuing AI process.

Two sets of *PND* application results, both derived using SVM, are reported. Both show a clear separation between artificial data for the major and minor outcomes. Summarizing:

1. Combustion monitoring: AUC $\sim 0.81 - 0.96$
2. Patient vital sign monitoring. AUC ~ 0.9, indicated by ROC curves.

We have found that the *PND* method resulted in a deterioration of our previous results when applied to credit data. We suggest reasons in Section 4.3.

The statistical outlier detection method in [18] adopts a different approach. Outliers (equivalent to the 'minor outcome' set in the Clifton method) are determined by first dividing a training set into as many partitions as there are classes. An instance of a feature is considered an outlier if any feature value is more than three times the inter-quartile range from the third quartile for feature values in each partition. The German data set ([8]) mentioned in Section 2 was analysed in this way, and it was found that less than 10% of that data set could be considred as an outlier. The significance of this result is discussed in Section 6.

Subsequent research advanced the *PND* method further. Gorokhov et al [10] applied a convolutional neural networks to extract features from text data by sequentially filtering features from training and test sets (AUC=0.92). Pidhorskyi et al. [15] use a *generative-PND* method to compute the density function of image data on a training set, and generate samples from it (AUC ≥ 0.98 using the MNIST data). Two further studies adopt the same general approach: Rad et al [17] (mobility assessment, AUC $\in (0.65, 0.95)$), and Contreras et al [7] (robotics, 77% of predictions exceeded 90% accuracy). Bhattacharjee et al [3] treats data that cannot be classified with confidence as 'novelties' (image classification, AUC $\in (0.77, 0.90)$)

3 Methodology: Artificial Data generation and use

We have found that the algorithm presented in [6] did not produce satisfactory results. Reasons are suggested in Section 4. Therefore we have developed an alternative, the overall strategy for which is summarised the following algorithm. The step numbers correspond to the steps in Figure 1.

1. Partition the original data into training and test sets, D_{train} and D_{test} respectively (Step A).
2. Partition D_{train} into two subsets $D_{train,0}$ and $D_{train,1}$ according to the binary outcomes 0 and 1 respectively (Step A).
3. Generate artificial data $D_{art,0}$ and $D_{art,1}$ from the subsets $D_{train,0}$ and $D_{train,1}$ respectively (Steps B and C).
4. Combine $D_{art,0}$ and $D_{art,1}$ to form a single artificial data set D_{art} (Step D).
5. Use D_{art} for training and D_{test} for testing.

The steps above are summarised in Figure 1. The source of importance weights is discussed in Step B.1 of the detailed algorithm. (Section 3.1). The numbers in black roundals refer to the steps in Section 3.1.

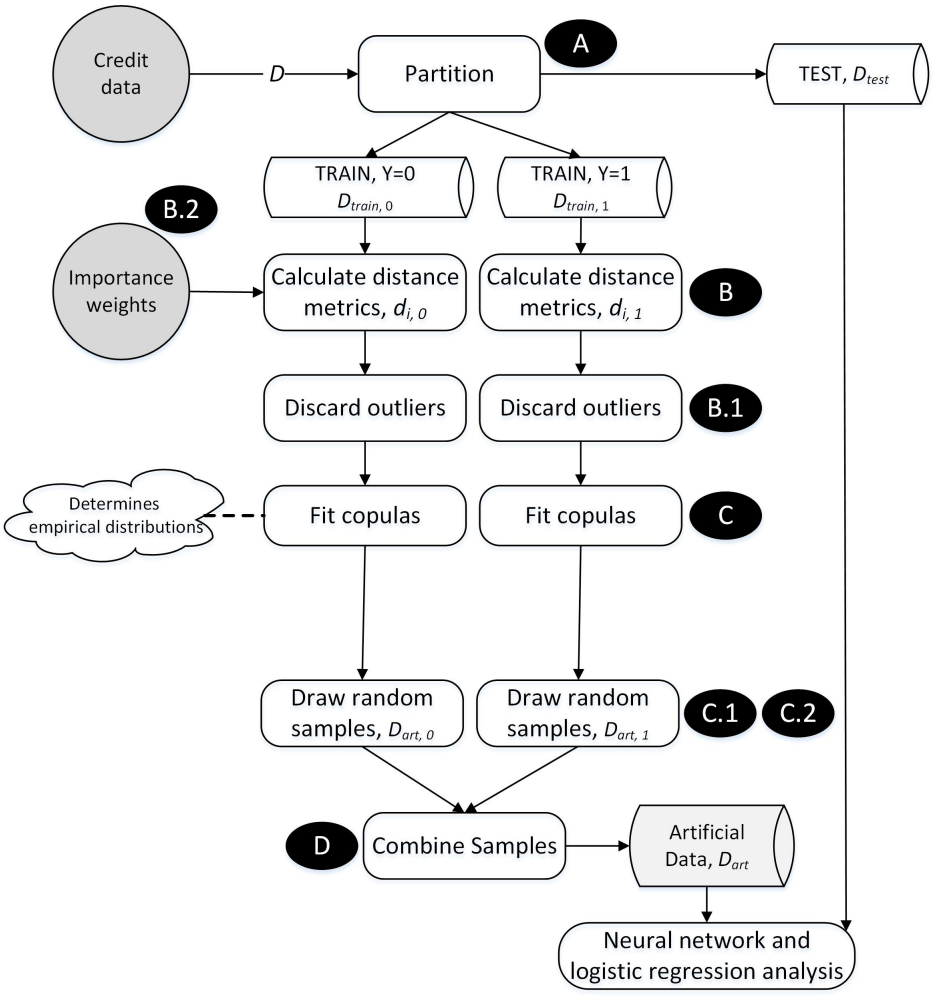

Fig. 1. Artificial Data generation algorithm. The step numbers in black roundals refer to sub-sections in 3.1.

3.1 Artificial Data algorithm: Details

The details of our algorithm to generate artificial data are summarised in the steps that follow. The starting point is a dataset comprising N feature columns labelled $X_1, X_2, ...X_N$. The outcome column is labelled Y and takes values zero for the major outcome (correct prediction-credit pass) and one for the minor outcome (incorrect prediction-credit fail). The data are imbalanced: the number of major outcomes is approximately 1/3 of all outcomes.

Step A
Partition the original data D such that there is sufficient data in each partition to model the empirical data accurately. In our case, four partitions $P_{01}, P_{02}, P_{03}, P_{04}$ for the major outcome $Y = 0$, and two partitions P_{11}, P_{12} for the minor outcome $Y = 1$.

Step B
The empirical distribution of each of the six partitions was determined by formulating a histogram based on the values of each feature. The method to formulate the histogram is described in *Step B.1*. Empirical distributions were the most generally applicable across all features (categorical and non-categorical). The outputs of this step are labelled $D_{01}, D_{02}, D_{03}, D_{04}, D_{11}, D_{12}$, corresponding to the partitions in *Step A*. The histogram comprises metrics d_i (Equation 1). In our case, $N = 22$.

$$d_i = \sum_{j=1}^{N} w_j (M_j - x_{ij})^2 \tag{1}$$

Each corresponding empirical distribution $E_{ij} (i \in (1, 2, 3, 4); j \in (0, 1))$ is characterised by a vector of feature values and corresponding relative frequencies.

Step B.1
It was found that outliers diminish the prediction accuracy considerably. The outliers correspond to empirical distributions E_{40}, E_{30} and E_{21} and are discarded.

Step B.2
Importance weighting plays a significant part in determining the distributions D_{ij}. For each feature, a distance metric d_i is defined in equation 1 above. This metric is the sum of the deviation of each feature value x_{ij} (datum i with feature j) from the mean of all values for feature j, multiplied by an importance weight for that feature, w_j (clarified below).

Of the importance weighting schemes considered, two were more significant than others. The most significant (termed *ISSE* - Inverse Sum of Squared Expectation) used the inverse of the sum of residuals of a logistic regression fit to data. Importance weights derived using the Boruta algorithm also worked well. The *ISSE* importance weights, w_i are calculated from Equation 2 which summarises the *ISSE* calculation for a logistic regression function ρ acting on each

of N features in the training data $x_i[Train]$ and outcome y_i, with a logistic regression prediction function *Pred* which takes test data $x_i[Test]$ as an additional argument.

$$w_i = \left(\sum_{j=1}^{N} Pred\Big(\rho\big(x_i[Train], y_i\big) - x_i[Test]\Big)^2 \right)^{-1} \tag{2}$$

Step C
Fit a copula C_{ij} to pseudo observations of each partition P_{ij}. The copula preserves the dependency structure of the features of the original data. The *Normal* and *Frank* copulas proved to be optimal.

Step C.1
Uniformly distributed random samples U_{ij} were extracted from each copula C_{ij}. The sample size was set for each partition so as to be sufficient to generate enough artificial data to use in a neural network and to produce approximately the same number of '$Y = 0$' cases as '$Y = 1$' cases. It was found that using partitions P_{12}, P_{03}, P_{04} resulted in diminished results, and sample sizes of 1 were allocated to these sets.

Step C.2
The random samples U_{ij} were transformed to the appropriate empirical distributions D_{ij} using inverse empirical distribution function transformations.

Step D
The outputs of the previous step were combined columnwise. This combination constitutes the artificial data.

4 Results

4.1 Data and Implementation

The data set used was the data set labelled *INT* in [14]. It comprises 8202 records: 2690 records for the minor outcome $Y = 1$ (credit fail), and 5512 for the minor outcome $Y = 1$ (credit pass). Each record had $N = 22$ features, each normalised to [0,1], and a binary decision flag Y. Calculations were done using R on an i7 processor with 16MB RAM. We are grateful for the *Tensorflow* neural network code supplied by Chollet and Allaire in [4].

4.2 Copula and Importance Weighting Results

In order to choose an importance weighting scheme for *Step B* of the Artificial Data algorithm, the overall algorithm at the start of Section 3 was run with the most generally applicable copula (the *Normal* copula), cycling through a

range of importance weighting schemes. Repeated trials showed that *ISSE* importance weighting (see Section 3.1, *Step B.1*) was optimal (AUC=0.865), and produced particularly stable results. The *Boruta* method was almost as good (AUC=0.845). The AUC without importance weighting was 0.649, so is not a viable option. Other weighting schemes tested were *Principal Components*, *Pseudo-R^2*, *Recursive Feature Elimination*, *Log-Likelihood ratio*, *Random Forest*, *Logistic Regression* and *LVQ*.

Given the optimal *ISSE* choice, the copulas tested were *Normal*, *Student-t*, *Joe*, *Clayton*, *Gumbel* and *Frank*. There was very little variation between them, and the *Frank* copula was optimal (AUC=0.871). The Frank copula stresses outlier and near-origin data more than the others, which may explain its optimality.

4.3 Results using Artificial Data

Table 1 shows a comparison of neural network and logistic regression results with original data only, with data derived from the *PND* method [6], and with data derived from our *Artificial Data* method. The mean and standard deviation results for 25 runs using each method are shown.

Table 1. Neural network and logistic regression results (Mean, SD), using the Artificial data method (see note 1), the Probabilistic Novelty Detection method (see note 2), and with original data exclusively (see note 3).

Method	*Metric mean*	Original data	*PND*	Artificial data
NN	% Success	(65.91, 6.88)	(95.89, 0.86)	(77.26, 4.89)
NN	% Success Major	(65.83, 7.01)	(98.5, 0.98)	(76.74, 6.89)
NN	% Success Minor	(74.07, 7.25)	(5.18, 3.98)	(77.27, 5.18)
NN	Gini	(0.63, 0.01)	(0.23, 0.07)	(0.72, 0.03)
NN	AUC	(0.82, 0.01)	(0.61, 0.04)	(0.86, 0.01)
LR	% Success	(72.02, 1.54)	(97.11, 0.06)	(55.43, 4.8)
LR	% Success Major	(72.13, 1.71)	(99.87, 0.1)	(54.58, 5.02)
LR	% Success Minor	(69.6, 5.19)	(1.11, 1.79)	(85.04, 3.46)
LR	Gini	(0.52, 0.04)	(0.24, 0.03)	(0.65, 0.04)
LR	AUC	(0.76, 0.02)	(0.62, 0.02)	(0.83, 0.02)

Note 1: Artificial Data. Frank copula, *ISSE* importance weighting, 2000 major outcome data, 5000 minor outcome data. 25 runs, each $\sim$ 10 minutes
Note 2: PND, with parameters defined in [6] da=0.25, dn=0.01, $r_a = 3\boldsymbol{r}$. 2000 records generated, 10 runs, each $\sim$ 5 hours
Note 3: Results with original data only, from [14]. LR training sets were obtained by random sampling.

The results in Table 1 indicate that using *Artificial Data* gives an improvement on the results derived using original data only. In particular, the balance

between % success for the major and minor outcomes is preserved using a neural network. If logistic regression is used instead, it is not. In contrast, there is a marked deterioration of results using *PND*. We suggest that the reason is some or all of the following points.

- The dependency structure of the original data is not preserved.
- It is assumed that the minor outcome corresponds to outliers, as defined by the hypersphere. That is unlikely to be the case for credit data.
- There is no clear way to tune the model parameters.
- There is an over-dependence on uniformly-distributed data. Only a few credit data feature distributions resemble uniform distributions.

In contrast, our *Artificial Data* set is specifically designed to preserve the dependency structure of the original data, and models individual features for the major and minor outcomes as closely as possible.

5 Discussion: analysis of the Lorenz curve

We now consider an alternative approach, in the context of credit risk, to measuring 'success' by AUC or % of correct predictions. Lorenz curves are a useful tool to measure, in the context of credit risk, the proportion of predictive success in the binary outcomes $Y = 0$ and $Y = 1$. More often they are used to quantify economic inequality: proportion of income against proportion of population. See a recent discussion in [9]

A Lorenz curve is a plot, parameterised by threshold, of modelled propensity against % minor outcome class included up to a given threshold (horizontal) and % major outcome class included up that threshold (vertical). Lorenz curves are well established for visualizing the ability of a model to rank order by likelihood of default. A perfect rank ordering would start at the origin, rise vertically as it works through the major class, and then horizontally across the top of the unit square (note, there is no requirement for the propensity cut off to be 0.5). The power of such a model is given by gini (=2*AUC-1). Gini values lie between -1 and 1, with 0 representing random selection and negatives a reverse ordering). Modelling is typically geared towards maximizing the gini, because of a broad relationship between gini and the capital a bank needs to hold for credit risk exposure. Figure 2 shows a hypothetical Lorenz curve.

However, the practical use of this model is often focused on a particular region. For illustration:

- Always lend to people with a predicted default probability less than 1%,
- Never lend to people with a predicted default probability more than 5%.

So in terms of decisioning, the area of the curve near $p = 3\%$ might be critical. It shows how different the population between $p = 3\%$ and $p + \Delta = 4\%$ looks compared to the population between $p - \Delta = 2\%$ and $p = 3\%$. In that way we get a sense of the performance of the model at the decision boundary. This

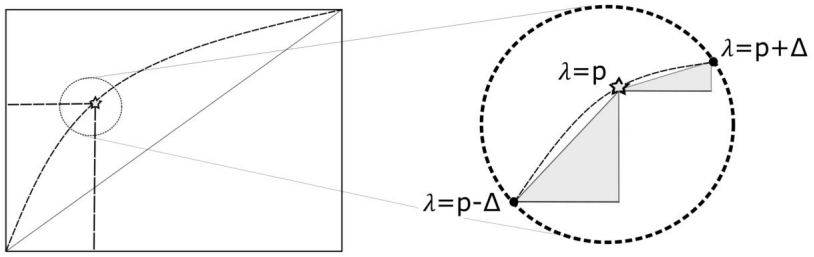

Fig. 2. Lorenz curve illustration, showing a gradient discontinuity near a typical decision boundary. The axes are explained in the associated text.

may be thought off as the difference in gradient of line segments as shown in Figure 2. The flatter the gradient the higher the local density of defaults per non-defaulter. Although not considered in this paper, understanding the effect near the decision boundary would be required for implementation. The point of such analysis would be to reduce the incidence of false-negatives, which cause far more harm to a bank than false-positives.

6 Conclusion

In this paper we have attempted to improve upon a previous result obtained when applying neural network technology to credit data. Using *Artificial Data* has made it possible to improve the previous result marginally, in terms of both AUC and success rates. Correct predictions of the minor outcome (credit fail) is a major factor in credit analysis, since every defaulted loan requires multiple non-defaulted loans to compensate for any shortfall incurred. Therefore a valuable theme to pursue is to improve on the minor outcome success rate without compromising the major outcome, using the idea suggested in Section 5.

In section 2 a method of outlier detection ([18]) was noted. The particular case of the German credit data ([8]) has a bearing on the results of this paper. In that case, less than 10% of instances were classified as outliers. We consider, following analysis ising the Novelty Detection method of [6], that there are similarities between the data used in our analysis and the German credit data. Specifically, outliers cannot be used to generate artificial data, because outliers are sparse. The subsets corresponding to 'credit fail' and 'credit pass' are almost coincident. Outliers comprise a mixture of the two subsets.

References

1. Addo, P.M., Guegan, D. and Hassani, B. Credit Risk Analysis Using Machine and Deep Learning Models, Risks 6(38), doi:10.3390/risks6020038, 2018
2. Awan R., Koohbanani N.A., Shaban M., Lisowska A., Rajpoot N. Context-Aware Learning Using Transferable Features for Classification of Breast Cancer Histology

Images. In: Campilho A., Karray F., ter Haar Romeny B. (eds) Image Analysis and Recognition. ICIAR 2018. LNCS 10882, 2018
3. Bhattacharjee,S., Mandal,D. and Biswas,S. Multi-class Novelty Detection Using Mix-up Technique. Proc. WACV2020 DOI: 10.1109/WACV45572.2020.9093303, 2020
4. Chollet,F and Allaire, J.J. Deep Learning with R. Manning NY, 2018
5. Chabon, J.J., Hamilton, E.G., Kurtz, D.M. et al. Integrating genomic features for non-invasive early lung cancer detection. Nature 580, 245–251. https://doi.org/10.1038/s41586-020-2140-0, 2020
6. Clifton, L., Clifton, D.A., Zhang, Y., Watkinson, P., Tarassenko, L. and Yin, H. Probabilistic novelty detection with support vector machines. IEEE Transactions on Reliability, 63(2), pp.455-467, 2014.
7. Contreras-Cruz,M.A., Ramirez-Paredes,J.P., Hernandez-Belmonte,U.H., and Ayala-Ramirez,V. Vision-Based Novelty Detection... Sensors 19, 2965; doi:10.3390/s19132965, 2019
8. Dua, D. and Graff, C., Statlog Data: UCI Machine Learning Repository Irvine CA, http://archive.ics.uci.edu/ml, 2019
9. Costa,R.N. and Perez-Duarte, S. Not all inequality measures were created equal. European Central Bank Statistics Paper Series 31, https://www.ecb.europa.eu//pub/pdf/scpsps/ecb.sps31 269c917f9f.en.pdf, 2019
10. Gorokhov, O, Petrovskiy,M and Mashechkin,I. Convolutional Neural Networks for Unsupervised Anomaly Detection in text data. 18th IDEAL conference Guilin China (pp. 500-507). LNCS 10585 Springer (eds. Yin,H., Gao,Y., Chen,S. et al) 2017
11. Kvamme, H., Sellereite, N., Aas, K. and Sjursen, S., Predicting mortgage default using convolutional works, Expert Systems with Applications 102: pp207-217, https://doi.org/10.1016/j.eswa.2018%.02.029, 2018
12. Louzada, F., Ara, A. and Fernandes, G.B., Classification methods applied to credit scoring, Surveys in Operations Research and Management Science 21(2):pp117-134, https://doi.org/10.1016/j.sorms.2016.10.001, 2016
13. McKinney, S.M., Sieniek, M., Godbole, V. et al. International evaluation of an AI system for breast cancer screening. Nature 577, 89–94, https://doi.org/10.1038/s41586-019-1799-6 , 2020
14. Mitic, P. A Metric Framework for Quantifying Data Concentration. 20th IDEAL conference Manchester UK (pp. 181-190). LNCS 11872 Springer, (eds. Yin,H., Camacho D., Tino P. et al), 2019
15. Pidhorskyi,S., Almohsen,R., Adjeroh,D.A. and Doretto,G. Generative Probabilistic Novelty Detection with Adversarial Autoencoders. Proc. NIPS'18. Montreal Canada, 6823–6834, https://dl.acm.org/doi /10.5555/3327757.3327787, 2018
16. Pimentel,M., Clifton,D.A., Clifton,L. and Tarassenko,L. A Review of Novelty Detection. Signal Processing 99, 215–249, 2014
17. Rad,N.M., van Laarhoven,T., Furlanello,C. and Marchiori,E. Novelty Detection using Deep Normative Modeling...Sensors 18, 3533, doi:10.3390/s18103533, 2018
18. Tallon-Ballesteros, A. J., and Riquelme, J. C. Deleting or keeping outliers for classifier training?. Sixth World Congress on Nature and Biologically Inspired Computing (NaBIC) (pp. 281-286). IEEE, 2014
19. Yala, A., Lehman, C., Schuster, T., Portnoi, T. and Barzilay, R., A Deep Learning Mammography-based Model for Improved Breast Cancer Risk Prediction, Radiology Online 07/05/2019, https://doi.org/10.1148/radiol.2019182716, 2019