

Fraud detection: contribution of complementary strategies to traditional methods

D. Behar , O. Claeys, A. Couloumy, Z. Zouira



1. Context

Business case

- The business tasks consist in **highlighting fraud** within a dataset that contains characteristics of claims individuals (daily allowances health insurance contracts).
- Fraud team have already developed a rule based approach to assess fraud probability. This was based on **6 criteria**. Such techniques are limited: rules are too simple and do **not allow any proactivity**
- A first work was developed using **pre annotated data and supervised ML**. Results are presented below:

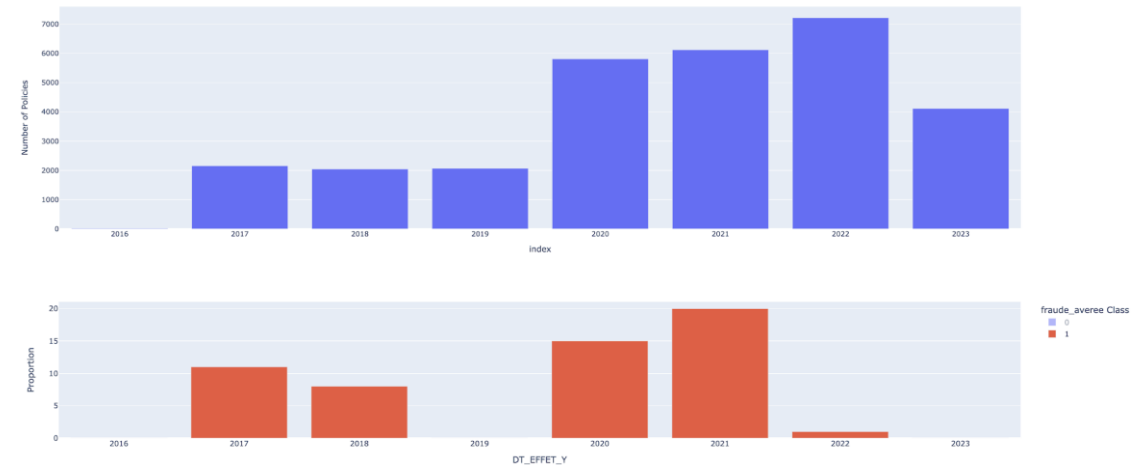
Performances	
Recall - TPR	35,7%
Precision	5,7%
Accuracy	98,9%
F1 Score	9,8%

- The idea of this second batch is to provide more capabilities by:
 - Getting **better results** (specially regarding recall)
 - Providing **more interpretability**, nuances, and allowing the identification and prioritization of underlying potential frauds.

1. Context

Data preparation

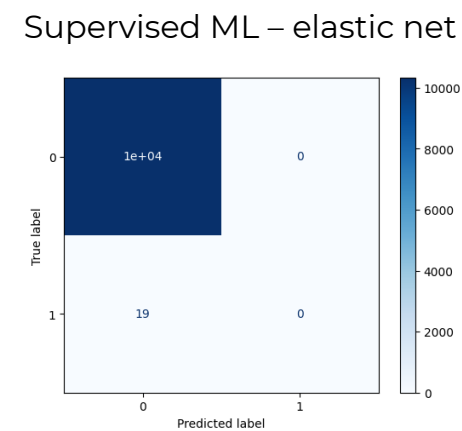
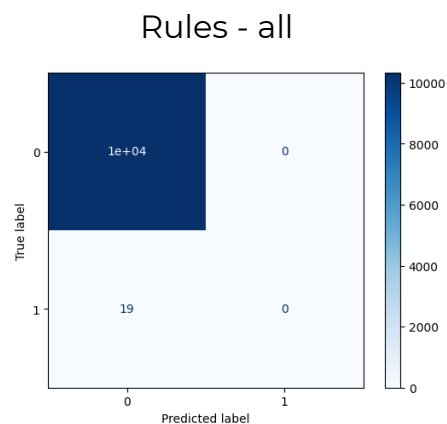
- Initial dataset contains **29k rows and 100 columns** that describes claims (daily allowance). Most of the variables are numerical (60%).
- Main steps of **data processing** are:
 - Delete a part of feature
 - Force variable type to be categorical or numerical
 - Default missing data imputation : mean and mod
 - Standardization, one hot encoding, re discretization
- Dataset is very unbalance, sparse through time.
- Strategy for splitting train/test :
 - Using a **random stratify sample** of 35% observation as test set
 - Initially another historical train/test split was experimented



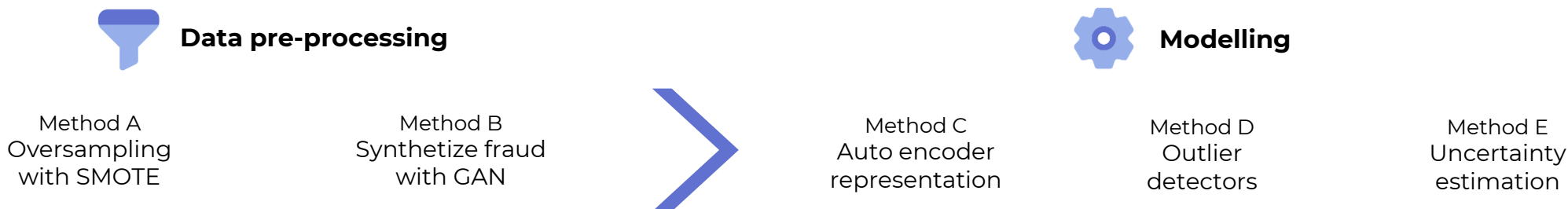
1. Context

Baseline vs New strategy

- We take back the **initial baseline approaches** (rules and initial elastic net model) and we evaluate with test set.



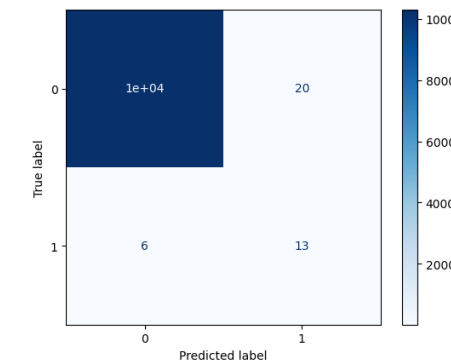
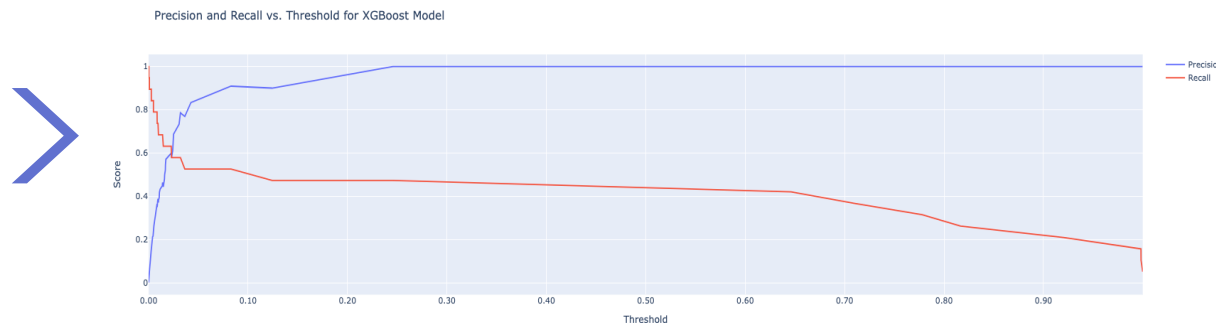
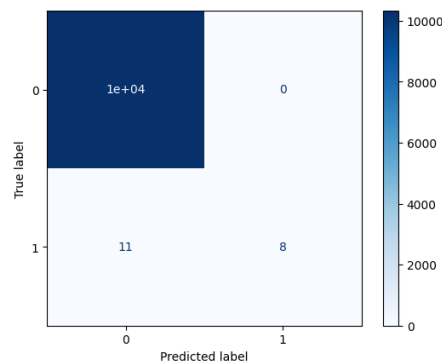
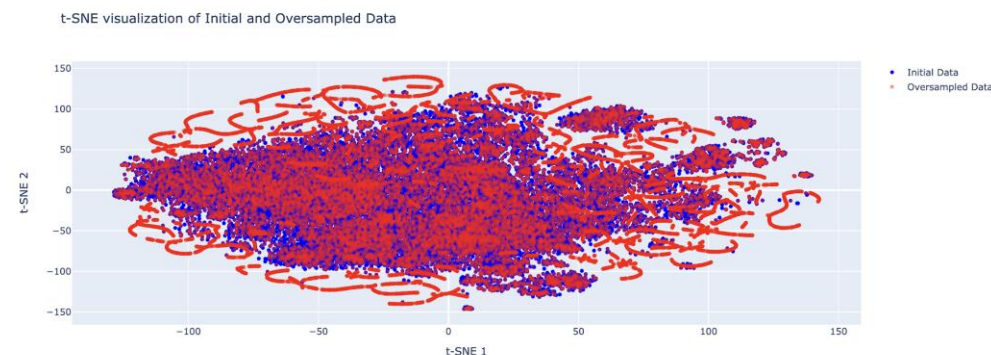
- First we are going to experiment **five techniques** at different steps of the process:



2. Data processing

Method A – Oversampling

- Considering dataset is **very unbalance** (few fraud observations) we want to extend examples using **oversampling**.
- To do so we can apply some **SMOTE** [1] approaches. First we simply explore default SMOTE method by comparing new data sampled (in red) with the initial one (in blue). We use t-SNE [2] dimension reduction.
- After this exploration we define a supervised ML model (XGBoost), exploring optimal parameters using grid search. We obtain:



- Looking at matrix confusion and recall depending on threshold, we finally get **68% fraud recall**.

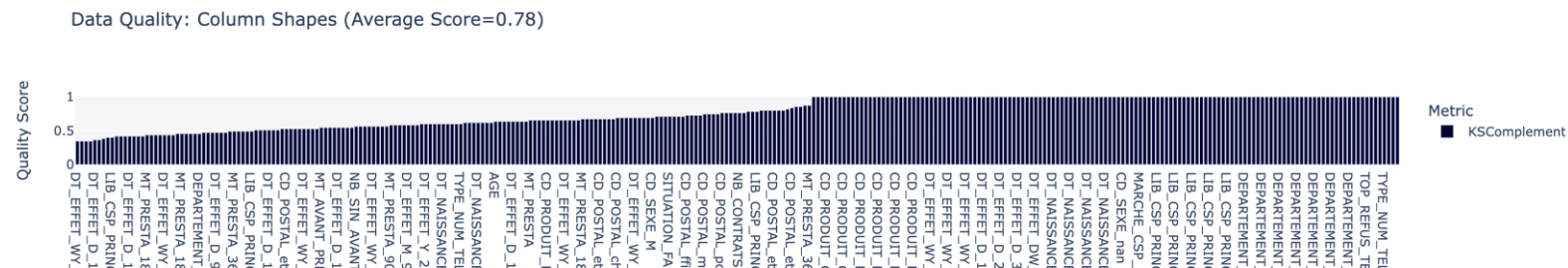
2. Data processing

Method B – Synthetic Data with GAN

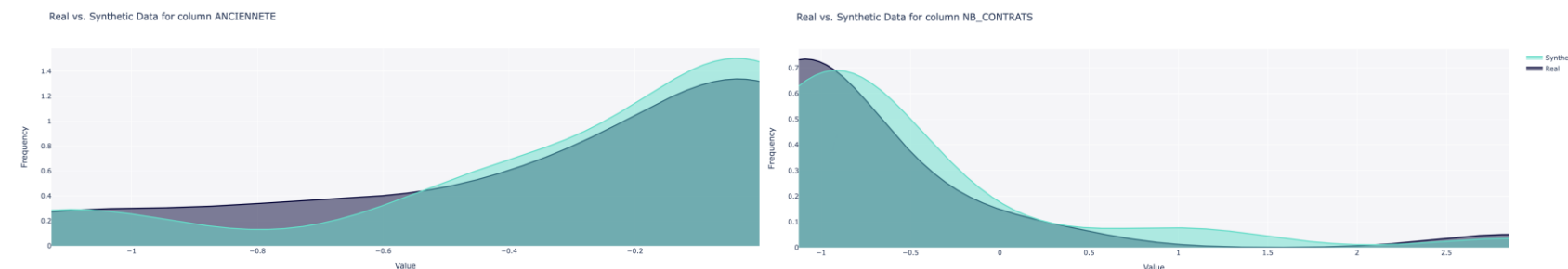
- Another method is to use **Generative Adversarial Network (GAN)** [5] [6] to generate synthetic fraud that will help training.
- We create a CTGAN [7] [8], to generate fake data on train set. We study quality of synthetiser using :

```
Overall Quality Score: 83.87%

Properties:
Column Shapes: 77.66%
Column Pair Trends: 90.08%
0.8387194350650711
```



- After iteration of CTGAN parameters we create a synthetic fraud sample of 20 000 observations we are going to use later.



[5] Little, Claire, Mark Elliot, Richard Allmendinger, and Sahel Shariati Samani. "Generative Adversarial Networks for Synthetic Data Generation: A Comparative Study." arXiv preprint arXiv:2112.01925 (2021). https://unece.org/sites/default/files/2021-12/SDC2021_Day2_Little_AD.pdf

[6] Xu, Lei, and Kalyan Veeramachaneni. "Synthesizing tabular data using generative adversarial networks." arXiv preprint arXiv:1811.11264 (2018). <https://arxiv.org/abs/1811.11264>

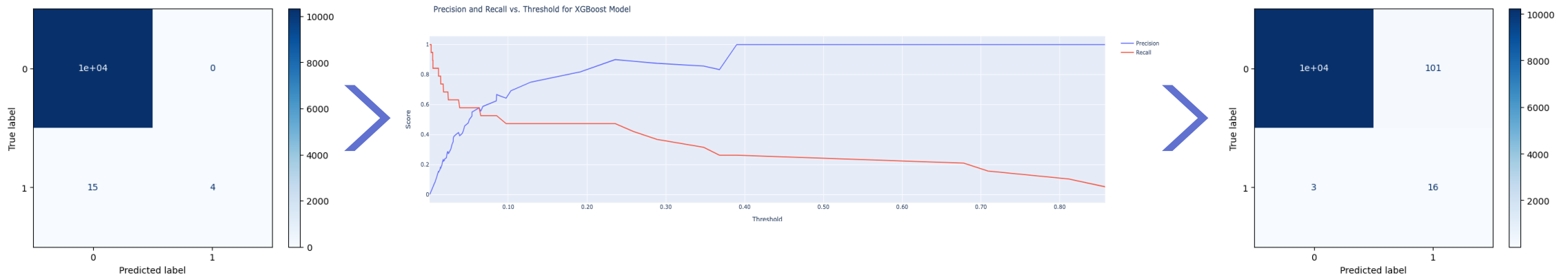
[7] Xu L, Skoularidou M, Cuesta-Infante A, Veeramachaneni K. Modeling tabular data using conditional gan. Advances in Neural Information Processing Systems. 2019. <https://arxiv.org/abs/1907.00503>

[8] Zhao, Zilong, Aditya Kumar, Robert Birke, and Lydia Y. Chen. "CTAB-GAN+: Enhancing Tabular Data Synthesis." arXiv preprint arXiv:2204.00401, 2022. <https://arxiv.org/abs/2204.00401>

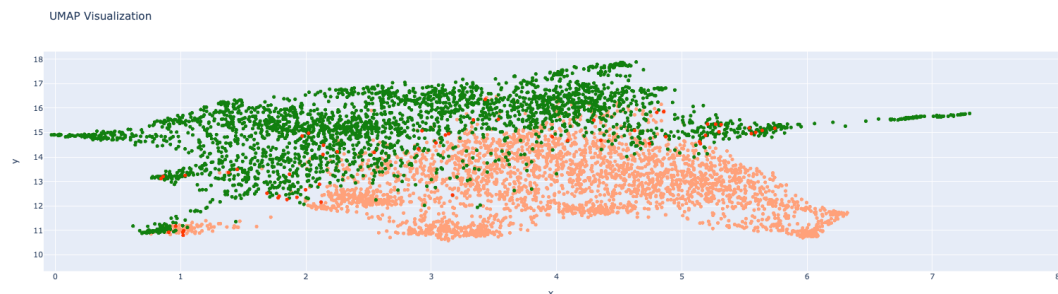
2. Data processing

Method B – Synthetic Data with GAN

- We have trained an **Xgboost using synthetic fraud data** in training set. We want to compare results to SMOTE like techniques.
- Initial fraud recall is lower than SMOTE but threshold optimization allows to obtain a good results (84%) Precision on contrary is lower with CTGAN.



- We can also visualize non fraud, fraud and synthetic fraud on 2D UMAP plan:

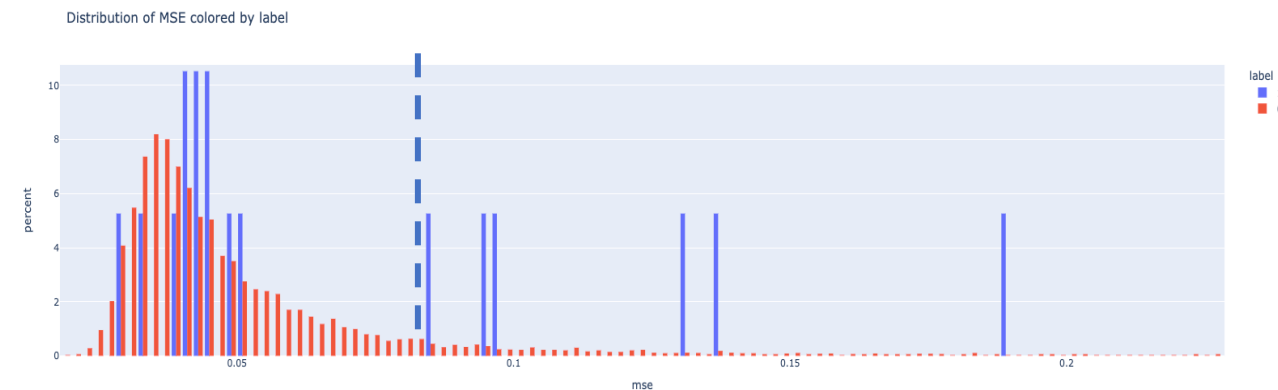


3. Modelling

Method C – Auto-encoder

- To dive a deeper on modelling aspects, we explore **auto-encoder neural networks** [9] [10] with two goals:
 - Be able to identify fraud using a reconstruction loss threshold
 - Use latent Auto encoder representation to facilitate supervised tasks
- To train the model we consider the following hypothesis:
 - We train only on non fraud data
 - We use Adamax optimizer and MSE as loss function
 - We consider encoding and decoding parts of 10 layers and latent encoding has a dim of 12
 - We introduce drop out to facilitate generalization.
- We study **reconstruction loss** of each group. We expect having a small loss for non fraud and higher losses for fraud.
- We want to have **individuals losses separated enough** between fraud/ non fraud to define a threshold that may help in the detection. We can pick up 0.07.
- We create a confusion matrix to compare methods.

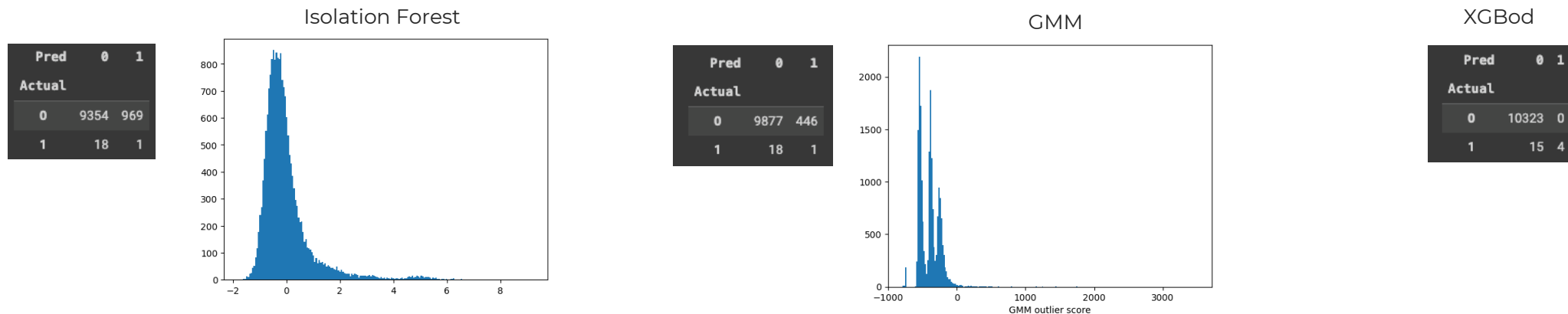
threshold_result	0	1	Total
label			
0	8774	1549	10323
1	11	8	19
Total	8785	1557	10342



3. Modelling

Method D – Outlier detector

- Then we explore several techniques to detect outliers (considering a statistical outlier may be a fraud – or not..).
- We explore the following techniques (not exhaustive): Isolation forest [11], Gaussian mixture model (GMM) [12], XGBod [13]. For each technique we can first explore outlier score distribution and deduce outlier thresholds:

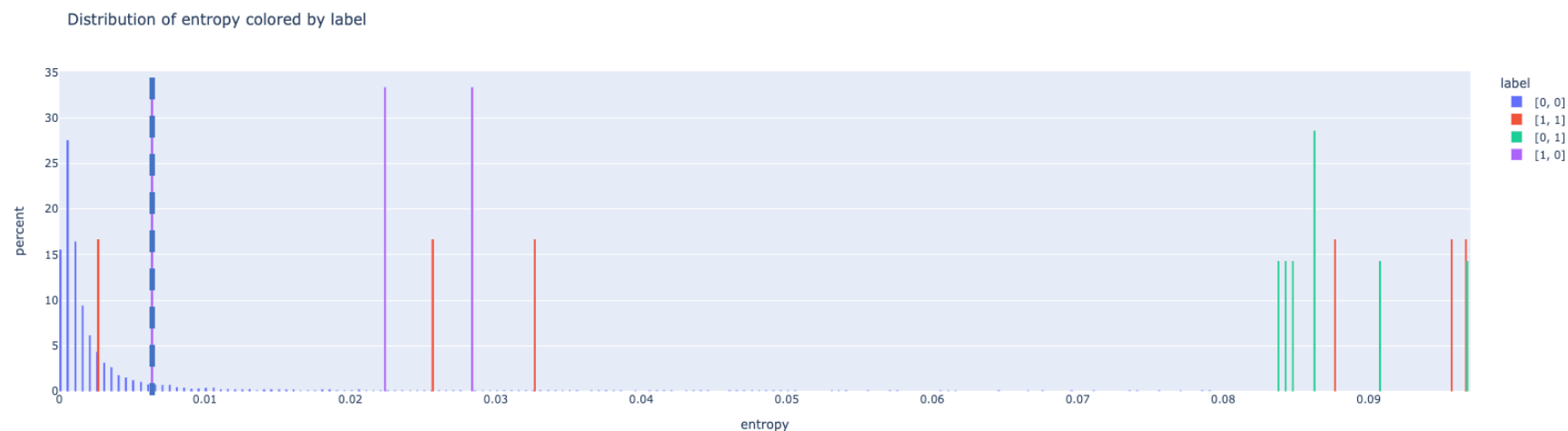


- If we use outlier score as rule to define fraud / non fraud, we can also compute a kind of confusion matrix.
- We notice that XGBod model may be helpful to highlight fraud cases (21% recall). Other models are bad.

3. Modelling

Method E – Uncertainty

- It is not a method in itself but using **uncertainty measure** [14] on supervised classifier (or using dedicated techniques like Bayesian neural networks [15]) may be helpful to provide a better understanding of modelling results
- In our case (for simplicity reason and time constraint) we introduce a **entropy measure** to computer uncertainty.
- We illustrate **distribution of entropy** according different classes:
- We can define a threshold like 0.006 for instance to improve understanding of fraud cases.



4. Mixed techniques

Method A + C – Oversampling & Auto-encoder

Method A
**Oversampling
 with SMOTE**

Method B
 Synthesize fraud
 with GAN



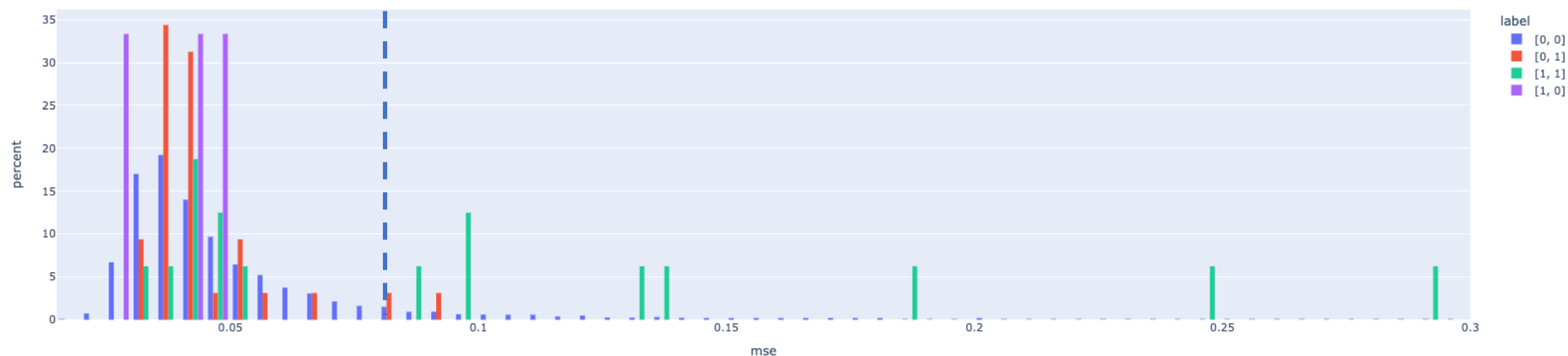
Method C
**Auto encoder
 representation**

Method D
 Outlier
 detectors

Method E
 Uncertainty
 estimation

- Let us mix former techniques.
- We consider as prerequisite the fact we have selected a default supervised ML technique, fine tuned (XGBoost)
- Regarding results of **oversample supervised approach and reconstruction loss** distribution, we notice that true fraud (green) may have high losses, while wrong normal (red), true normal (blue) have small losses (logical) and wrong fraud also have small losses (purple). This is not very helpful.

Distribution of MSE colored by label



4. Mixed techniques

Method A + D – Oversampling & Outliers

Method A
Oversampling
with SMOTE

Method B
Synthesize fraud
with GAN



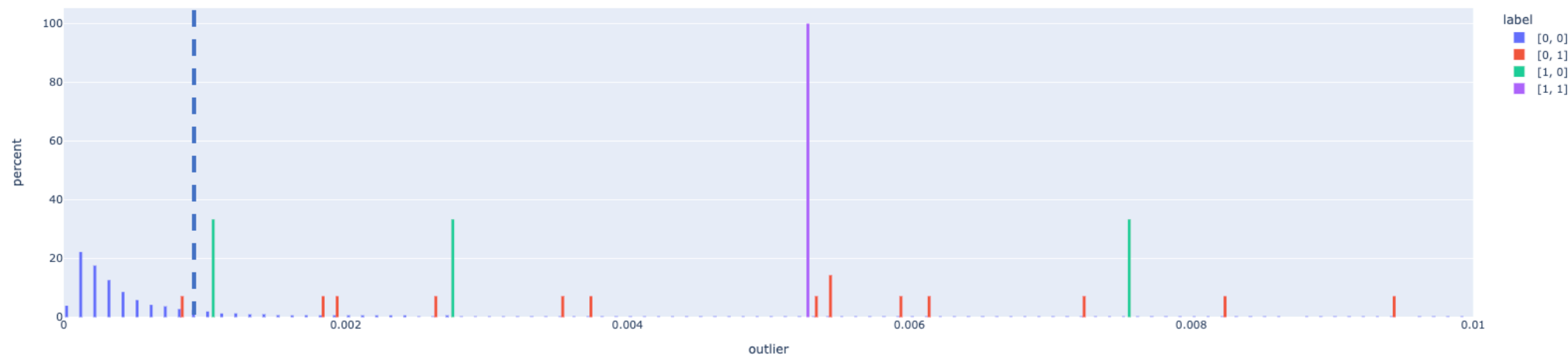
Method C
Auto encoder
representation

Method D
Outlier
detectors

Method E
Uncertainty
estimation

- We reproduce the previous graph but getting **outlier score** to investigate value of adding such score threshold after supervised modelling task.
- True normal (blue) have small outlier score whereas after a certain score there are almost only fraud (true and wrong) cases. This may be helpful to define a threshold.

Distribution of Outlier colored by label



4. Mixed techniques

Method A + D + E – Oversampling, Outliers & Uncertainty

Method A
Oversampling
with SMOTE

Method B
Synthesize fraud
with GAN

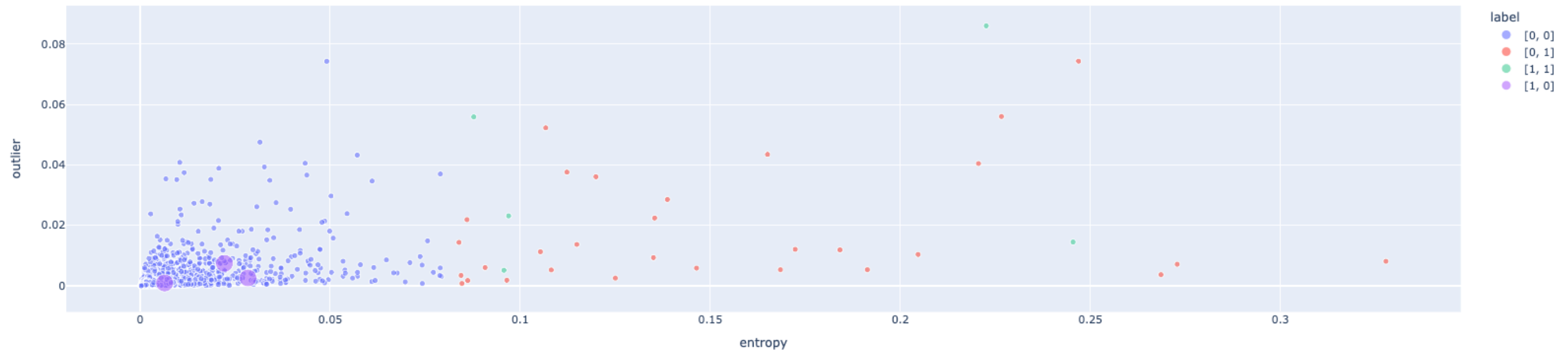


Method C
Auto encoder
representation

Method D
Outlier
detectors

Method E
Uncertainty
estimation

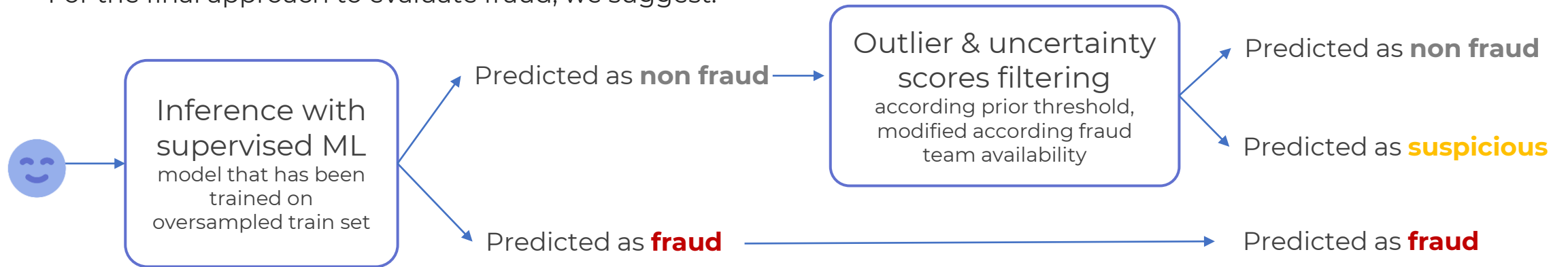
- Analysing the **co distribution of outlier score and uncertainty** according to model results allows to put in evidence areas of actions:
 - Blue area : we can really improve anything within it – main true normal
 - Green area: frontier zone where we could collect missing fraud
 - Red area : true fraud cases and or suspicious cases (false positive normal claims but not a lot so we don't care)



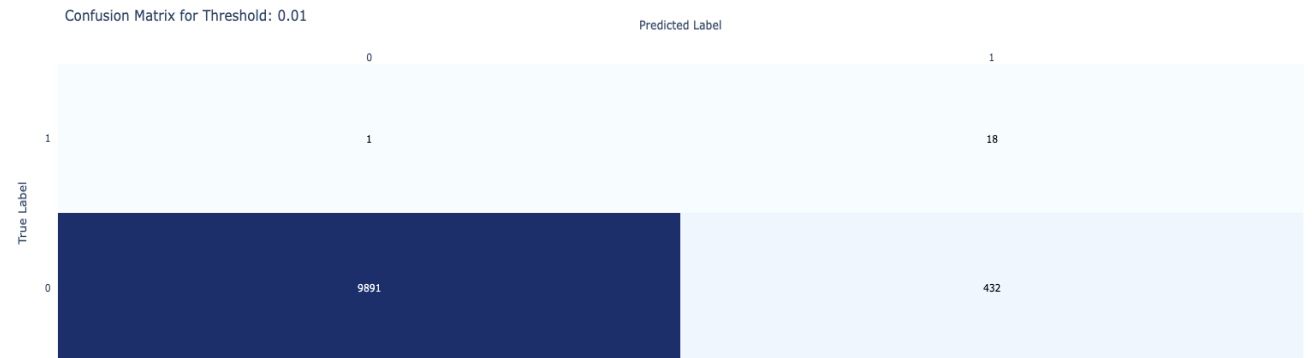
5. Conclusion Strategy

- For the final approach to evaluate fraud, we suggest:

```
outlier_tresh = 0.005
entropy_tresh = 0.018
```



- We improve the fraud recall significantly
- We provide a configurable thorough approach to share suspicious policies to fraud team



5. Conclusion


Industrialization

- In practice, Fraud analyst loads new data on Dataiku
- Outlier model and supervised model provide scores and entropy for thresholds (that can be edited)
- Inference on new data to output results, explore and prioritize cases

Fraud Inference

Adjust your fraud detection engine and produce list of potential frauds to be investigated

Upload Cases

 /y_test_smple_20231116.xlsx 125.54 KB

Drag and drop your files here or [ADD A FILE](#)

Creates a single dataset: multiple files must have the same schema.

[View settings of dataset cases](#) (excel, 2 column(s))

Edit Thresholds

Outlier Score Threshold

Entropy Threshold

Run Inference

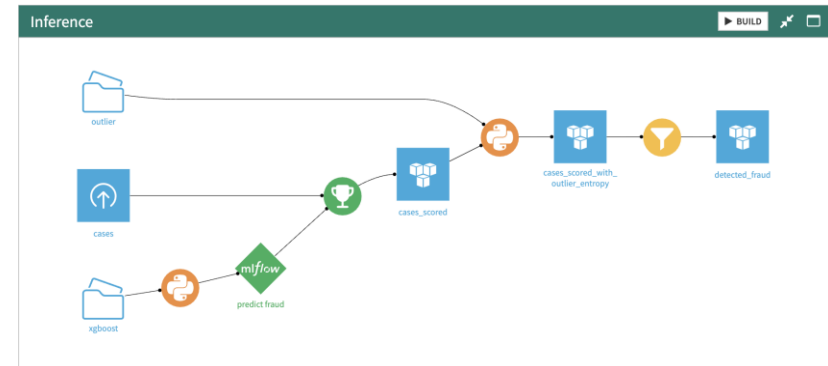
[RUN NOW](#) ✔ success Nov. 17 11:27:06 [RUN DETAILS](#)

Analyze Results

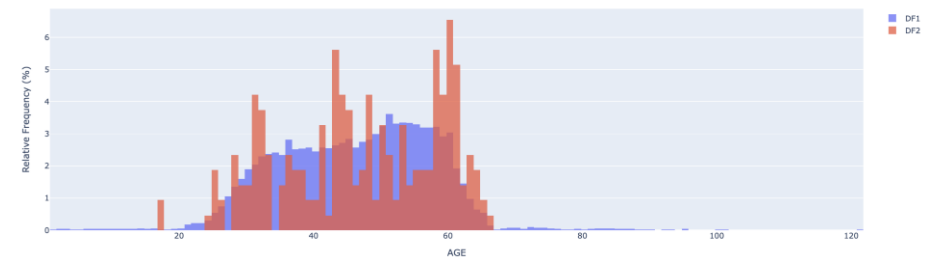
[Detected Fraud](#)

Download Cases to Investigate

[DOWNLOAD](#)



Relative Distribution Comparison of AGE



6. Appendix

- [1] Chawla, N. et al. "SMOTE: Synthetic Minority Over-sampling Technique." ArXiv abs/1106.1813 (2002): n. pag.
- [2] Maaten, Laurens van der and Geoffrey E. Hinton. "Visualizing Data using t-SNE." Journal of Machine Learning Research 9 (2008): 2579-2605.
- [3] Wang, He-Yong. "Combination approach of SMOTE and biased-SVM for imbalanced datasets." 2008 IEEE international joint conference on neural networks (IEEE World Congress on Computational Intelligence). IEEE, 2008.
- [4] McInnes, Leland and John Healy. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction." ArXiv abs/1802.03426 (2018): n. pag.
- [5] Little, Claire, Mark Elliot, Richard Allmendinger, and Sahel Shariati Samani. "Generative Adversarial Networks for Synthetic Data Generation: A Comparative Study." arXiv preprint arXiv:2112.01925 (2021). https://unece.org/sites/default/files/2021-12/SDC2021_Day2_Little_AD.pdf
- [6] Xu, Lei, and Kalyan Veeramachaneni. "Synthesizing tabular data using generative adversarial networks." arXiv preprint arXiv:1811.11264 (2018). <https://arxiv.org/abs/1811.11264>
- [7] Xu L, Skoularidou M, Cuesta-Infante A, Veeramachaneni K. Modeling tabular data using conditional gan. Advances in Neural Information Processing Systems. 2019. <https://arxiv.org/abs/1907.00503>
- [8] Zhao, Zilong, Aditya Kurnar, Robert Birke, and Lydia Y. Chen. "CTAB-GAN+: Enhancing Tabular Data Synthesis." arXiv preprint arXiv:2204.00401, 2022. <https://arxiv.org/abs/2204.00401>
- [9] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning internal representations by error propagation." , Parallel Distributed Processing. Vol 1: Foundations. MIT Press, Cambridge, MA, 1986.
- [10] Bank, Dor et al. "Autoencoders." ArXiv abs/2003.05991 (2020): n. pag.
- [11] Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest." 2008 eighth IEEE international conference on data mining. IEEE, 2008.
- [12] Reynolds, Douglas A. "Gaussian mixture models." Encyclopedia of biometrics 741.659-663 (2009).
- [13] Zhao, Yue, and Maciej K. Hryniewicki. "Xgbod: improving supervised outlier detection with unsupervised representation learning." 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018.
- [14] Gal, Uncertainty in Deep Learning, www.cs.ox.ac.uk/people/yarin.gal/website/thesis/thesis, 2016
- [15] N. G. Polson, V. Sokolov et al., "Deep learning: a Bayesian perspective," Bayesian Analysis, vol. 12, no. 4, pp. 1275–1304, 2017.