Confiance.ai Days 2022. Booklet of articles & posters

Patrice Aknin, Bertrand Braunschweig, Loic Cantat, Faïcel Chamroukhi, Georges Hebrail, Frédéric Jurie, Angelique Loesch, Juliette Mattioli, Guillaume Oller

OCTOBER 5-6 2022 – PARIS-SACLAY, FRANCE

Confiance.ai Days 2022
Booklet of articles & posters

General introduction

This booklet gathers 52 papers, either in the form of articles or posters, presented during the second edition of the Confiance.ai Days held in Saclay on October 4-6, 2022. Altogether they give a good snapshot of the research and development work done in the Confiance.ai program, an industrial and academic initiative of the national Grand Challenge on provable and certifiable AI, launched in support of the France 2030 strategy.

Among these papers, a dozen are presented as « external contributions » that were selected by an ad-hoc committee following a call for papers. All other communications belonged to one of five so-called « villages » with physical implementation in the conference hall, distributing the work done in Confiance.ai into five topics : « End-to-end approach » ; « from Operational Design Domain to Data » ; « Explainability and Understanding» ; « Robustness and Monitoring » ; « Embedded AI ».

After two years of activity, and complementing the Confiance.ai white paper, this document shows the diversity and the quality of the work done in the programme. Some important and up-to-date subjects are addressed, such as – only to name a few - out-of-distribution detection, adversarial robustness, semi- or self-supervised learning, explainability by design, verification and validation, embedded AI etc. We hope that you will enjoy reading parts of this document as much as we enjoyed preparing and attending the 2022 Confiance AI days.

Summary

External contributions

- Héléna Vorobieva. Design method for improving the detection of out of distribution data of type anomaly by multi-epoch ensemble method
- Thomas Cordier, Victor Bouvier, Gilles Hénaff, Céline Hudelot. Test-Time Adaptation with Principal Component Analysis
- Adrien Chan Hon Tong. Features which are robust to adversarial attacks are also robust to several poisoning attacks
- Timothée Fronteau, Arnaud Paran, Aymen Shabou. Evaluating Adversarial Robustness on Document Classification
- Etienne Bennequin, Myriam Tami, Antoine Toubhans, Céline Hudelot. Few-Shot Image Classification Benchmarks are Unrealistic: Build Back Better with Semantic Task Sampling
- Ramzi Ben Mhenni, Mohamed Ibn Khedher, Stéphane Canu. Robustness of Neural Networks Based on MIP Optimization
- Loris Berthelot, Andrés Troya-Galvis, Christophe Gouguenheim. A method and metrics to evaluate confidence score performances
- Arthur Ledaguenel, Céline Hudelot, Mostepha Khouadjia. Multi-Category Classification with Semantic Projection and Semantic Regularization
- Mehdi Elion, Sonia Tabti, Julien Budynek. Interpretability of deep learning models for visual defect detection: a preliminary study
- Fateh Boudardara, Abderraouf Boussif, Mohamed Ghazel, Pierre-Jean Meyer. Deep Neural Networks Abstraction using An Interval Weights Based Approach
- Tarek Ayed, Etienne Bennequin, Antoine Toubhans. Detecting Outliers in Few-Shot-Learning Support Sets
- Romain Xu-Darme, Georges Quénot, Zakaria Chihani, Marie-Christine Rousset. CASUAL: Case-based Reasoning using Unsupervised Part Learning

Village Bringing trust from ODD to Data (posters)

Introduction to the themes of the village
Flora Dellinger, Morayo Adedjouma

- Benoît Langlois, Jean-Luc Adam, Xavier Baril, Eric Feuilleaubois, Faouzi Adjed, Flora Dellinger. Towards Trustworthiness for Data Engineering in AI
- Adrien Le Coz, Stéphane Herbin, Faouzi Adjed. Expression and validation of an operational domain using extreme examples for computer vision applications
- Olivier Antoni, Marielle Malfante. Self-supervised Learning for Anomaly Detection on Time Series using 1D-CNN
- Laurence Guillon, Amélie Bosca, Michel Poujol. Anomaly Detection on Vibratory Sensors with Perceivers
- Evgenii Chzhen, Mohamed Hebiri, Jean-Michel Loubes, Gayane Taturyan. Robustness using fairness: problem formulation
- Fred Ngole Mboula. Sparsity based anomaly detection framework
- Fritz Poka Toukam, Nicolas Granger, Oriane Siméoni, Angélique Loesch. Leveraging unlabeled data to improve active learning for trustworthy data selection and annotation
- Christophe Bohn, Kévin Mantissa, Gabriel Burtin. Proposition of an ODD engineering process
- Georges Jamous, Morayo Adedjouma. ODD usages in a data and ML monitoring perspectives

Village End-to-end approach for trusted AI systems and V&V (posters)

Introduction to the themes of the village
Guillermo Chaley Gongora, Boris Robert, Cyprien de la Chapelle

- Boris Robert. Modeling for the description of use and architecture of Confiance.ai's Trustworthy Environment
- Boris Robert, Afef Awadid. Capturing and modeling the engineering processes for trustable AI based systems
- Boris Robert, Xavier Le Roux, Christophe Alix. End to end method for the engineering of trustable AI based systemssystems
- Christophe Alix, Guillermo Chale-Gongora, Jean-Luc Voirin. Engineering Trustworthy AI Systems End to End Visio
- Juliette Mattioli, Agnès Delaborde, Henri Sohier. Can we assess AI based system trustworthiness ?
- Morayo Adejouma, Christophe Alix, Loic Cantat, Eric Jenn, Juliette Mattioli, Boris Robert, Fabien Tschirhart, Jean-Luc Voirin. Engineering Dependable AI Systems
- Eric Jenn, Ramon Conejo, Vincent Mussot, Florent Chenevier. Assurance Cases and V&V Strategy
- Cyprien De La Chapelle, Ingrid Fiquet, Josquin Foulliaron. End to end use of trustworthy environment

Posters Village: Explainability tools and processes for understanding

Introduction to the themes of the village
Philippe Dejean

- Philippe Dejean, Thierry Allouche, Antoine Coppin, Caroline Gardet, Alice Petit, David Petiteau, Antonin Poché. Transversal studies around explainability
- Antonin Poché. Explainability: Methods and libraries
- Weituo Dai , David Cortés. Regional Explaination for ML Models
- Yannick Prudent, David Vigouroux. Counterfactuals-based metrics for the evaluation of image classifiers
- Elodie Guasch. Prototype-based models for explainability
- Baptiste Abeloos , Stéphane Herbin. Explaining objet detection : the case of Transformers architecture
- Maxime Desbois, Mathilde Guillemot, Antonin Poché. Explainable Unsupervised Anomaly Detection for Time Series
- Alice Petit, Antoine Coppin, Caroline Gardet. Explainability: State of the Art on explainability for NLP
- Romaric Besançon, Olivier Ferret, Vincent Feuillard, Caroline Gardet, Elies Gherbi, Lucas Schott, François-Paul Servant, Julien Tourille, Ayhan Uyanik. Methodology for Trustworthy Natural Language Process Models with Limited Training Data

Posters Village: Robust AI and monitoring

Introduction to the themes of the village
Hatem Hajri, Fateh Kaakai

- Corentin Friedrich, Thibaut Boissin, Franck Mamalet. Robustness by design with 1-Lipschitz networks
- Martin Gonzalez, Nelson Fernandez-Pinto. Robustification of NN by Diffusion Purification
- Pol Labarbarie, Stéphane Herbin, Adrien Chan-Hon-Tong, Milad Leyli-Abadi. Benchmarking and deeper analysis of adversarial patch attacks on object detectors
- Fateh Kaakai, Paul-Marie Raffi, Guillaume Bernard. MultiTimescale Monitoring of AI Models
- Kevin Pasini. Confidence indicators based on time series uncertainty decomposition for system monitoring
- Ramzi Ben Mhenni, Mohamed Ibn Khedher, Stéphane Canu. Robustness of Neural Networks Based on MIP Optimization

- Fabio Arnez, Ansgar Radermacher. Out-of-Distribution Detection using DNN Latent Space Uncertainty
- Héléna Vorobieva. Design method for improving the detection of out of distribution data of type anomaly by multi-epoch ensemble method

Posters Village: Trustworthy Embedded AI

Introduction to the themes of the village
Jacques Yelloz, Thomas Wouters

- Jacques Yelloz, Thomas Wouters. Trustworthy Embedded AI : scope and challenges
- Nassim Abderrahmane, Theo Allouche, Lionel Daniel, Frédéric Feresin, Omar Hlimi, Eric Jenn, Christophe Marabotto, Floris Thiant. Benchmarking of AI on Embedded Platforms
- Hugo Pompougnac, Dumitru Potop Butucaru, Albert Cohen, Floris Thiant. Embedded/Reactive Machine Learning Programming
- Marie-Charlotte Teulieres. Definition of a format for a safe embeddability of a ML Model
- Michaël Adalbert, Christine Rochange, Thomas Carle, Serge Tembo Mouafo, Eric Jenn, Makhlouf Hadji. Worst-Case Execution Time Analysis of Neural Networks on GPU accelerators
- Houssem Ouertatani, Cristian Maxim, El Ghazali Talbi, Smail Niar. Bayesian optimization with deep ensembles for AutoDL

External contributions

1. Héléna Vorobieva. Design method for improving the detection of out of distribution data of type anomaly by multi-epoch ensemble method
2. Thomas Cordier, Victor Bouvier, Gilles Hénaff, Céline Hudelot. Test-Time Adaptation with Principal Component Analysis
3. Adrien Chan Hon Tong. Features which are robust to adversarial attacks are also robust to several poisoning attacks
4. Timothée Fronteau, Arnaud Paran, Aymen Shabou. Evaluating Adversarial Robustness on Document Classification
5. Etienne Bennequin, Myriam Tami, Antoine Toubhans, Céline Hudelot. Few-Shot Image Classification Benchmarks are Unrealistic: Build Back Better with Semantic Task Sampling
6. Ramzi Ben Mhenni, Mohamed Ibn Khedher, Stéphane Canu. Robustness of Neural Networks Based on MIP Optimization
7. Loris Berthelot, Andrés Troya-Galvis, Christophe Gouguenheim. A method and metrics to evaluate confidence score performances
8. Arthur Ledaguenel, Céline Hudelot, Mostepha Khouadjia. Multi-Category Classification with Semantic Projection and Semantic Regularization
9. Mehdi Elion, Sonia Tabti, Julien Budynek. Interpretability of deep learning models for visual defect detection: a preliminary study
10. Fateh Boudardara, Abderraouf Boussif, Mohamed Ghazel, Pierre-Jean Meyer. Deep Neural Networks Abstraction using An Interval Weights Based Approach
11. Tarek Ayed, Etienne Bennequin, Antoine Toubhans. Detecting Outliers in Few-Shot-Learning Support Sets
12. Romain Xu-Darme, Georges Quénot, Zakaria Chihani, Marie-Christine Rousset. CASUAL: Case-based Reasoning using Unsupervised Part Learning

Design method for improving the detection of out of distribution data of type anomaly by multi-epoch ensemble method

Héléna Vorobieva *°

*Safran Tech, Digital Sciences & Technologies Department, Magny-Les-Hameaux, France, helena.vorobieva@safrangroup.com
°*IRT SystemX*, Palaiseau, France, helena.vorobieva@irt-systemx.fr

Ensemble methods with the training of a single neural network, but taken at different epochs are known to improve results in deep learning. In this work, we propose a new score to choose the best epochs, which is adapted to use cases of non-destructive testing of industrial parts where images have to be divided into patches before being proceeded by the network. This score is tested on the Safran use case of the Confiance.ai program.

I Introduction

In the context of non-destructive testing of industrial parts, one possible solution for automatic inspection, is to place the part on a support, to illumination it and to take good resolution photos. From these photos, it is then required to automatically determinate whether an anomaly is present and it approximate location, while making few false alarms. Either when the system returns an alert with an anomalous area, the part is discarded, or it is examined by hand, or the area is given for inspection to another system that is more expensive in terms of computational time or power.

For this study, we consider use cases where the images cannot be directly proceeded and have to be inspected as patches by the system. This can occur for example when the parts have a specular and textured surface that can vary or have a non-trivial curvature. In these cases, it is possible to take pictures with different illuminations of the same position of the part, and then select only the best illumination of each patch. Two strategies are then possible to find the anomalies: a classification strategy of the whole patch or a semantic segmentation strategy inside each patch. To train a neural network to perform these tasks, cost functions are used. They penalize a bad response: misclassified patch or pixel.

The automatic control of parts by neural network is usually done by using a single network [1]. In order to gain in robustness, the use of several networks via ensemble methods is an approach known to the deep learning community [2], [3]. The classification from the obtained networks is classically done by calculating the average of the predictions, by voting or by more advanced techniques [4]. The authors of [5] obtain improvement of robustness by using a set of networks with guaranteed good coverage of the parameter space. When several neural networks are used, they can have different architectures, or the same architecture but a different initialization. Another way to gain robustness and improve results is to use ensemble methods on a single neural network but taken at different convergence points (training epoch number), with the advantage of training only one neural network. Thus, [4] randomly selects four different epochs to apply ensemble methods. In a more relevant way, [6] chooses the best epochs according to the cost function used for the training of the neural network.

Various ensemble methods can then be used alone or in combination with each other, or in combination with other methods for improving the results. This work studies the case of ensemble methods with the training of a single neural network, but taken at different epochs.

II Problem statement

The general industrial problem is to determinate whether an anomaly is present in the whole image and it approximate location, while making few false alarms. The division of the images into patches does not change this problem.

The technical problem we study here is how to select the epochs for the ensemble methods with the training of a single neural network that answers the industrial problem. In [6] the authors choose the best epochs according to the cost function used for training the neural network. Classically, there are other indicators in addition to the cost function to measure the performance of detection or semantic segmentation, for example Mean Intersection over Union (mIoU), Accuracy or Recall. To our knowledge, these indicators are not used in the state of the art for the selection of epochs in ensemble methods.

However, these measures do not answer the industrial problem as they only favor a good response in relation to the raw ground truth (healthy or anomalous patch or pixel). Indeed, the industrial problem considers another level of precision: approximate location of anomalies in images, with few false alarms. Thus, the classic measurements will be very penalizing if, for example, only a part of the pixels of an anomalous area are well classified, whereas it answers correctly the industrial problem. Similarly, if a large area of anomalies is found on several patches, it would be sufficient to classify only some of these patches as anomalies, whereas the measurements will be penalizing for the other poorly classified patches in this area. Symmetrically for healthy areas, it is identical from the point of view of the industrial

problem to return an aggregation of misclassified healthy pixels or patches, whatever the size of the aggregation, whereas the measurements will be more penalizing for larger aggregations. Thus, the state-of-the-art measures are not suitable for selecting the best epochs for our problem.

We therefore propose in this study a new score for selecting the best epochs, adapted in particular to the Safran use case of the Confiance.ai program.

III Detailed Design

We describe in this section in detail our method for semantic segmentation networks and specify at the end the few modifications for a classification neural network. The proposed method works whatever the semantic segmentation neural network and whatever the associated semantic segmentation cost function.

1. Calibration

Figure 1 gives the overall view of the calibration process with the calculation of the proposed score. Training and validation images are subdivided into patches provided to the neural network. During the training, at the end of each epoch, the new performance measure proposed in this work and explained in the steps below is calculated on the validation set.

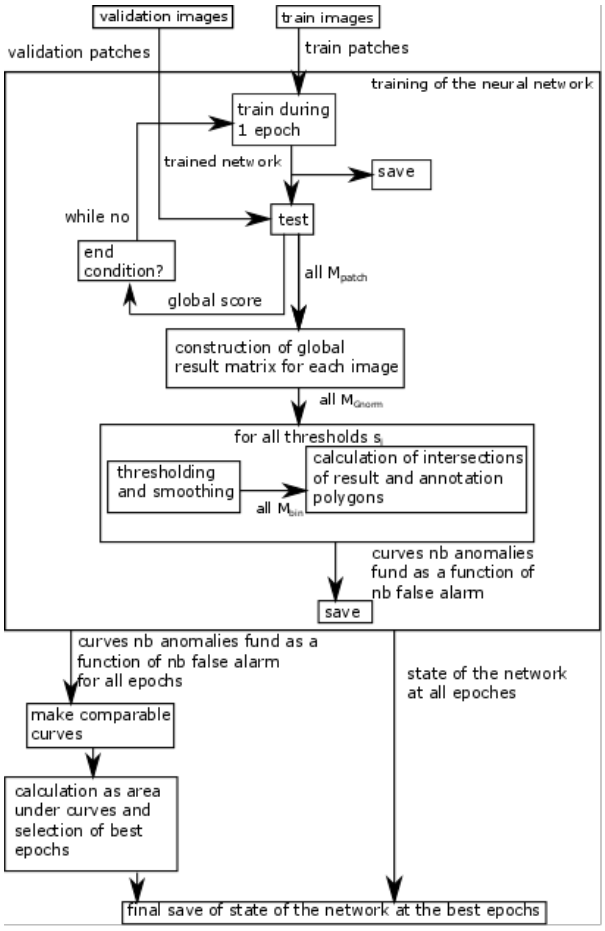


Figure 1 Schematic view of the calculation of the proposed score

Step 1: Construction of a global result matrix for each image of the validation set

For each image of the validation set, we test all the patches and we thus obtain as many small result matrices M_{patch} of the size of the patches, with scores between 0 and 1. As we know the position of the patches in the image, we construct a global result matrix M_G of the image size by placing the small matrices M_{patch} at the corresponding coordinates of the patches.

In the case of overlay of the patches, we merge the results. First, in case of overlaying the results are added in M_G. In parallel, we create a matrix $M_{contribution}$ containing for each coordinate, the number of small matrices having contributed to the score at this location. Then, when all the patches of the image have been tested and integrated in M_G and their contribution in $M_{contribution}$, we normalise M_G: $M_{Gnorm} = M_G/M_{contribution}$. At the end, we obtain an M_{Gnorm} for each image in the validation set.

2) Thresholding and classification of detected anomalies polygons

We fix different thresholds S regularly spaced between 0 and 1. The greater the number of thresholds, the more refined the results can be, but the longer the calculation time. The following is to be done for each threshold S_i.

Each M_{Gnorm} is thresholded and thus produces binary images M_{bin} where black pixels correspond to anomalies. Optionally, morphological smoothing can be performed.

We list all the black polygons (for example with connectedness 4) P_{result} in M_{bin} whatever their size. We then look at whether or not these polygons intersect annotation polygons P_{annot} :

- For each P_{annot}, we look if there is at least one P_{result} polygon having a non-null intersection with this P_{annot}. If this is the case, we consider that the P_{annot} anomaly is found.
- For each P_{result}, we look if there is at least one P_{annot} having a non-null intersection with this P_{result}. If this is not the case, we consider that the polygon P_{result} is a false alarm.

This operation being done for all the polygons P_{result} and P_{annot} of all the images of the validation set, we have a couple number of anomalies found and number of false alarms, for each threshold S_i.

3) Obtain comparable curves

At the end of the previous step, we can plot for each epoch the curve of the number of anomalies found as a function of the false alarms (each point of the curve corresponding to a different threshold S). We note these points $S_i(nbf\ alsealarm, nb_{anomalies\_found})$.

It is then necessary to check that these curves respect some rules. The closer the threshold is to 0, the greater the number of anomalies found must be and the more false

alarms we must see. Thus, for 2 given thresholds S_1 and S_2, if S_1 is smaller than S_2, then (assumption 1) the number of anomalies found for S_1 is greater than or equal to the number of anomalies found with S_2 and (assumption 2) the number of false alarms for S_1 is greater than or equal to the number of false alarms with S_2. Mathematically, (assumption 1) is always respected. However, for too low thresholds, (assumption 2) is no longer respected because instead of having many small false alarm areas, we end up with few very extensive false alarm areas. We thus find the list of the points S_h not respecting (hypothesis 2). For these points, the value of the number of anomalies found and the value of the number of false alarms must be modified. Let $S_N(nb_{falsealarmSN}, nb_{anomalies\_foundSN})$ be the first threshold from which (hypothesis 2) is respected for the considered period. Let F be the maximum number of false alarms over all epochs among the thresholds respecting (hypothesis 2). Then for the considered epoch, we modify the abscissa and ordinate of S_h such that: $S_h(nb_{falsealarm}, nb_{anomalies\_found}) = (F, nb_{anomalies\_foundSN})$. This is to be done for all epochs (thus for all curves). Thanks to this step, the maximum abscissa for all curves is the same. In order to make the minimum abscissa the same for all curves, for the curves where there is no point with abscissa 0, a point $S_0(0, 0)$ is added.

This gives curves with the same abscissa values, so that they are comparable. An illustration is given in Figure 2.

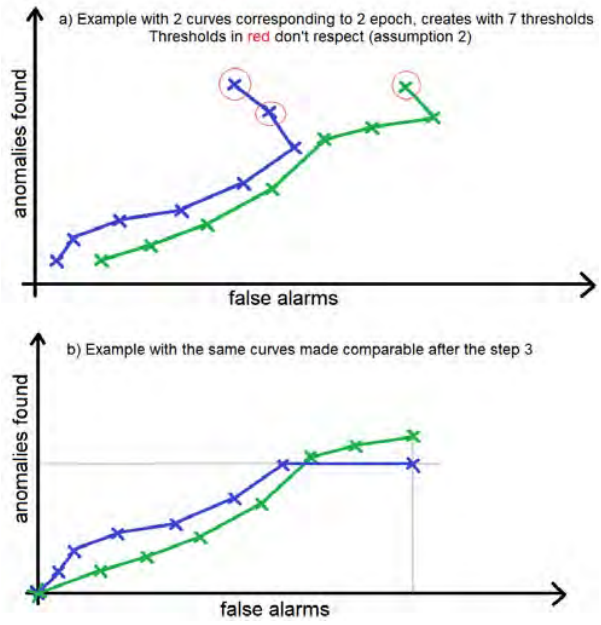


Figure 2 Example of how obtaining comparable curves

4) Final score

Let $N_{admitted}$ be the maximum number of false alarms that we accept to have on the whole validation base in a sub-optimal operating regime, for example $N_{admitted}$ can be equal to the number of images in the validation set. We then calculate the area under the curve for abscises between 0 and $N_{admitted}$, which gives us the final score. The higher the score, the better. Thus, for the ensemble methods, we use the epochs for which this score is the higher.

2. Test

The parts are tested only for the epochs selected during calibration and ensemble methods are then used to merge the results.

3. Modifications for a classification network

It is possible to apply this method with a classification neural network. In this case, instead of returning a matrix of scores for each pixel of the patch, the network returns a single score, determining whether the patch contains an anomaly or not. For the annotation of the patch, we also have a single score: classically 1 if the patch contains an anomaly and 0 otherwise. In this case, we modify step 1 by creating a matrix M_{patch} of the same size as the patch and by putting the score returned by the network at all the locations of the matrix. The rest is unchanged.

IV Experiment

1. Use case and used parameters

The method was experimented on the Safran use case, presenting 5-channels images of 2432x2050 pixels with delimited areas of interest and anomalies polygonal annotation. The images are divided into patches of 256x256 pixels with overlap. We used the training, validation and test sets provided with this use case. The network for which we tested our method is the resnet-18 classification network also provided with the use case.

The tuning elements during training are as follows:

- The training was done on 182 epochs using the 5 channels for each patch, directly masking the areas outside the masks on these patches.
- As suggested in step 4 for the calculation of the final score, we took $M_{admitted}$ equal to the number of images in the validation set (273 in our case).
- When searching for the polygons, we used a connection 4.
- We did not use any morphological operations

The consolidation during test was done by averaging the results over the N best epochs found during training.

2. Results

We present the results in the form of the industrial problem we have highlighted, i.e. giving the detection rate as a function of the number of false alarms (the same method as for steps 1 and 2 is used for this). In order to obtain result curves, we performed several thresholds, in a similar way to the thresholds in step 2. We thus present the results for the 5 best epochs and 5 thresholds on the final result in Figure 3 and the results for the 10 best epochs and 7 thresholds on the final result in Figure 4.

Note: we present the results according to the mean number of false alarms per image (1148 images on the test set).

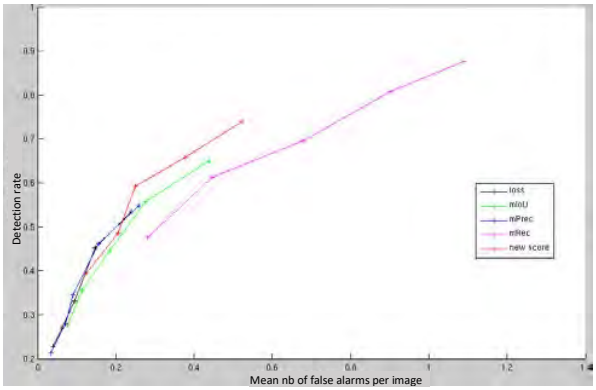


Figure 3 Detection rate function of mean number of false alarms per image, consolidation over 5 epochs

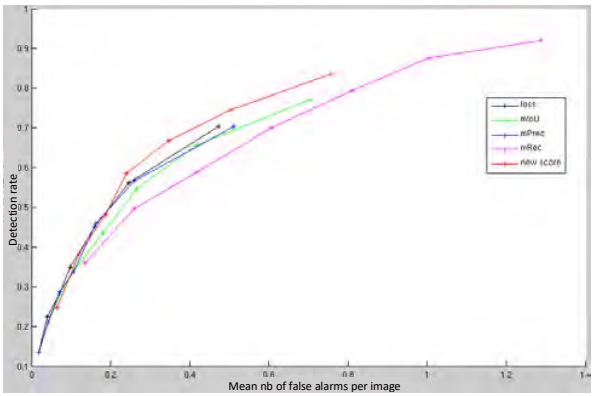


Figure 4 Detection rate function of mean number of false alarms per image, consolidation over 10 epochs

Our score (new score, in red) is compared to the scores considering the cost function (loss), the mean Intersection over Union (mIoU), the mean precision (mPrec) and the mean recall (mRec). We can notice that the score we propose is equivalent or better (depending on the points of the abscissa where we place ourselves) to the other scores, knowing that a perfect result would be a detection rate of 1 with no false alarm.

V Conclusions and future work

This work proposes a design method to improve the results of a neural network for anomaly detection. Without changing the learning strategy (same network, same cost function, same meta-parameters), the final decision is made from several epochs, chosen according to a new score. This score is particularly adapted to use cases with images divided into patches. This method can be used either alone or to monitor the initial neural network results (taken at the epoch of convergence of the cost function). In this case, a possible strategy could be to filter out discrepancies in the results of the two methods.

As a future work, this method will be tested on the Renault Welding use case of the Confiance.ai program. It will also be combined with other anomaly and out-of-domain detection networks.

Bibliography

[1] Morard, V. (2018). Procédé de contrôle non-destructif de pièce métallique. Patent

[2] Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In ICML

[3] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In NeurIPS

[4] Cheng, J., Aurélie, n. B., and Mark, v. d. L. (2018). The relative performance of ensemble methods with deep convolutional neural networks for image classification. Journal of Applied Statistics, 45(15):2800–2818

[5] Chapdelaine, C. and Picard, S. (2020). Procédé de contrôle d'une pièce mécanique par apprentissage en ligne de réseaux de neurones et dispositif associé. Patent

[6] Chen, H., Lundberg, S., and Lee, S.-I. (2017). Checkpoint ensembles: Ensemble methods from a single training process.

Test-Time Adaptation with Principal Component Analysis

Thomas Cordier [1 2] **Victor Bouvier** [3] **Gilles Hénaff** [1] **Céline Hudelot** [2]

Abstract

Machine Learning models are prone to fail when test data are different from training data, a situation often encountered in real applications known as distribution shift. While still valid, the training-time knowledge becomes less effective, requiring a test-time adaptation to maintain high performance. Following approaches that assume batch-norm layer and use their statistics for adaptation(Nado et al., 2020), we propose a Test-Time Adaptation with Principal Component Analysis (TTAwPCA), which presumes a fitted PCA and adapts at test time a spectral filter based on the singular values of the PCA for robustness to corruptions. TTAwPCA combines three components: the output of a given layer is decomposed using a Principal Component Analysis (PCA), filtered by a penalization of its singular values, and reconstructed with the PCA inverse transform. This generic enhancement adds fewer parameters than current methods (Mummadi et al., 2021; Sun et al., 2020; Wang et al., 2021). Experiments on CIFAR-10-C and CIFAR-100-C (Hendrycks & Dietterich, 2019) demonstrate the effectiveness and limits of our method using a unique filter of 2000 parameters.

1. Introduction

Deep neural networks are optimized to achieve high accuracy on their training distribution, given the hypothesis that they will be deployed on the same distribution during inference. However, distribution shift occurs in many industrial applications, for instance, when a sensor malfunctions. The accuracy of a predictive task drops as the distribution of test data shifts (Hendrycks & Dietterich, 2019; Quionero-Candela et al., 2009). Domain adaptation prevents such failures by jointly training on source and target data. Instead, Test-time adaptation mitigates the domain gap either by test-time training or fully test-time adaptation according to the availability of source data. Test-time training augments the training objective on source data with an unsupervised task that remains at test time to optimize domain-invariant representations. *Fully test-time adaptation* (Wang et al., 2021) does not alter training and only needs testing observations and a pre-trained model for privacy, applicability, or profit (Chidlovskii et al., 2016).

To enhance generalization, Spectral regularization (Bartlett et al., 2017) especially for GANs (Miyato et al., 2018) and L^2-regularization are standard tools during training (Neyshabur et al., 2017). L^2-regularization reduces model variance for different potential training sets and constrains the model complexity by lowering the weights of its layers. Spectral normalization penalizes the weight matrices by their largest singular value to ensure the Lipschitz continuity of the neural network.

Taking inspiration from these previous works, we aim to learn the best fitting parameters of a spectral filter on a corrupted dataset without supervision. We introduce TTAwPCA, which projects a batch of inputs onto a spectral basis, filters the projected data points, and reconstructs the filtered batch. As (Wang et al., 2021), we minimize entropy to learn the parameters of the filter. This generic unsupervised learning loss makes few assumptions about the data.

In this paper, we first overview state-of-the-art test-time adaptation (Sec. 2). Then, we introduce a simple yet effective method: TTAwPCA (Sec. 3). We demonstrate its effectiveness experimentally in tackling corrupted data (Sec. 4 and we discuss our results compared with other methods (Sec. 5).

2. Related work

Unsupervised Domain Adaptation jointly adapts on source and target domain through transduction, thus requiring both simultaneously. Several properties have been optimized: cross-domain feature alignment (Gretton et al., 2009; Baochen et al., 2017; Quionero-Candela et al., 2009), adversarial invariance (Tzeng et al., 2017; Ganin & Lempitsky, 2015; Ganin et al., 2016; Hoffman et al., 2018), and shared

[1]Thales Land and Air Systems, 2 Avenue Gay-Lussac, 78990 Elancourt, France [2]Université Paris-Saclay, CentraleSupélec, Mathématiques et Informatique pour Complexité et les Systèmes, 91190, Gif-sur-Yvette, France [3]Dataiku, 203 Rue de Bercy, 75012, Paris, France. Correspondence to: Thomas Cordier <thomas.cordier@centralesupelec.fr>.

proxy tasks (Sun et al., 2019) such as predicting rotation and position. In our work, we want to use only the target domain at test time.

Test-time adaptation indicates methods tackling the domain gap during inference. *TTT* (Sun et al., 2020) augments the supervised training objective with a self-supervised loss using source data. Only the self-supervised loss keeps adapting at test time on target domain. It relies on predicting the rotation of inputs, a visual proxy task, but designing suitable proxy tasks can be challenging. Training parameters are altered during training and test-time adaptation. *Test-time batch normalization* (Schneider et al., 2020; Nado et al., 2020) allows statistics of batch norm layers to be tracked during the distribution shift at test time. *TENT* (Wang et al., 2021) exhibits entropy minimization at test time on feature modulators extracted from spatial batch normalization to adapt to distribution shift. Entropy minimization is a generic and standard loss for domain adaptation to penalize classes overlap. Information maximization (Krause et al., 2010; Shi & Sha, 2012; Hu et al., 2017) used by (Liang et al., 2020; Mummadi et al., 2021) involves entropy minimization and diversity regularization. The diversity regularizer averts collapsed solutions of entropy minimization. *SLR+IT* (Mummadi et al., 2021) argues that Information maximization compensates for the vanishing gradient issues of entropy minimization for high confidence predictions. Moreover, an additional trainable network shares the input samples with the tested network to partially correct the domain shift. Principal Component Analysis cuts out noisy eigenvalues to remove uncorrelated noise (Li, 2018; Murali et al., 2012). In addition, we propose to add fully test-time learnable parameters to reduce the remaining noise of corrupted data onto the spectral basis.

3. Filtering the corrupted Singular Values

Let a neural network f_θ with parameters θ be trained to completion on a source set $X_\mathcal{D}$ of N samples from a distribution $\mathcal{D}$. Parameters θ are thus frozen after training. The initialization of our method takes place before testing. TTAwPCA is added after the jth layer. It consists of a Principal Component Analysis (PCA) and, for now, a pass-through filter. To fit its PCA, the concatenated output $A_{j,\mathcal{D}}$ of the jth layer has to be flattened from the shape N elements of the batch times c channels times the spatial dimensions $h \times w$ to a rectangular matrix of size $N \times p$ where $p = c \cdot h \cdot w$ and then mean normalized. Singular Value Decomposition breaks down the flattened training output $\tilde{A_\mathcal{D}}$ as:

$$A_{j,\mathcal{D}} = U \Lambda V^\top \tag{1}$$

where Λ is an $N \times p$ matrix of singular values, U an $N \times N$ matrix of left singular vectors and V an $p \times p$ matrix of right singular vectors. We define a hyperparameter L such that only the first L singular values are conserved. Note that this operation belongs to the training procedure.

At test time, the filter F_Γ is enabled to optimize its parameters $\Gamma = \{\gamma_i; i \in [0, L-1]\}$ of the corrupted singular values. Let the t-th batch of corrupted observation $x_t \sim D'$ be presented to the model $f_{\theta,\Gamma}$. Let $A_{j,\mathcal{D}',t}$ be the tth batched output of the jth layer. After the flatten operation and the mean normalization, $A_{j,\mathcal{D}',t}$ is projected onto the singular basis vectors by V_L, filtered by F_Γ and reconstructed by $V_L^\top$ as $O_{t,\mathcal{D}'}$ in its original basis:

$$O_{t,\mathcal{D}'} = A_{j,\mathcal{D}',t} V_L F_\Gamma V_L^\top \tag{2}$$

We designed a filter F_Γ related with L^2-regularization as demonstrated in A of diagonal element $F_{i,i}$ based on the singular values Λ_L of the training set and L learning parameters γ_i:

$$F_{i,i}(\gamma_i) = \frac{\lambda_i}{\lambda_{i,i} + \mathrm{ReLU}(\gamma_i)} \tag{3}$$

The ReLU activation assures the stability of the filter.

Similarly, we designed a negative exponential filter F_Γ of diagonal element $F_{i,i}$:

$$F_{i,i}(\gamma_i) = \frac{1}{1 + \exp(\gamma_i^2 - \lambda_i)} \tag{4}$$

We denote this model $f_{\theta,\Gamma}$ composed of f_θ and TTAwPCA. The learning parameters Γ are optimised over the batch x_t using entropy minimization of model prediction $\hat{y}_t = f_{\theta,\Gamma}(x_t)$ as test-time objective.

4. Experiments

Dataset. We classify CIFAR-10-C and CIFAR-100-C (Hendrycks & Dietterich, 2019). Both test sets contain 10,000 images of CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) augmented by 15 common corruptions and five severity levels.

Models. We use pre-trained WideResNets-28-10 (Zagoruyko & Komodakis, 2016). TTAwPCA is set after the first convolutional layer with only 2000 parameters for our best results on both datasets. We compare our two different filters with TENT (Wang et al., 2021) and test-time batch statistics updates (Schneider et al., 2020; Nado et al., 2020).

Settings. Episodic and online settings describe whether the model is reset after optimization on each batch or after optimization on the corruption at a given severity.

Optimization. We optimize the parameters Γ of the filter by Adam (Kingma & Ba, 2015) for one step on both offline and episodic fully test-time adaptation settings. We set the batch size at 200 samples and the learning rate at 0,001. $L = 2000$ proved to be sufficient for our method, as shown in B.1.

Table 1. Episodic corruption error benchmark on CIFAR-10-C and CIFAR-100-C with the highest severity [in %].

Dataset	Method	Mean	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
CIFAR-10-C	No Adaptation	43.53	72.33	65.71	72.92	46.94	54.32	34.3	42.02	25.07	41.30	26.01	9.30	46.69	26.59	58.45	30.30
	BN	20.44	28.08	26.12	36.27	12.82	35.28	14.17	**12.13**	17.28	17.39	15.26	8.39	12.63	23.76	19.66	27.30
	TENT	**19.96**	28.05	26.11	36.31	12.80	35.28	14.16	12.14	**17.27**	**17.36**	15.23	8.37	**12.59**	23.77	**19.61**	27.31
	exp-TTAwPCA (ours)	20.35	**25.5**	**23.55**	**33.77**	14.82	35.04	15.24	13.76	17.73	17.43	16.09	8.62	14.58	24.44	20.00	**24.68**
	ReLU-TTAwPCA (ours)	20.42	28.10	25.99	36.13	**12.72**	**34.93**	**14.00**	12.24	17.29	17.8	**15.07**	**8.26**	13.09	**23.47**	19.76	27.41
CIFAR-100-C	No Adaptation	85.54	93.84	93.60	96.63	91.49	92.79	86.51	88.69	70.91	82.30	84.74	47.26	96.30	85.02	89.50	83.49
	BN	36.61	47.21	46.72	55.59	**27.33**	47.75	**28.23**	26.65	32.74	33.63	32.92	**21.35**	29.64	37.79	33.99	47.56
	TENT	**34.56**	**42.91**	**41.94**	**49.76**	28.27	**44.55**	28.75	27.38	**30.99**	**31.59**	**30.72**	21.88	30.81	**35.42**	**31.27**	**42.09**
	exp-TTAwPCA (ours)	37.89	45.92	45.71	54.23	32.82	47.88	31.98	30.04	33.53	35.12	36.26	22.46	32.92	39.18	34.91	45.37
	ReLU-TTAwPCA (ours)	36.62	47.41	46.80	55.50	27.61	47.76	28.28	**26.54**	32.67	33.46	32.80	21.41	**29.55**	37.67	34.25	47.53

Table 2. Online corruption error benchmark on CIFAR-10-C and CIFAR-100-C with the highest severity [in %].

Dataset	Method	Mean	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
CIFAR-10-C	TENT	**18.57**	**25.09**	**22.76**	**32.71**	**12.01**	**31.88**	**13.25**	**11.12**	**15.9**	**16.32** $\pm$ 0.59	**13.82**	**8.21**	**11.66**	**22.02**	**17.29**	**24.5** $\pm$ 0.43
	exp-TTAwPCA (ours)	20.28	25.42	23.44	33.92	14.79	34.81	15.18	13.71	17.52	17.53 $\pm$ 0.17	16.09	8.62	14.58	24.44	20.00	**24.68** $\pm$ **0.12**
	ReLU-TTAwPCA (ours)	20.45	28.14	25.84	36.23	12.85	35.04	14.01	12.22	17.27	17.63	15.08	8.37	13.05	23.58	19.93	27.44
CIFAR-100-C	TENT	**31.7**	**38.74**	**36.88**	**44.00**	**26.91**	**41.03**	**27.33**	**25.54**	**28.18**	**28.85**	**28.03**	**20.44**	**28.81**	**33.93**	**28.41**	**38.41**
	exp-TTAwPCA (ours)	37.89	46.02	45.8	54.15	32.56	47.87	31.91	30.14	33.62	35.19	35.98	22.33	33.08	39.18	34.93	45.51
	ReLU-TTAwPCA (ours)	36.83	47.39	46.82	55.95	27.80	48.30	28.49	26.85	32.92	33.78$\pm$ 0.20	32.91	21.64	29.58	37.94	34.48	47.59$\pm$ 0.10

5. Discussion

TTAwPCA tackles common corruptions (Hendrycks & Dietterich, 2019) by improving the accuracy of each perturbed set. With only the 2000 parameters, TTAwPCA achieves state-of-the-art performance on various corruptions in the episodic CIFAR-10-C setting. Namely: Gaussian Noise, Shot Noise, Impulse Noise, Glass Blur, and JPEG compression for the exponential filter and Defocus Blur, Glass Blur, Motion Blur, Fog, Brightness, and Elastic Transformation for the ReLU filter whereas performing close to TENT (Wang et al., 2021) on the rest. Our method achieves a better trade-off between accuracy retrieval and the number of parameters. On the other hand, TTAwPCA does not take advantage of the online setting and does not scale well to CIFAR-100-C. We provide intuitions to explain this observation.

TTAwPCA enables PCA to filter noisy singular values on the remaining dimensions, assuming additive noises increase singular values. However, we observe some corruptions to reduce singular values effectively, thus filtering crucial information to the tested task. A penalizing filter is unable to recover this loss of information. Adding a multiplicative parameter to each diagonal element of our filter became a subject of our interest but was found unstable. To increase stability, we normalized each singular value λ_i by its higher value: λ_0. The instability of the tested filter prevents its convergence in an online setting.

Our results on CIFAR-100-C tend to be underperforming. High similarity between classes of CIFAR-100 might be too complex for TTAwPCA to reach over-parametrized methods such as TENT. A subtle change in the first principal components of the PCA can significantly affect the discriminability of the model if corruption occurs and the classes are too close. The first convolutional layer might not be discriminative enough to perform reliable principal components. On the other hand, the following layers merge the corruption and the features relevant to the task.

We argue that TTAwPCA follows the setting of *Fully test-time adaptation* (Wang et al., 2021) as TTAwPCA does not change the training objective. TTAwPCA expects a model to have a fitted PCA after completing the training procedure. Equivalently TENT needs spatial batch normalization layers to operate.

Lastly, TTAwPCA is the only method that does not alter any training parameter. Its test-time update can be fully deactivated without reloading the model instead of TENT or batch adaptation at test time (BN). The batch normalization parameters are forgotten through their processes. PCA also offers a linear adaptation of the model.

6. Conclusion

This paper introduced a new layer called TTAwPCA, filtering the singular values to tackle the out-of-distribution shift at test time. This spectral filter, initialized after training, is optimized on the test dataset with a task agnostic loss. We compared the effectiveness of our method in an online and an episodic setting to TENT (Wang et al., 2021) on CIFAR-10-C and CIFAR-100-C (Hendrycks & Dietterich, 2019). We argue our technique to adapt efficiently, reaching a new state-of-the-art on some corruptions without altering training parameters. We provided explanations of the success and the flaws of spectral penalization and its connections with standard methods in Machine Learning.

References

Baochen, S., Jiashi, F., and Kate, S. *Correlation Alignment for Unsupervised Domain Adaptation*, pp. 153–171. Springer, 2017.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/b22b257ad0519d4500539da3c8bcf4dd-Paper.pdf.

Chidlovskii, B., Clinchant, S., and Csurka, G. Domain adaptation in the absence of source domain data. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 451–460, 2016. doi: https://doi.org/10.1145/2939672.2939716.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1180–1189, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/ganin15.html.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. URL http://jmlr.org/papers/v17/15-239.html.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. *Covariate shift and local learning by distribution matching*, pp. 131–160. MIT Press, Cambridge, MA, USA, 2009.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., and Darrell, T. CyCADA: Cycle-consistent adversarial domain adaptation. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1989–1998. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/hoffman18a.html.

Hu, W., Miyato, T., Tokui, S., Matsumoto, E., and Sugiyama, M. Learning Discrete Representations via Information Maximizing Self-Augmented Training. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1558–1567. PMLR, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Krause, A., Perona, P., and Gomes, R. Discriminative clustering by regularized information maximization. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper/2010/file/42998cf32d552343bc8e460416382dca-Paper.pdf.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Li, B. A principal component analysis approach to noise removal for speech denoising. In *2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS)*, pp. 429–432, 2018. doi: 10.1109/ICVRIS.2018.00111.

Liang, J., Hu, D., and Feng, J. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pp. 6028–6039, July 13–18 2020.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Mummadi, C. K., Hutmacher, R., Rambach, K., Levinkov, E., Brox, T., and Metzen, J. H. Test-time adaptation to distribution shift by confidence maximization and input transformation, 2021. URL https://arxiv.org/pdf/2106.14999.pdf.

Murali, Y., Babu, M., Subramanyam, D., and Prasad, D. Pca based image denoising. *Signal & Image Processing*, 3, 04 2012. doi: 10.5121/sipij.2012.3218.

Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B., and Snoek, J. Evaluating prediction-time

batch normalization for robustness under covariate shift, 2020.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/10ce03a1ed01077e3e289f3e53c72813-Paper.pdf.

Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D. Dataset shift in machine learning. In *MIT Press*, 2009.

Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., and Bethge, M. Improving robustness against common corruptions by covariate shift adaptation. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 11539–11551. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/85690f81aadc1749175c187784afc9ee-Paper.pdf.

Shi, Y. and Sha, F. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In Langford, J. and Pineau, J. (eds.), *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, ICML '12, pp. 1079–1086, New York, NY, USA, July 2012. Omnipress. ISBN 978-1-4503-1285-1.

Sun, Y., Tzeng, E., Darrell, T., and Efros, A. A. Unsupervised domain adaptation through self-supervision, 2019.

Sun, Y., Wang, X., Zhuang, L., Miller, J., Hardt, M., and Efros, A. A. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *BMVC*, 2016.

A. Connection with L^2-Regularization

Let $X \in \mathbb{R}^{n \times d}$, where n is the number of samples and d is the number of features. We consider the simple case of linear regression where $Y = X\theta$ where θ is the parameter of the model. The optimal parameter are defined as follows:

$$\theta^\star := \arg\min_\theta ||Y - X\theta||^2 \tag{5}$$

and it is straightforward to observe the following closed form:

$$\theta^\star = \left(X^\top X\right)^{-1} X^\top Y \tag{6}$$

it is also straightforward to observe that:

$$\theta^\star_\gamma := \arg\min_\theta ||Y - X\theta||^2 + \gamma \cdot ||\theta||^2 \tag{7}$$

leads to the close form:

$$\theta^\star_\gamma = \left(X^\top X + \gamma I_d\right)^{-1} X^\top Y \tag{8}$$

In the following, we note $C = X^\top X$. C is has an orthogonal eigen decomposition (symmetric, positive and definite).

$$C = U^\top D U \tag{9}$$

where $U \in \mathbb{U}(d)$ which is the unitary group $U^\top U = I_d$. We note the basis change of X as follows:

$$\tilde{X} := XU^\top \tag{10}$$

By construction, $\tilde{X}$ has a diagonal covariance,

$$\tilde{X}^\top X = UX^\top XU^\top = UX^\top XU^\top = UCU^\top = D \tag{11}$$

Now, what happens when regressing from $\tilde{X}$ to obtain $\tilde{\theta}^\star$:

$$\tilde{\theta}^\star := D^{-1}\tilde{X}Y \tag{12}$$

Now,

$$\tilde{X}\tilde{\theta}^\star = \tilde{X}D^{-1}\tilde{X}^\top Y = Y \tag{13}$$

$$X\underbrace{U^\top D^{-1} U X^\top Y}_{=\theta} = \tilde{X}D^{-1}\tilde{X}^\top Y = Y \tag{14}$$

$$X\underbrace{U^\top (D+\gamma I_d)^{-1} U X^\top Y}_{=\theta} = \tilde{X}D^{-1}\tilde{X}^\top Y = Y \tag{15}$$

$$\theta^\star \tilde{X} = \theta^\star X \tag{16}$$

Let break the equation of $\theta^\star_\gamma$:

$$\begin{aligned}
\theta^\star_\gamma &= \left(X^\top X + \gamma I_d\right)^{-1} X^\top Y && (17)\\
&= \left(U^\top (D + \gamma I_d) U\right)^{-1} X^\top Y && (18)\\
&= U^\top (D+\gamma I_d)^{-1} U X^\top Y && (19)\\
&= U^\top \underbrace{D(D+\gamma I_d)^{-1}}_{F_\gamma} D^{-1} U X^\top Y && (20)\\
&= U^\top F_\gamma U U^\top D^{-1} U X^\top Y && (21)\\
&= U^\top F_\gamma U \theta^\star_0 && (22)
\end{aligned}$$

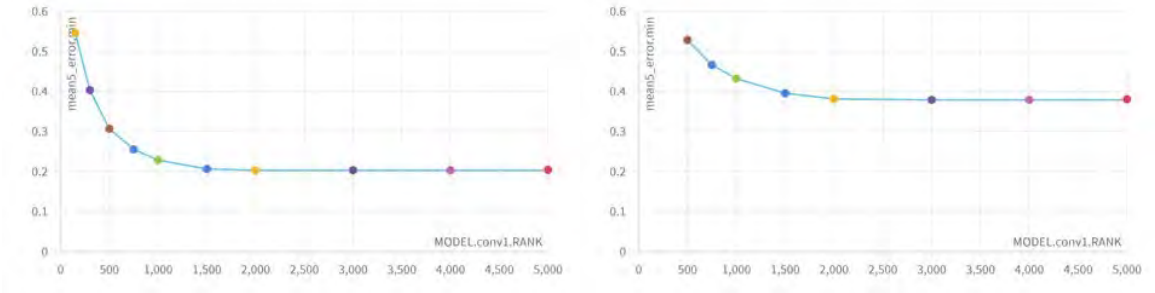
where F_γ is a diagonal matrix such that:

$$F_{\gamma,i,i} = \frac{\lambda_i}{\lambda_i + \gamma} \tag{23}$$

where λ_i is the $i-$th eigen-value of C.

B. Ablation Studies

B.1. PCA rank and parameters of the filter

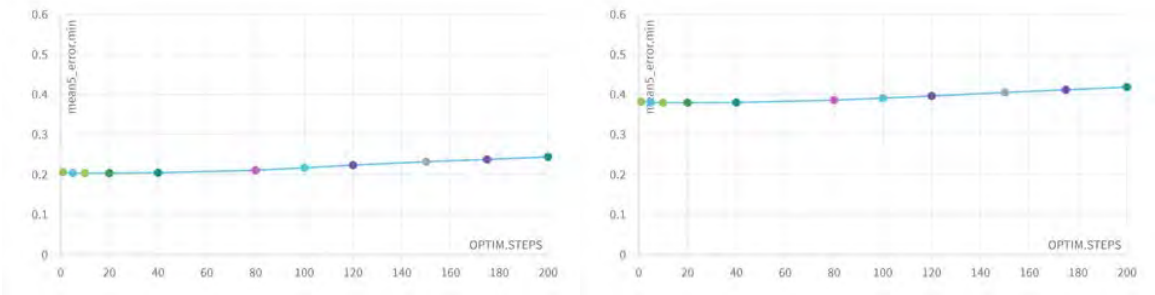


(a) CIFAR-10-C (b) CIFAR-100-C

Figure 1. Episodic mean error along all corruptions at severity 5 for different rank of the PCA of TTAwPCA.

Our experiments investigated how many parameters are enough to tackle corrupted data points. While these results only apply to CIFAR-10-C and CIFAR-100-C, we experienced that 2000 parameters are enough to effectively train a model to regain accuracy after a distributional shift at test time. In Figure 1, we show the mean error on all corruptions at severity 5 for different ranks of the PCA on both datasets. We averaged over three runs for each PCA rank with minor variations. The optimization has been done in an episodic setting.

B.2. Optimizing steps



(a) CIFAR-10-C (b) CIFAR-100-C

Figure 2. Online mean error along all corruptions at severity 5 for different number of learning steps of TTAwPCA.

As shown in (Mummadi et al., 2021), error degrades over optimization steps as entropy minimization lacks target distribution regularization. Still, this effect is minor compared with the accuracy retrieval achieved by our simple method.

C. Insight on CIFAR-10-C

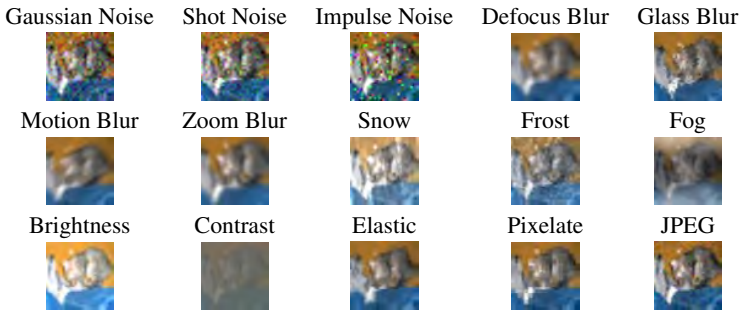


Figure 3. CIFAR-10-C (Hendrycks & Dietterich, 2019) consists of 15 corrupted versions of the CIFAR-10 test dataset (Krizhevsky, 2009) with 5 levels of severity (level 5 here).

D. Insight on Principal Component Analysis

Principal Component Analysis (PCA) linearly separates multivariate systemic variation from noise. Consider A an $N \times p$ data matrix. PCA defines its principal components as the $q \leq p$ unit vectors such that the i-th vector satisfy orthogonality with the first $i-1$ and best fits the direction of data. The process performs a change of basis on the data according to the principal components. They are computed by Singular Value Decomposition (SVD) of A and ranked by the corresponding singular value scale. Thus irrelevant principal components can be ignored.

Incremental PCA can be performed if the dataset is too large to fit in the memory. Incremental PCA uses an amount of memory independent of the number of input data samples to build a low-rank approximation.

Features which are robust to adversarial attacks are also robust to several poisoning attacks

Adrien CHAN-HON-TONG

June 10, 2022

Abstract

Most data poisoning methods target naive deep networks. Yet, it is well known that those networks exhibit strong sensitivity to perturbations.

Inversely, in this paper, I show that several data poisoning attacks (e.g. poison frog) are ineffective as soon as there are applied on features made robust to adversarial attacks, on both CIFAR and MNIST datasets.

1 Introduction

Deep learning (**DL**) which appears with [7] (see [9] for a review) is now at the core of most computer vision pipelines. Yet, many challenges have to be tackled before real life applications of deep learning for critical tasks: fairness, privacy, explainability...

One of these challenge, which has received a very strong attention from the community, is robustness. Indeed, it is known that naive deep learning is vulnerable under adversarial attacks [12, 20, 17, 19, 16, 4]: at test time, it is possible to design a specific invisible perturbation such as a targeted network eventually predicts different outputs on original and disturbed input. Worse, producing adversarial examples does not require to have access to the internal structure of the network [2, 14] and can have physical implementation [8].

Another issue is data poisoning [13] where an hacker modifies the training data to force the model to get a specific behavior.

Even more, the motivation of this paper is that both issues may be related. Indeed, both data poisoning and adversarial attacks are related with the idea of moving some data in a feature space (despite adversarial attack moves after decision boundary is selected while poisoning tries to change this boundary).

Yet, adversarial defenses had never been considered as potential way to mitigate some data poisoning attack in particular the ones based on small perturbation of the input image. The contribution of this paper is to prove that deep features trained with adversarial defense are more robust to those poisoning attacks than naive ones.

Precisely, three poisoning attacks are considered: PoisonFrog [18], adversarialpoisoning [1] and a labelflip attack (related to [13]). The evaluations will

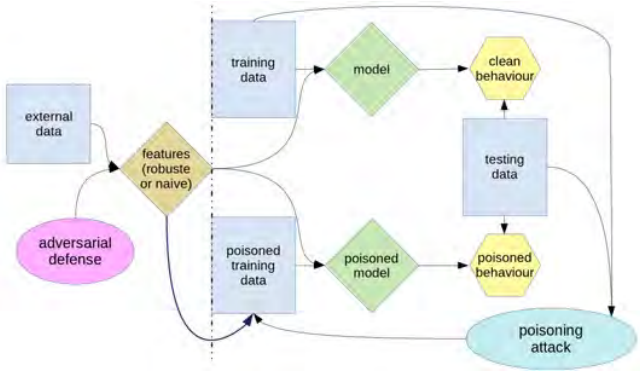


Figure 1: Illustrations of the framework to evaluate the impact of adversarial defense (and so feature robustness) on poisoning attacks.

rely on classical computer vision datasets CIFAR10 and CIFAR100 [6], MNIST [10], SVHN [15] with or without adversarial retraining (as adversarial defense). Consistently with [18, 1], deep features rather than full deep networks are considered. A consistent trend in all those experiments is that adversarial defense prevents poisoning attack based on small perturbations of the input image.

2 Adversarial defense against data poisoning

In order to evaluate the different poisoning attacks against the different features with more or less robustness, I rely on the following framework illustrated by figure 1. Importantly, the framework relies on frozen features following [18, 1]. Yet, both those papers have then been extended to poisoning against deep networks. Thus, the fact to rely on frozen features may not restrict too much the scope of the paper.

Classically, data poisoning is about comparing poisoned/clean behaviour related to poisoned/clean model where the poisoned model is trained on poisoned data (and clean model on clean data). This is the right part of the figure 1. Here, the objective is to see how those poisoning attacks behave as function of the features (brown lozenge in figure 1). Those features are produced from an external data (e.g. Imagenet [3] in [18, 1]). This is the left part of the figure 1 which corresponds to the classical training of a deep network with or without adversarial defense (e.g. FSGM from [5] or PGD from [11]).

So classically, data poisoning papers focus on the poisoning attack (the cyan ellipsoid in figure 1). Inversely, in this paper, the attacks are selected from state of the art. But, the contribution is to evaluate the impact of the adversarial defense (purple ellipsoid in figure 1).

Dataset	CIFAR	MNIST
AD on naive feature	24%	68%
AD on FSGM feature	30%	93%
AD on PGD feature	**34%**	**95%**

Table 1: Features robustness has positive impact on the accuracy under adversarial poisoning attack [1] on CIFAR and MNIST (here with VGG13).

3 Results

See https://github.com/achanhon/AdversarialModel for complete code and implementation detail description.

$\varepsilon = \frac{3}{255}$ on CIFAR (except if specified) like in [1], resulting in an invisible perturbation (for human eyes). But $\varepsilon = \frac{7}{255}$ on MNIST which is known to be less prone to adversarial sensibility (except if specified).

3.1 Adversarial poisoning [1]

The table 1 shows the accuracy after an AD attacks on VGG13 with features learnt with or without adversarial defense (for both MNIST with SVHN feature and CIFAR10 with CIFAR100 feature). This table shows that poisoning has less influence on PGD than FSGM, and, less influence on FSGM than on naive feature.

Currently, the clean performance of naive feature is higher than defended ones on CIFAR: accuracy of PGD features on clean CIFAR10 is only 41% i.e. poisoning has almost not effect but starting performances are much lower. However, it mainly show that transferring features from CIFAR100 to CIFAR10 is not a good idea. Inversely, adversarial defenses provide a very efficient protection with a better poisoned accuracy (despite a much lower clean accuracy).

On MNIST, the result is very interesting with a very high accuracy under poisoning with FSGM or PGD features: AD does not work at all on MNIST with PGD feature.

So, adversarial defenses are a data poisoning defense against [1] on MNIST (and mitigate the loss of accuracy related to [1] on CIFAR).

3.2 Poison frog [18]

The table 2 shows the ratio of points (over 100 trial) on which poison frog attack is successful on both naive, FSGM or PGD features on CIFAR with $\varepsilon = \frac{7}{255}$. The number of trial is slower than in [18]. However, it has to be stressed that each trial require to learn a SVM on the top of the features resulting in an expensive process (in particular with 3 different types of features).

Currently, on MNIST, PF works from Imagenet feature, but, not from SVHN features even with $\varepsilon = \frac{25}{255}$. So the MNIST results are not reported. Maybe,

feature	CIFAR
PF on naive feature	85%
PF on FSGM feature	53%
PF on PGD feature	**16%**

Table 2: Critical impact of features robustness on ratio of successful poison frog attacks on CIFAR.

Dataset	MNIST	CIFAR
LF on naive feature	-2%	-7%
LF on PGD feature	-2%	-11%

Table 3: Robust features do not suffer more than naive ones under label flip attack (here 2% of label is random) with VGG13.

the SVHN features are very robust on MNIST (even naive ones) making PF completely ineffective.

Again, this experiment shows that adversarial defense strongly decreases the impact of a data poisoning attack (PF here). Currently, PF is still active even with PGD feature on CIFAR (16% is still an issue) but much less than when targeting naive features (with 85% of successful attacks in this last case).

3.3 Label flip [13]

Both previous subsections shows that adversarial defense improves robustness to two poisoning attacks. Yet, those poisoning attacks are image-based.

Thus, it could be interesting to check if this results holds for label based poisoning attacks. Indeed, as robust features tend to increase distance between point in feature space, it could be even more sensible to label based attack. Currently, [13] combines both label and image perturbation. Yet, image perturbations are not bounded in [13] (see figure 5 and 6 of [13]), so there is no sense to consider norm bounded adversarial defenses against [13]. This is why, I focus on simple LF attack.

The table 3 shows the difference of accuracy (between clean model and poisoned one) after an LF attacks (2% of random label) on VGG13 with features learnt on with or without adversarial defense. The result is that adversarial defense does not increase the sensibility to label noise (accuracy gap is only slightly larger with robust features).

3.4 Conclusion

The main contribution of this paper is to prove that both poison frog and adversarial poisoning lose most of their effectiveness when targeting robust deep features (produced using adversarial defense).

References

[1] Adrien CHAN-HON-TONG. An algorithm for generating invisible data poisoning using adversarial noise that breaks image classification deep learning. *Machine Learning and Knowledge Extraction*, 1(1):192–204, Nov 2018.

[2] Moustapha M Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured visual and speech recognition models with adversarial examples. In *Advances in Neural Information Processing Systems*, pages 6977–6987, 2017.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[4] Chris Finlay, Aram-Alexandre Pooladian, and Adam Oberman. The log-barrier adversarial attack: Making effective use of decision boundary information. In *The IEEE International Conference on Computer Vision*, October 2019.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[6] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[8] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations (ICLR)*, 2017.

[9] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[10] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

[11] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[12] Seyed Mohsen Moosavi Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[13] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017.

[14] N. Narodytska and S. Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017.

[15] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[16] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

[17] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016.

[18] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *arXiv preprint arXiv:1804.00792*, 2018.

[19] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *technical report arxiv:1312.6199*, 2013.

[20] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Evaluating Adversarial Robustness on Document Classification

Timothée Fronteau Crédit Agricole S.A.
Arnaud Paran Crédit Agricole S.A.
Aymen Shabou Crédit Agricole S.A.

{timothee.fronteau, arnaud.paran, aymen.shabou}@credit-agricole-sa.fr

Abstract

Adversarial attacks and defense have gained increasing interest on computer vision systems in recent years, but as of today most investigations are limited to images. However, many artificial intelligence models actually handle documentary data which is very different from real world images. Hence, in this work, we try to apply the adversarial attack philosophy on documentary and natural data and to protect models against such attacks. To the best of our knowledge, no such work has been conducted by the community in order to study the impact of these attacks on the image document classification task.

As documents contain visual data in combination with text data, we want to explore visual attacks as well as text and multi-modal ones.

1 Introduction

Adversarial attacks targeted towards AI systems may be a new source of security vulnerabilities, as those systems are extensively used in industry for intelligent document processing processes. In the banking sector, humanly imperceptible adulterations of a document can create hard to detect frauds if the forgery results in misclassification of said document during a credential check. As part of any company's efforts to meet the requirements of the future European AI Act, we are evaluating robustness of visual and multi-modal approaches to document classification against state-of-the-art adversarial attacks.

In our study, we use state-of-the-art models to generate simple and efficient adversarial attacks, then we adapt those attacks to the documentary use case which mostly uses gray-scale images and finally we challenge our adversarial examples by using JPEG compression which considerably improves the robustness of the classification systems.

2 Data and Models

The *RVL-CDIP* dataset (Harley et al., 2015) is commonly used for research in document classification. Therefore we chose to use this dataset for the experiments we present in this paper. It is an open dataset containing 400,000 black-and-white document images split into 16 categories.

We chose to use a simple convolutional neural network (CNN) for our first robustness evaluations. In fact, state-of-the-art CNNs are easy to implement, light-weight and perform quite well, so they are commonly used for industrial purposes. Furthermore, a wide variety of adversarial attacks are designed to perform against image models like CNNs (Chakraborty et al., 2021). Following these motivations, we have implemented the visual model that currently performs best on RVL-CDIP[1], described in the article of Ferrando et al. (2020). It is an *EfficientNetB0* pre-trained on *ImageNet* that we fine-tuned on RVL-CDIP.

The evaluation method presented below was also used with an inhouse database to evaluate the robustness of a model similar to EfficientNetB0, using only the visual modality, and currently being used in production. For confidentiality reasons, the results of this study are not presented here.

[1] Meta AI. *Document Image Classification on RVL-CDIP*. June 2020. URL: https://paperswithcode.com/sota/document-image-classification-on-rvl-cdip

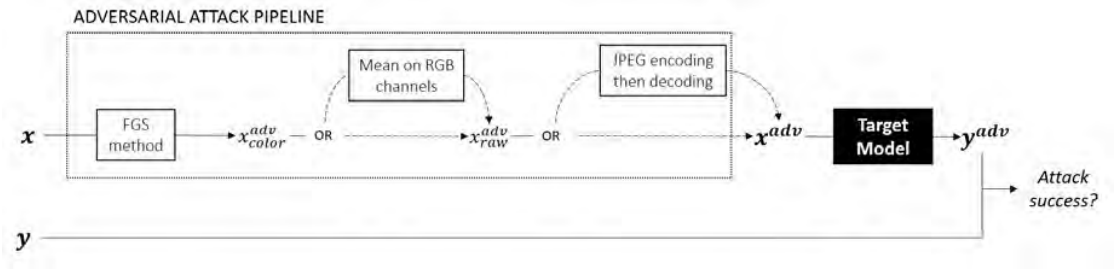


Figure 1: The robustness evaluation pipeline of a target model

3 Experiments

3.1 Threat Model

In order to have correct metrics, and to assess the validity of our approach in a real-world context, it is essential that we define a threat model using a precise taxonomy. We define this threat model according to Carlini et al. (2019) by defining the goals, capabilities and knowledge of the target AI system that the attacker has.

Let $C(x) : X \to Y$ be a classifier, and $x \in X \subset \mathbb{R}^d$ a document image entry. Let y be the ground truth of the entry x. In our first experiments, we perform *untargeted attacks*: the adversary's goal is to generate an adversarial example x^{adv} for x that is misclassified by C. x^{adv} should also be as *optimal* as possible, which means that it fools the model with high confidence in the wrong predicted label, with an input x^{adv} that is indistinguishable from x (Machado et al., 2021). Formally, we define the *capability* of the adversary by defining a *perturbation budget* ε so that $\|x - x^{adv}\| < \varepsilon$ where $\|\cdot\|$ is the L_∞ norm. An other factor of invisibility of the perturbation $\delta = x^{adv} - x$ specific to gray-scale document images is whether the perturbation is in gray scale or in color. In these early works, we implement *gradient-based white-box attacks*, which means that the adversary *knows* the architecture and parameters of the model. We intend to extend our research to transfer-based and decision-based attacks.

3.2 Attacks and defenses

We focus our first experiments on the robustness against an attack using variations of the Fast-Gradient Sign (FGS) method (Goodfellow et al., 2014). This easy to implement method is very efficient, enables transfer-based attacks (a threat configuration in which the attacker does not know the exact parameters of the model but knows the kind of architecture it uses e.g. a CNN) and is the basis of many other complex attacks that require less adversary knowledge. In the FGS method, an untargeted adversarial example is computed as $x^{adv} = x + \epsilon \cdot \text{sign}(\Delta_x J(x, y))$, where J is the cross-entropy loss.

Depending on the dataset and the model, we may want to generate perturbations that are gray-scale. To do so, we compute the mean value of each RGB pixel of the perturbation δ, which gives us a gray-scale perturbation. We call it the grayFGS method, in opposition to the FGS method where this post-processing step is not executed.

For industrial applications, the processed documents are available in a specific format, such as JPEG. This format has been identified by Dziugaite et al. (2016) as a factor of robustness against adversarial attacks. This is why we have designed a second post-processing step, which consists in converting the adversarial examples to JPEG format, then decoding them again before feeding them to the model. We call this step the JPEG step.

The two post-processing steps combined with the FGS method provide us with four attack methods that we use to evaluate robustness. Figure 1 summarizes our evaluation pipeline. Samples of perturbations and adversarial examples generated with the FGS and grayFGS methods are saved in JPEG format and are rendered in Figure 2.

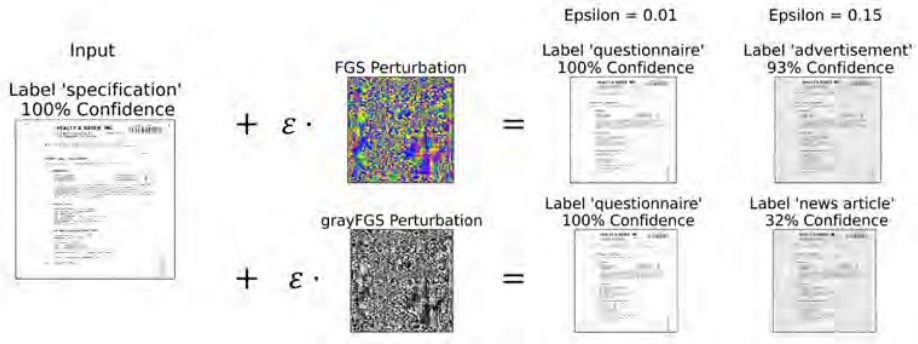


Figure 2: A base document with the perturbations and the adversary image

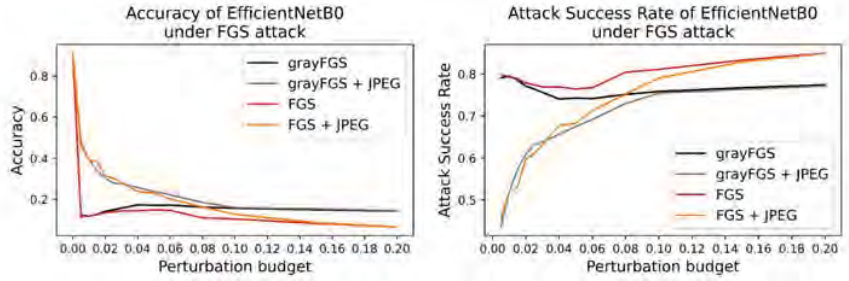


Figure 3: Accuracy and Attack Success Rate depending on the perturbation budget

4 Results

Similar to Dong et al. (2020), we selected two distinct measures to assess the robustness of the EfficientNetB0 model under attack, defined as follows. Given an attack method A that generates an adversarial example $x^{adv} = A(x)$ for an input x, the accuracy of a classifier C is defined as $Acc(C, A) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(C(A(x_i)) = y_i)$, where $\{x_i, y_i\}_{i=1}^{N}$ is the test set and $\mathbf{1}(\cdot)$ is the indicator function. The attack success rate of an untargeted attack on the classifier is defined as $Asr(C, A) = \frac{1}{M}\sum_{i=1}^{N} \mathbf{1}(C(x_i) = y_i \wedge C(A(x_i)) \neq y_i)$, where $M = \Sigma_{i=1}^{N}\mathbf{1}(C(x_i) = y_i)$. We performed the attacks for perturbation budgets within the range of 0.5% to 20% and evaluated the model accuracy under no attack. The evolution of model accuracy and attack success rate are presented in Figure 3.

The test accuracy of our EfficientNetB0 model on RVL-CDIP is 91.2%. The accuracy under attack is rendered in Table 1. We observe that adversarial examples we computed with the grayFGS method stay optimal with an adversarial budget of up to roughly 2%. Under such perturbation budget, the accuracy of the model drops to 14.1%, while with the JPEG post-processing step, the model accuracy is twice as good (30.3%) but still low. For the smallest perturbation budget of 0.5%, the accuracy is respectively 12.2% and 47.7% for the gray-scale attack without and with JPEG post-processing step.

perturbation budget		0.5%	2%	6%	20%
Accuracy	grayFGS	0,122	0,141	0,17	0,142
	grayFGS + JPEG	0,477	0,303	0,221	0,144
	FGS	**0,114**	**0,134**	**0,145**	**0,064**
	FGS + JPEG	0,458	0,315	0,202	0,065

Table 1: Accuracy of EfficientNetB0 under attack for several perturbation budgets

5 Conclusion and Future Works

As expected, a convolutional model such as our EfficientNetB0, trained without any strategy to improve its robustness, is very sensitive to optimal adversary examples generated with the new grayFGS method. Compressing and then decompressing the adversarial examples that will be provided as input to the model using JPEG protocol improves its robustness.

There are many ways to improve the robustness of a model that would only use the visual modality of a document (Chakraborty et al., 2021) (Machado et al., 2021). However, state-of-the-art approaches to document classification take advantage of other information modalities, such as the layout of the document, and the text it contains [1]. Therefore, after exploiting transfer-based and decision-based attack methods to evaluate the robustness of our visual models, we will evaluate the transferability of the generated examples to a multimodal model such as DocFormer (Appalaraju et al., 2021), which uses optical character recognition (OCR) and transformer layers. On the other hand, we will explore the possibility of designing adversarial attacks to which these models are more sensitive, for example by targeting OCR prediction errors (Song and Shmatikov, 2018) that affect the textual modality and may also affect the robustness of such models (Zhang et al., 2020).

References

[Appalaraju et al.2021] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 993–1003.

[Carlini et al.2019] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. 2019. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*.

[Chakraborty et al.2021] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2021. A survey on adversarial attacks and defences. *CAAI Transactions on Intelligence Technology*, 6(1):25–45.

[Dong et al.2020] Yinpeng Dong, Qi-An Fu, Xiao Yang, Tianyu Pang, Hang Su, Zihao Xiao, and Jun Zhu. 2020. Benchmarking adversarial robustness on image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 321–331.

[Dziugaite et al.2016] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. 2016. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*.

[Ferrando et al.2020] Javier Ferrando, Juan Luis Domínguez, Jordi Torres, Raúl García, David García, Daniel Garrido, Jordi Cortada, and Mateo Valero. 2020. Improving accuracy and speeding up document image classification through parallel systems. In *International Conference on Computational Science*, pages 387–400. Springer.

[Goodfellow et al.2014] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

[Harley et al.2015] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 991–995. IEEE.

[Machado et al.2021] Gabriel Resende Machado, Eugênio Silva, and Ronaldo Ribeiro Goldschmidt. 2021. Adversarial machine learning in image classification: A survey toward the defender's perspective. *ACM Computing Surveys (CSUR)*, 55(1):1–38.

[Song and Shmatikov2018] Congzheng Song and Vitaly Shmatikov. 2018. Fooling ocr systems with adversarial text images. *arXiv preprint arXiv:1802.05385*.

[Zhang et al.2020] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. 2020. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41.

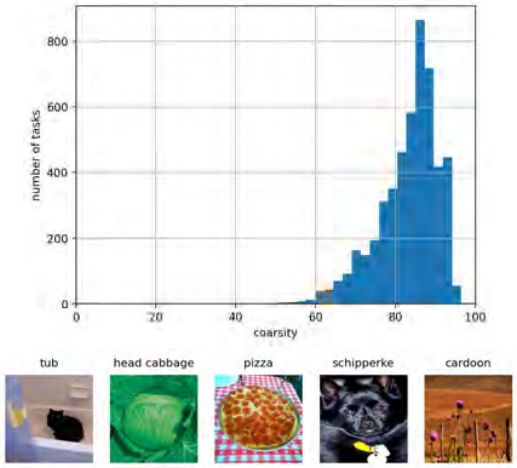
Few-Shot Image Classification Benchmarks are Unrealistic: Build Back Better with Semantic Task Sampling

1. Authors

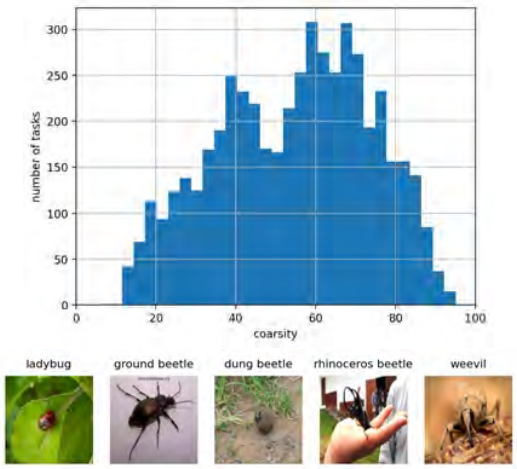
- **Etienne Bennequin**, PhD Candidate at Sicara and CentraleSupélec
  etienneb@sicara.com
- **Myriam Tami**, Associate Professor at CentraleSupélec
  myriam.tami@centralesupelec.fr
- **Antoine Toubhans**, Head of Science at Sicara
  antoinet@sicara.com
- **Céline Hudelot**, Professor at CentraleSupélec
  celine.hudelot@centralesupelec.fr

2. Summary

Every day, a new method is published to tackle Few-Shot Image Classification, showing better and better performances on academic benchmarks. Nevertheless, we observe that these current benchmarks do not accurately represent the real industrial use cases that we encountered. We show that widely used benchmarks are strongly biased towards tasks of differentiating between classes that would never be observed in the same context. We propose a novel task generation method to alleviate this bias, thus bridging the gap between academic Few-Shot Learning Research and real-life applications.



(a) Coarsity histogram (top) and an example of task (bottom) of a testbed designed from *tiered*ImageNet with uniform class sampling. This task presents a coarsity of 85.1, which is the median coarsity for this testbed. "We really need a machine to distinguish bathroom tubs from cabbage, pizzas, cardoon, and some very specific kind of dog!" said no one in the history of humankind.



(b) Coarsity histogram (top) and an example of task (bottom) of our testbed *better-tiered*ImageNet. This task presents a coarsity of 15.8. Tasks with this coarsity never occur in the uniformly sampled testbed, although they are more representative of real few-shot classification use cases.

Figure 1. Comparison, in terms of coarsity (see Equation 2), between a testbed designed with uniform class sampling (left) and a testbed designed with semantic awareness (right, ours). Our testbed gives a better representativity to tasks with low coarsity *i.e.* composed of classes semantically relevant to one another.

3. Motivations

Few-Shot Learning is the science of learning new concepts with only a few examples. This task is one of the key abilities of humans but one of the major shortcomings of standard deep learning methods [4]. It is also required for the success of many industrial applications of computer vision. Businesses that need automatic image recognition do not necessarily possess hundreds of labeled images for each class. It can be because some (or all) classes are rare, or appeared recently, or because classes change every day. In the last three years, our team was involved in a variety of industrial use cases: retrieving a tool in a merchant's catalog, identifying a floor to facilitate recycling, recognizing dishes in a cafeteria given today's menu, or finding the reference of a printed circuit board. In all of these use cases, some or all classes were represented in our database by only 1 to 5 examples. They also had in common that they consisted in recognizing an object among many classes that were semantically similar to one another.

Sadly, because of this, we could not rely on academic benchmarks to identify the most appropriate methods. The first reason is that, as we show in our work, standard Few-Shot Image Classification benchmarks generate tasks using uniform random sampling from a wide range of semantically dissimilar or unrelated classes, which leads to evaluating our models mostly on tasks composed of objects that we would never need to distinguish in real-life use cases (see Figure 1a). The second reason is that the Few-Shot Learning community has chosen to formalize the Few-Shot Image classification problem as an accumulation of n-way k-shot classification tasks *i.e.* classifying query images, assuming that they belong to one of n classes for which we have k labeled examples each. In practice, most works compared their methods on benchmarks for which they fixed $n = 5$ (sometimes $n = 10$) and $k = 1$ or $k = 5$[1]. To the best of our knowledge, only one method was evaluated with $n > 50$ [7]. The choice made by the community, while relevant to facilitate experiments in the early stages of Few-Shot Learning research, casts a dark shadow on the robustness of state-of-the-art few-shot learning methods when discriminating between a large number of classes.

4. Contributions

How can we improve our current evaluation processes to better fit real-life use cases? In this work, we bring out the limitations of current Few-Shot Classification benchmarks with both quantitative and qualitative studies and propose new benchmarks to get past these limitations. More specifically:

1. We use the WordNet taxonomy [6] to evaluate *semantic distances* between classes of the popular Few-Shot Classification benchmark *tiered*ImageNet. Based on these semantic distances we put forward the concept of *coarsity* of an image classification task, which quantifies how semantically close are the classes of the task.

2. We conduct both quantitative and qualitative studies of the tasks generated from the test set of *tiered*ImageNet *i.e.* the tasks composing the benchmark on which most papers evaluate different methods. We show that this benchmark is heavily biased towards tasks composed of semantically unrelated classes.

3. We harness the semantic distances between classes to generate the improved benchmark *better-tiered*ImageNet reestablishing the balance between fine-grained and coarse tasks. We compare state-of-the-art Few-Shot Classification methods on this new benchmark and bring out the relation between the *coarsity* of a task and its difficulty.

These contributions are part of our paper *Few-Shot Image Classification Benchmarks are Too Far From Reality: Build Back Better with Semantic Task Sampling* [1], presented at CVPR 2022 in the 1st Workshop on Vision Datasets Understanding. All our implementations, datasets and experiments are publicly available[2].

4.1. Measure the bias in *tiered*ImageNet

Since *tiered*ImageNet is a subset of ImageNet, its classes are the leaves of a directed acyclic graph which is a subgraph of the WordNet graph [6] (see Figure 2). Using this graph, it is possible to establish a semantic similarity between classes. We use the Jiang & Conrath pseudo-distance between classes [3], which is defined for two classes c_1 and c_2 as:

$$D^{JC}(c_1, c_2) = 2 \log |lso(c_1, c_2)| - (\log |c_1| + \log |c_2|) \tag{1}$$

[1] https://paperswithcode.com/task/few-shot-image-classification
[2] https://github.com/sicara/semantic-task-sampling

where $|c|$ is the number of instances of the dataset with class c, and $lso(c_1, c_2)$ is the lowest superordinate, *i.e.* the most specific common ancestor of c_1 and c_2 in the directed acyclic graph.

From this pseudo-distance, we define the *coarsity* κ of a task $\mathbf{T_C}$ constituted of instances from a set of classes $\mathbf{C}$ as the mean square distance between the classes of $\mathbf{C}$ *i.e.*

$$\kappa(\mathbf{T_C}) = \underset{c_i, c_j \in \mathbf{C} c_i \neq c_j}{\text{mean}} D^{JC}(c_i, c_j)^2 \tag{2}$$

This coarsity is an indicator of how semantically close are the classes that constitute a task. As shown in [2], on datasets derived from ImageNet, this measure is closely linked to the visual similarity between items of these classes.

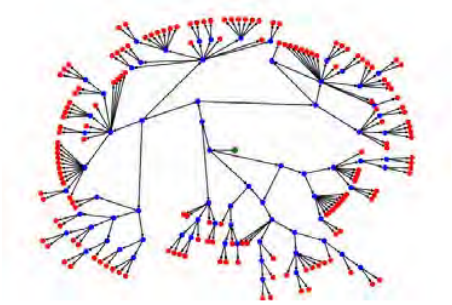


Figure 2. Subgraph of the Directed Acyclic Graph WordNet [6] spanning the 160 classes of *tiered*ImageNet's test set, which are shown in red. The root (in green) corresponds to the concept of "entity". This is a Directed Acyclic Graph. Best viewed in color.

4.2. Generate more informative tasks with semantic task sampling

We define a unique, reproducible set of testing tasks to evaluate all models. This testbed is built with a dual objective:

- We want tasks with a smooth repartition in terms of coarsity to ensure that the testbed also evaluates the ability of a model to distinguish between classes close to each other. Providing a good span of coarsities also allows to compare models on different types of tasks: a model might be better for coarse tasks but not for fine-grained tasks.
- This first objective inherently creates a bias towards classes with many neighboring classes. However, we want our testbed to be balanced, *i.e.* all images must be sampled roughly as many times as the others [3].

To achieve these goals, we define a semantic task sampler based on the Jiang & Conrath pseudo-distance (see Equation 1). We build an initial potential matrix [5] $\mathcal{P}_0$ such that $\mathcal{P}_0(i,j) = e^{-\alpha D^{JC}(c_i, c_j)}$ with $\alpha \in \mathbb{R}_+$ an arbitrary scalar. For the first task, the probability for a pair of classes (c_i, c_j) to be sampled together is proportional to $\mathcal{P}_0(i,j)$. To enforce that the testbed is balanced, once the $t-1^{\text{th}}$ task is sampled we update the number $occ_t(i)$ of occurrences of class c_i in previous tasks. Then we update the potential matrix to penalize classes with higher values of occ_t:

$$\mathcal{P}_t(i,j) = \mathcal{P}_0(i,j) \times \exp(-\beta \frac{occ_t(i) + occ_t(j)}{\max_k(occ_t(k))}) \tag{3}$$

with $\beta \in \mathbb{R}_+$ an arbitrary scalar. Intuitively, a larger α gives more weight to pairs of semantically close classes, while a larger β forces a stricter balance between classes.

We then sample instances from these classes uniformly at random. As shown in Figure 1b, our 5000-tasks testbed gives far greater representation to fine-grained tasks compared to a uniformly sampled testbed. Our testbed offers a wide range and balance of task coarsities, allowing to test models on both coarse and fine-grained tasks, while the uniformly sampled testbed only allows the evaluation on coarse tasks.

4.3. Experiments

We conducted the necessary experiments to bring out the need for novel few-shot classification benchmarks and showcase the limitations of state-of-the-art methods in more challenging settings. All parameters of our experiments can be found on our publicly available code [4].

The results are detailed in our paper [1] and show that changes in coarsity are strongly linked with the performance of Few-Shot Learning models.

[3] In the case of *tiered*ImageNet, which presents as many images for each class, this is equivalent to ensuring the balance between classes.

[4] https://github.com/sicara/semantic-task-sampling

4.4. Conclusion

We used to define a few-shot classification task by its number of ways and its number of shots, addressing n-way k-shot classification as an indivisible problem. What we did here can be seen as a novel framework, in which the number of classes is not sufficient to define a task: we need to know what these classes are. In the same fashion, future works may go beyond tasks defined by their number of shots, and consider which images are chosen for the support set.

In this work, we addressed what we believe to be a very limiting bias of current Few-Shot Learning benchmarks *i.e.* a bias towards coarse tasks. We chose to tackle this particular shortcoming because we observed that it was the main difference between academic benchmarks and the industrial applications of Few-Shot Learning that we encountered. However, many more limitations of few-shot learning benchmarks are yet to address: the fixed shape of the tasks, the strict balance in both support and query sets, the empty overlap between large-scale classes (currently only used for base training) and few-shot classes, no prior in the choice of support instances, and many other of which we did not think yet. We believe that addressing these shortcomings must be considered a priority in our field, and we encourage any and all who agree to join us in this effort.

References

[1] Etienne Bennequin, Myriam Tami, Antoine Toubhans, and Céline Hudelot. Few-shot image classification benchmarks are too far from reality: Build back better with semantic task sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4767–4776, 2022. 2, 3

[2] Thomas Deselaers and Vittorio Ferrari. Visual and semantic similarity in imagenet. In *CVPR 2011*, pages 1777–1784. IEEE, 2011. 3

[3] Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997. 2

[4] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 2

[5] Chenghao Liu, Zhihao Wang, Doyen Sahoo, Yuan Fang, Kun Zhang, and Steven CH Hoi. Adaptive task sampling for meta-learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 752–769. Springer, 2020. 3

[6] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 2, 3

[7] Tiago Ramalho and Marta Garnelo. Adaptive posterior learning: few-shot learning with a surprise-based memory module. *arXiv preprint arXiv:1902.02527*, 2019. 2

Robustness of Neural Networks Based on MIP Optimization

Ramzi Ben Mhenni[1], Mohamed Ibn Khedher[1] and Stéphane Canu[2]
[1] *IRT - SystemX, 8 Avenue de la Vauve, 91120 Palaiseau, France*
[2] *INSA Rouen Normandie, 685 Avenue de Universite 76800 Rouen, France*
{ramzi.ben-mhenni, mohamed.ibn-khedher}@irt-systemx.fr, stephane.canu@insa-rouen.fr

Keywords:
Neural network, Adversarial attack, Mixed Integer Programming, Branch-and-Bound algorithm.

Abstract:
Even though Deep Learning methods have demonstrated their efficiency, they do not currently provide the expected security guarantees. They are known to be vulnerable to adversarial attacks where malicious perturbed inputs lead to erroneous model outputs. The success of Deep Learning and its potential use in many safety-critical applications has motivated research on formal verification of Neural Network models. A possible way to find the minimal optimal perturbation that change the model decision (adversarial attack) is to transform the problem, with the help of binary variables and the classical *bigM* formulation, into a Mixed Integer Program (MIP). In this paper, we propose a global optimization approach to get the optimal perturbation using a dedicated branch-and-bound algorithm. A specific tree search strategy is built based on greedy forward selection algorithms. We show that each subproblem involved at a given node can be evaluated *via* a specific convex optimization problem with box constraints and without binary variables, for which an active-set algorithm is used. Our method is more efficient than the generic MIP solver Gurobi and the state-of-the-art method for MIPs such as MIPverify.

1 Introduction

Evaluating Robustness to adversarial examples is a very active research field in Deep Learning, which aims at finding an adversarial attack $\boldsymbol{a} \in \mathbb{R}^N$ *"–perturbed inputs vector that are very similar to some regular input but for which the output is radically different [Szegedy et al., 2014]–"*, approximating original data vector $\boldsymbol{x} \in \mathbb{R}^N$. This problem can be tackled through the minimization of the least-squares approximation error constrained by the change in the model decision [Carlini and Wagner, 2017].

$$\min_{\boldsymbol{a}} \; d(\boldsymbol{a}-\boldsymbol{x}) \;\; \texttt{s.t.} \begin{cases} \max_{i \neq y}(f_i(\boldsymbol{a})) > f_y(\boldsymbol{x}) \\ \boldsymbol{a} \in X_{valid} \end{cases}$$

where $d(-,-)$ denote a distance metric that measures the perceptual similarity between two input images and $y(\boldsymbol{x})$ is the true label of the input $\boldsymbol{x}$.

To evaluate the robustness of a neural network, several approaches are proposed in the state of the art, that can be grouped according to the formulation of the problem into: feasibility, optimization and reachability problems. A feasibility problem consists in converting the neural network to a feasibility problem for the existence of a counter-example [Katz et al., 2017, Ehlers, 2017, Bunel et al., 2018]. The reachability approach based consists in computing all the reachable set by the neural network and given the input dataset. Then, it checks if this set verify the desired constraints [Xiang et al., 2018, Gehr et al., 2018, Xiang et al., 2018]. Generally, the reachable dataset is computed by approximation. Finally, the optimization approaches consist in computing the maximum perturbation that can be applied to input data without changing the decision of the neural network [Tjeng et al., 2019, Lomuscio and Maganti, 2017]. Our approach lies in the optimization based approaches.

In this paper, we build a dedicated branch-and-bound algorithm for this problem. One key element in our work relies on showing that each node evaluation involved in the search tree can be performed through the optimization of con-

vex problem without binary variables and we build a specific tree-search exploration strategy. Section 2 describes the branch-and-bound algorithm principle, details our implementation strategy and links the node evaluation. Then, numerical results are given in Section 3, where the running time of the proposed implementation is compared to the MIP resolution with the Gurobi solver. A conclusion and directions for future work are finally given in Section 4.

1.1 Formulating Robustness as a Mixed Integer Program (MIP)

In this paper, we are focusing on Feed-Forward Neural Network where each neuron in a layer is connected with all the neurons in the previous layer. To simplify the process, we take a simple example of a network with one hidden layer. The problem can be written as follows:

$$\min_{\boldsymbol{a},\boldsymbol{h},\hat{\boldsymbol{h}}} \; d(\boldsymbol{a}-\boldsymbol{x}) \;\; \texttt{s.t.} \begin{cases} \boldsymbol{h} = \mathbf{W}\boldsymbol{a} + \boldsymbol{\beta}^{\boldsymbol{w}} \\ \hat{\boldsymbol{h}} = max(\boldsymbol{h}, 0) \\ \boldsymbol{s} = \mathbf{V}\hat{\boldsymbol{h}} + \boldsymbol{\beta}^{\boldsymbol{v}} \\ s_i \leqslant s_y \end{cases}$$

Formulating ReLU : Let $\hat{\boldsymbol{h}} = max(\boldsymbol{h}, 0)$ and $-M \leqslant \boldsymbol{h} \leqslant M$. There are three possibilities for the phase of the ReLU. If $\hat{\boldsymbol{h}} = 0$, we say that such a unit is stably inactive. Similarly, if $\boldsymbol{h} = \hat{\boldsymbol{h}}$, we say that such a unit is stably active. Otherwise, the unit is unstable. For unstable units, we introduce an indicator decision variable b witch indicates if the ReLU is active or not:

$$b_i = \begin{cases} 1 & \text{ReLu is active} \\ 0 & \text{ReLu not active} \end{cases}$$

Then, Evaluating Robustness problem is formulating as a Mixed Integer Program:

$$\mathcal{P} \; : \; \min_{\boldsymbol{a},\boldsymbol{h},\hat{\boldsymbol{h}},\boldsymbol{b}} \; d(\boldsymbol{a}-\boldsymbol{x}) \;\; \texttt{s.t.} \begin{cases} \boldsymbol{b} \in \{0,1\} \\ \boldsymbol{h} = \mathbf{W}\boldsymbol{a} + \boldsymbol{\beta}^{\boldsymbol{w}} \\ \hat{\boldsymbol{h}} \geqslant \boldsymbol{h} \; ; \hat{\boldsymbol{h}} \geqslant 0 \\ \hat{\boldsymbol{h}} \leqslant M\boldsymbol{b} \\ \hat{\boldsymbol{h}} \leqslant \boldsymbol{h} + M(1-\boldsymbol{b}) \\ \boldsymbol{s} = \mathbf{V}\hat{\boldsymbol{h}} + \boldsymbol{\beta}^{\boldsymbol{v}} \\ s_i \leqslant s_y \end{cases}$$

2 Branch-and-bound exploration

The branch-and-bound principle [Wolsey, 1998] relies on alternating between a *separation* step and an *evaluation* step. The first one consists in dividing a difficult problem into disjoint subproblems which are easier to solve, building a binary search tree. In our case, each separation corresponds to the decision: $b_{k_j} = 1$ or $b_{k_j} = 0$, for some variable b_{k_j} to be defined (see Figure 1). At node i, decisions have been made concerning

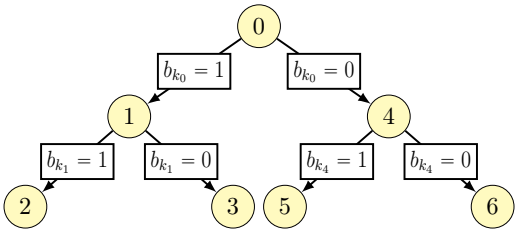


Figure 1: Separation step in a binary search tree: each node corresponding to the optimization problem $\mathcal{P}^{(n)}$, is divided into two children nodes obtaining by constraining one variable to be zero or non-zero.

the nullity of some variables: variables indexed by S^1 are non-zero, those indexed by S^0 are zero (and therefore are removed from the optimization problem) and decisions must still be made concerning the remaining undetermined variables, indexed by $\bar{\mathrm{S}}$.

The evaluation of node i of the search tree is based on the computation of a lower bound on $\mathcal{P}^{(i)}$, let say $z_\ell^{(i)}$ witch is obtained by the continuous Relaxation of the binary Variables:

$$\mathcal{P}^{\mathcal{R}^{(i)}} : \min_{\boldsymbol{a},\boldsymbol{h},\hat{\boldsymbol{h}},\boldsymbol{b}} \; d(\boldsymbol{a}-\boldsymbol{x}) \;\; \texttt{s.t.} \begin{cases} \boldsymbol{b} \in [0,1] \\ \boldsymbol{h} = \mathbf{W}\boldsymbol{a} + \boldsymbol{\beta}^{\boldsymbol{w}} \\ \hat{\boldsymbol{h}} \geqslant \boldsymbol{h} \; ; \hat{\boldsymbol{h}} \geqslant 0 \\ \hat{\boldsymbol{h}} \leqslant M_u\boldsymbol{b} \\ \hat{\boldsymbol{h}} \leqslant \boldsymbol{h} - M_l(1-\boldsymbol{b}) \\ \boldsymbol{s} = \mathbf{V}\hat{\boldsymbol{h}} + \boldsymbol{\beta}^{\boldsymbol{v}} \\ s_i \leqslant s_y \end{cases}$$

The continuous Relaxation $\mathcal{P}^{\mathcal{R}^{(i)}}$ will indicates if node i can contain an optimal solution. More precisely, let z_U denote the best known value of the objective function in $\mathcal{P}$ at a current step of the procedure—which is an upper bound on the optimal value. If $z_\ell^{(i)} \geqslant z_U$, then the node can be pruned. Otherwise, this node is separated into two subproblems according to some new decision: $b_{k_j} = 1$ or $b_{k_j} = 0$? The practical efficiency mostly depends on the tightness of the computed bounds (evaluation step) and on the branching and exploration strategies that are implemented (branching step).

2.1 Evaluation step

Lower bound and convex relaxation. At any node i of the search tree, a lower bound on $\mathcal{P}^{(i)}$ is obtained by solving $\mathcal{P}^{\mathcal{R}^{(i)}}$. Indeed, thanks to the box constraint $\|\boldsymbol{h}\|_\infty \leqslant M$ and convexity property, one has therefore the continuous relaxation $\mathcal{P}^{\mathcal{R}^{(i)}}$ is equivalent to $\mathcal{R}^{(i)}$.

$$\mathcal{P}^{\mathcal{R}^{(i)}} \Longleftrightarrow \mathcal{R}^{(i)}$$

with

$$\mathcal{R}^{(i)} : \min_{\boldsymbol{a},\boldsymbol{h},\hat{\boldsymbol{h}}} \; d(\boldsymbol{a}-\boldsymbol{x}) \;\; \texttt{s.t.} \begin{cases} \boldsymbol{h} = \mathbf{W}\boldsymbol{a} + \boldsymbol{\beta}^{\boldsymbol{w}} \\ \hat{\boldsymbol{h}} \geqslant \boldsymbol{h} \; ; \hat{\boldsymbol{h}} \geqslant 0 \\ \hat{\boldsymbol{h}} \leqslant \frac{\boldsymbol{h}+M}{2} \\ \boldsymbol{s} = \mathbf{V}\hat{\boldsymbol{h}} + \boldsymbol{\beta}^{\boldsymbol{v}} \\ s_i \leqslant s_y \end{cases}$$

Since both problems are defined on the same feasible domain, $\{\hat{\boldsymbol{h}} \geqslant \boldsymbol{h} \; ; \hat{\boldsymbol{h}} \geqslant 0 \; ; \hat{\boldsymbol{h}} \leqslant \frac{\boldsymbol{h}+M}{2}\}$ is a *convex relaxation* of the constraint $\hat{\boldsymbol{h}} = max(\boldsymbol{h}, 0)$ (see. figure 2). Let us remark that this well-known result (*the continuous, convex, relaxation of the ReLu*) is only valid under additional boundedness assumptions on the solution space, such as the box constraints that were introduced in problem P.

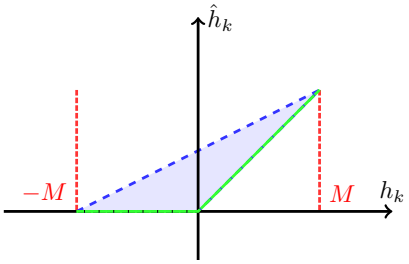


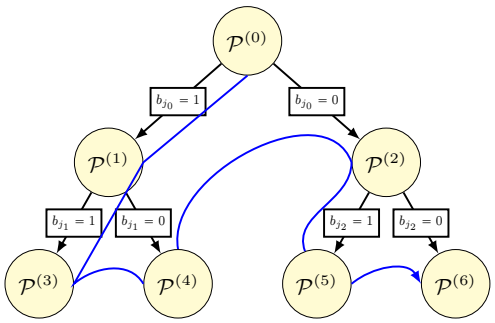
Figure 2: The tightest linear convex relaxation.

2.2 Branching rules and exploration strategy

The branching rule selects the index j of the variable which is used in order to subdivide problem $\mathcal{P}^{(i)}$ (see Figure 1). We propose to exploit the solution of $\mathcal{R}^{(i)}$, by selecting the variable with the highest absolute value in the minimizer:

$$j = \arg\max_{n \in \bar{\mathbb{S}}} \hat{h}_n^{(i)}.$$

This choice aims at selecting first the variables which are more likely to be nonzero at the optimal solution.



We use *depth-first search*, and our branching rule is based on selecting the binary variable, say b_i, with the highest value in the solution of the relaxed problem. We branch *up* first, that is, we first explore the branch corresponding to the decision $b_i = 1$. This strategy, similar to the principle of greedy forward selection algorithms [Mhenni et al., 2020], aims at activating first the most prominent nonzero variables in $x_{n_i} \neq 0$, therefore focusing on quickly finding satisfactory feasible solutions and subsequent upper bounds of good quality. Our proposed implementation is summarized in Algorithm 1, where L contains the queue of subproblems and $\widehat{\boldsymbol{x}}$ denotes the best known solution along the exploration. The Branch-and-

0. **Initialization:** $L \leftarrow \{\mathcal{P}^{(0)}\}$; $z_U = +\infty$; $\widehat{\boldsymbol{a}} := 0$.
1. **Optimality:** if $L = \varnothing$, then return the optimal solution $\widehat{\boldsymbol{a}}$.
2. **Node selection:** choose a subproblem $i \in L$ by depth-first search and remove it from L.
3. **Node evaluation:** compute $z_\ell^{(i)}$.
4. **Pruning:**
   - If $z_\ell^{(i)} \geqslant z_U$, prune node i and return to step 1.
   - If $z_\ell^{(i)} < z_U$:
   – If $\boldsymbol{b}^{\mathcal{R}(i)} \in \{0,1\}$, then $z_U \leftarrow z_\ell^{(i)}$ and $\widehat{\boldsymbol{a}} \leftarrow \boldsymbol{a}^{\mathcal{R}(i)}$. Prune node i and return to step 1.
5. **Branching:** subdivide node i by (2.2) and add the two subproblems to L.

Algorithm 1: Branch-and-bound algorithm for $\mathcal{P}$.

Bound algorithm converge to the global minimum in a finite number of steps. In the worst case, an exhaustive search is done (no node could be pruned).

MIP_{Gurobi}			MIP_{Verify}			$B\&B_{HOME}$		
T	Nds	F	T	Nds	F	T	Nds	F
35	1800	4	27	-	4	7	1200	4

Table 1: Computational efficiency for robustness problems averaged over 100 instances. Computing time (T) number of explored nodes (Nds) and number of instances that did not terminate in 1 000 s (F).

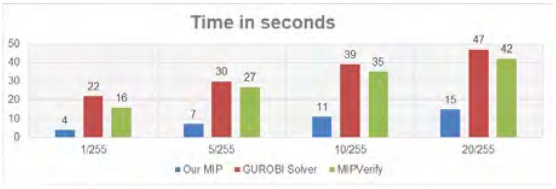


Figure 3: Computational Time for robustness problems averaged over 100 instances according to the intensity of the attack (maximum disturbance allowed in infinite norm)

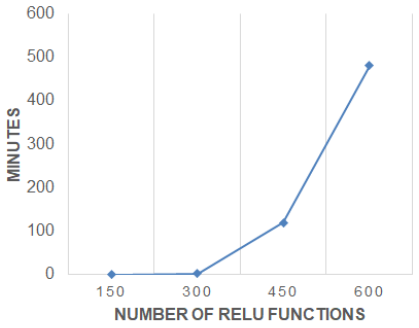


Figure 4: Computing time (Minutes) for robustness problems as a function of the number of ReLU in the model, average over 10 instances.

3 Performance Evaluation

We now evaluate the computational performance of our branch-and-bound strategy using Cplex MIP solver. We name this algorithm $B\&B_{HOME}$. Computing times are compared with the Gurobi Mixed quadratic programming solver (named MIP_{Gurobi}) [1] and MIP_{Verify} [2]. All methods are run on a UNIX machine equipped with 32 Go RAM and with four Intel Core i7 central processing units clocked at 2.6 GHz. For each instance, the running time is limited to 1 000 s. Note that we only focus here on the computational efficiency of algorithms which are guaranteed to find the global optimum $\mathcal{P}$; due to the lack of space we do not compare the obtained solutions to that of standard, suboptimal methods.

In order to evaluate the behavior of our method regarding the complexity of the model, we have varied the number of hidden layers from 1 to 4 (i.e. from 150 to 600 activation ReLU functions). Results averaged over 100 instances of each problem are given in Table 1 and Figure 3. $B\&B_{HOME}$ is much faster then MIP_{Gurobi} and MIP_{Verify} revealing the efficiency of our strategy. Most of all, we observe that the most important improvement achieved by $B\&B_{HOME}$ is due to the efficiency of our continuous relaxation: the computing time per node with the proposed formulation is at least 4 times smaller than that of MIP_{Gurobi}. Even with this improvement, the results in Figure 4 show the limit of our approach especially when the number of ReLU in the model increases. We can see that the complexity increases exponentially and becomes unfeasible in a reasonable time for complex models.

4 Conclusion

In this paper, we proposed a branch-and-bound algorithm which is able to find exactly the optimal attack. We have shown that such problems could benefit from dedicated resolution methods. Our algorithm outperforms Gurobi, which is considered as one of the best MIP solvers. The proposed exploration strategy exploits the sparsity of the searched solution, by preferring the activation of nonzero variables in the decision tree, conjugated with depth-first search. Moreover, evaluation of each node by continuous relaxation was recast as a specific convex problem without binary variables. Following the same principle, further works may include the building of more efficient relaxations, involving Lagrangian relaxation and specific cutting planes [Wolsey, 1998] for may also improve the quality of lower bounds computed at each node. **But unfortunately even with this improvement, the results show the limit of our approach especially when the number of ReLU in the model increases and we are still far from using large indistrual models**.

[1] https://www.gurobi.com/
[2] https://vtjeng.com/MIPVerify.jl/latest/

REFERENCES

[Bunel et al., 2018] Bunel, R., Turkaslan, I., Torr, P. H., Kohli, P., and Kumar, M. P. (2018). A unified view of piecewise linear neural network verification. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 4795–4804, USA. Curran Associates Inc.

[Carlini and Wagner, 2017] Carlini, N. and Wagner, D. (2017). Towards Evaluating the Robustness of Neural Networks. Number: arXiv:1608.04644 arXiv:1608.04644 [cs].

[Ehlers, 2017] Ehlers, R. (2017). Formal verification of piece-wise linear feed-forward neural networks. *CoRR*, abs/1705.01320.

[Gehr et al., 2018] Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., and Vechev, M. (2018). Ai 2: Safety and robustness certification of neural networks with abstract interpretation. In *Security and Privacy (SP), 2018 IEEE Symposium on*.

[Katz et al., 2017] Katz, G., Barrett, C. W., Dill, D. L., Julian, K., and Kochenderfer, M. J. (2017). Reluplex: An efficient SMT solver for verifying deep neural networks. In *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I*, pages 97–117.

[Lomuscio and Maganti, 2017] Lomuscio, A. and Maganti, L. (2017). An approach to reachability analysis for feed-forward relu neural networks. *CoRR*, abs/1706.07351.

[Mhenni et al., 2020] Mhenni, R. B., Bourguignon, S., and Idier, J. (2020). A greedy sparse approximation algorithm based on l1-norm selection rules. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5390–5394.

[Szegedy et al., 2014] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. Number: arXiv:1312.6199 arXiv:1312.6199 [cs].

[Tjeng et al., 2019] Tjeng, V., Xiao, K. Y., and Tedrake, R. (2019). Evaluating robustness of neural networks with mixed integer programming. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[Wolsey, 1998] Wolsey, L. A. (1998). *Integer Programming*. Wiley, New York, NY, USA.

[Xiang et al., 2018] Xiang, W., Tran, H., and Johnson, T. T. (2018). Output reachable set estimation and verification for multilayer neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):5777–5783.

[Xiang et al., 2018] Xiang, W., Tran, H.-D., and Johnson, T. T. (2018). Reachable set computation and safety verification for neural networks with relu activations. *In Submission*.

A method and metrics to evaluate confidence score performances

Berthelot Loris, Troya-Galvis Andrés, Christophe Gouguenheim, Ahmad Berjaoui, Marc Spigai

Abstract – Machine Learning models often suffer from poor calibration, meaning that the predicted probabilities do not match the real performance of the model. Confidence scores are one of the solutions to the calibration problem of Machine Learning models and yet, there seems to be no agreement towards a general method to test the performances and quality of confidence scores. Our work attempts to provide a such method along with some metrics suited to compare the confidence score with one another.

1. Introduction

Results given by machine learning algorithms are poorly calibrated when we try to move out of the training distribution, generally resulting in over-confident, yet wrong predictions which is a problem. Hence, it is hard to trust machine learning models especially in critical domains. Some works have been focusing on increasing the robustness of machine learning models while some others have been trying to support predictions with a confidence score which is supposed to quantify the probability of the model to fail or succeed in its prediction. However, it seems that there is no general and formal definition of what a confidence score is, and, to our knowledge, the literature seems to lack of a general method to assess the performances and the quality of confidence scores. Nevertheless, existing works aim their confidence scores to be robust to label noise, Out-Of-Distribution data or Adversarial Attacks, among others, but the evaluation procedures remain very specific to each method and there is a lack of metrics to quantify how robust a score is with regard with those factors. We provide a method as well as a set of metrics to evaluate the quality of a given confidence score and allow comparison between them. We show that the proposed metrics can reflect the desired properties of confidence scores and compare three variants of a confidence score based on conformal predictions with the raw softmax predictions of an image classification model.

2. Related work

(Berjaoui 2021) summarizes the state of the art for confidence metrics, including confidence scores. There are two main approaches for building a confidence score. The first one, is based on the k-Nearest Neighbors idea: one should be confident in a prediction if the neighbors of the predicted sample have the same label that the one being predicted (i.e. the sample is in the high density area of the given label). The second one, consists of training a meta-model with the goal of predicting whether the underlying algorithm is right or wrong given a prediction. In that case, the confidence score performances are usually given by the AUC ROC (Area Under the Recall Receiver Operating Characteristic Curve) (Hendrycks 2016) such as in (Mandelbaum 2017), (Corbière 2020) and (Chen 2019). However, it would be interesting to have more information about the correct and incorrect distributions in order to define some empirical thresholds, allowing automatic decision when the confidence scores are in the area of low density overlap (between correct and incorrect distribution) and human handling for score within the high density area.

Another desirable property of a confidence score is the ability to reflect the probability of a system to be right or wrong. To the best of our knowledge, this approach is referred as calibration in the literature, (Dormann 2019), (Vuk 2006) and is usually used directly on models and has not been used on confidence score. Thus, we suggest a metric to quantify the performances of confidence scores in that regard.

3. On the evaluation of confidence scores

Our main focus in this work is to set up a method as well as a set of metrics to be used for confidence score performance evaluation. Next, we present two different ways of thinking about the quality of a confidence score, and discuss some desirable properties of a *perfect* score.

3.1 Confidence score as a probability

Intuitively, a confidence score is an indicator which quantifies how much one can trust a prediction of a given machine learning model. That being said, a confidence score should be correlated to the probability that a machine learning model is right or wrong for each of its predictions. On a held-out (validation) dataset (composed of unseen data), 20% of the predictions with a confidence score of 0.2 should be correct, 70% for a score of 0.7, etc. The distribution of scores should have the widest possible range (between 0 and 1) to give as much information as possible. This is to prevent corner cases where the confidence scores close to ideal theoretical values but useless in practice. For example, on a unbalanced dataset where a model always predicts the majority class, one could define a confidence score that always return the accuracy of the model. Such score would be compliant to the previous definition butt does not give any insight on the actual prediction. That being said, we can use calibration plots to derive a confidence score quality metric. Indeed, if we split correct and incorrect predictions on a validation dataset, a close to perfect score should have a low ratio of correct predictions for low scores and a large ratio for large scores. We illustrate this principle on Fig 1. The ideal confidence score should be as close as possible to this graph. Thus, such distance measurement can be used as a quality metric for confidence scores. Note that the lowest probability

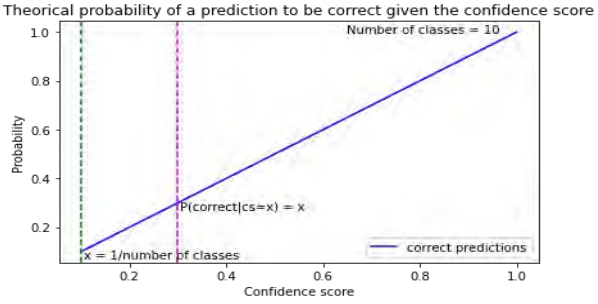


Fig. 1 Illustration of a perfect confidence score with respect to probabilities.

is not 0 because in the worst case, the model prediction is random, so the probability of the prediction being correct is $\frac{1}{k}$, with k the number of classes.

3.2 Confidence score as a binary classifier

A confidence score can also be considered as a binary classifier trying to predict whether a given prediction is correct or not. Binary classifier performances are usually qualified by the Area Under the Curve (AUC) ROC (Hendrycks 2016) which gives insights about the tradeoff between true positives and false positives rates, and thus about the separation of the two classes. However, this doesn't tells much about the relative position of the correct and incorrect prediction distributions.

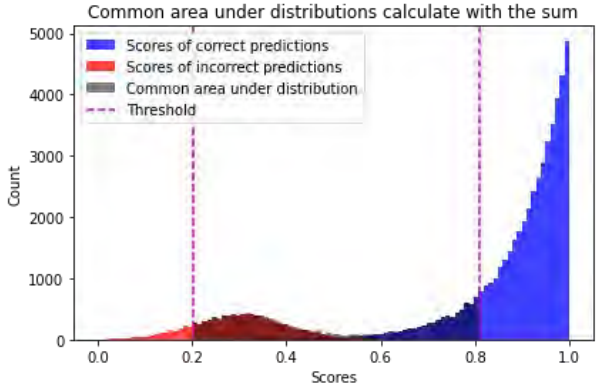


Fig. 2 : Correct and incorrect distributions with overlapping area to compute metrics.

Ideally, a good score should allow us to separate as much as possible the correct and incorrect predictions. Then, it would be possible to empirically define automatic decision thresholds based on the confidence score. In order to measure this property, we suggest to use the Common Area Under the Distribution which is equal to the area within the range of scores with both correct and incorrect predictions (See Fig 2.). In practice, this area can be approximated by computing the ratio of the number of samples within the range of scores with both correct and incorrect predictions over the total number of samples. Note that the range of scores considered can be adjusted in function of the criticity of the use-case by ignoring the p and (1-p) quantiles of the correct and incorrect distributions respectively.

On the example presented in Fig 2, the range of scores where the correct and incorrect distributions overlap is between 0.2 and 0.8. In this case, it is possible to empirically define two thresholds, 0.2 and 0.8. For a sample with a confidence score lower than 0.2, one directly assume the prediction is incorrect (with high confidence). Respectively, for a sample with a confidence score greater than 0.8, one can assume prediction is correct. However for predictions with a confidence scores between 0.2 and 0.8, the decision should be handled to a human. A *perfect* confidence score should present two disjoint distributions and automatic decision would be possible on every sample.

3.3 Desirable robustness

In (Berjaoui 2021), we can find three scenarios where a confidence score should be robust.

The first one is robustness against noisy labels in data which consist of observing the behavior of the confidence score when we train the underlying algorithm with a percentage x of mislabeled data. A good confidence score should show the same distribution shape for incorrect and correct predictions with a shift towards lower score. Indeed we should be less confident in the model since it learnt on partial mislabeled data. But we should keep the correct/incorrect probabilities for every value of the confidence score.

The second factor is the robustness to Out Of Distribution (OOD) data. This is the main factor to take into consideration when trying to industrialize a machine learning solution. In machine learning, the training is done on a dataset which should be representative of the reality as much as possible. However, in practice, training data is only a partial representation. Hence, it is likely to encounter scenarios that the model has never seen, or to have outlier data being fed to a model in production, leading to incorrect predictions and unexpected behavior. A good confidence score should give less confident scores to OOD data.

The last factor is robustness against Adversarial Attacks (Goodfellow 2014) (AA). AA consist of adding very little but carefully crafted noise in test validation inputs which may be insignificant for human eyes but which highly influences the prediction of a machine learning model towards a bad prediction. A confidence score robust to AA should give low confidence scores to adversarial samples, since if the model itself is sensitive to adversarial attacks, the label will be wrong most likely. Hence if we plot the correct and incorrect distributions, we should keep the same shape that previously with a shift towards lower scores.

4. Experiments

4.1 Compared scores

Conformal Predictions (CP), introduced (Vovk 2005) and summarized by (Zeni 2020) and (Angelopoulos A. N. 2021) is a framework which aims to produce a *prediction set* (multiple probable labels) instead of a single label prediction, with the guarantee that the true label is in the prediction set, given an error rate, with only the iid assumption needed. CP seek to produce sets of the smallest size and to be adaptive, meaning a hard sample will have a large set and an easy one will have a small one. Based on this framework, we derive three confidence scores on which we will test our method:

1. CPCS, which considers the size of the *prediction sets* as well as the softmax values.

2. CP-OOD, which considers the size of the *prediction sets* as well as a term which measures how far the sample is from training distribution.

3. CP-Mixt, which aggregates the two previous scores.

The formal definition of such scores are out of the scope of this paper. We compare these three scores with the raw softmax outputs of the model as a baseline, in order to validate the proposed methodology.

Table 1: Metric values on XVIEW to compare confidence scores. The model accuracy is equal to 57.29% on the clean validation set. The noisy training has been done with 30% of data randomly labeled, the rotation value is equal to 90°.

	AUC ROC				Sum				Distance			
	Clean	Noisy	Rotated	Negative	Clean	Noisy	Rotated	Negative	Clean	Noisy	Rotated	Negative
Softmax	0,7931	0,6867	0,7937	0,6306	0,0204	0,0277	0,0058	**0,0627**	20,63	21,64	24,66	35,09
CPCS	**0,807**	**0,7768**	**0,8136**	0,6441	0,035	0,0073	0,0131	0,0131	27,1	23,41	25,93	**27,04**
CP-OOD	0,6948	0,6782	0,7294	0,5995	0,1316	0,0614	**0,1314**	0,0365	20,77	16,74	21,22	38,57
CP-Mixt	0,75	0,7301	0,7738	**0,6469**	**0,1418**	**0,0848**	0,1036	0,0234	**18,6**	**15,61**	**20,29**	35,9

4.2 Datasets

We did our experiments a modified version of XVIEW adapted to classification. That dataset is composed of 11000 images (3x32x32) split evenly into 10 classes. The training set is composed of 10000 images and the validation set of 1000 images. We split the validation set into calibration (30%) and validation (70%) sets for conformal prediction calibration.

4.3 Models

The machine learning model used, is the pre-trained ResNet18 available in Pytorch. We only modified the classifier layers to fit better a problem with only 10 classes (L(512, 128), L(128, 64), L(64, 10)). We then fine-tuned ResNet on XVIEW with a learning rate of 1e-5.

4.4 Results

On XVIEW, following the method and according to metrics previously presented, confidence scores based on conformal predictions produced better performances than softmax, see Table 1. Conformal prediction-based confidence score outclass softmax with no modification in the validation set, when the model is trained on a noisy dataset, when a rotation is applied on the validation set, when the validation set is composed of negative images of the original validation set and when we create a dataset with classes not learned by the model. For illustration purposes, and to understand how the metrics reflect the quality differences between scores, we compare the CP-OOD score with the raw softmax scores, a more rigourous comparison of several existing scores is out of the scope of this paper.

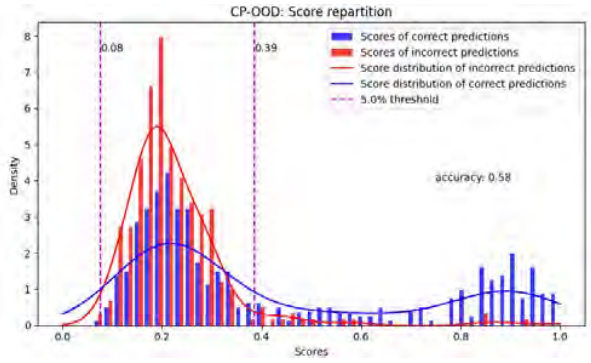


Fig 3: Correct and incorrect prediction distribution on raw validation dataset for CP-OOD.

Next, we demonstrate the possible analysis and information that can be obtained by the proposed methodology. The analysis focus on the histogram of correct and incorrect predictions, as well as the calibration plots.

Fig 3 shows the plot of correct and incorrect distributions for the CP-OOD on the validation dataset. One can notice very few

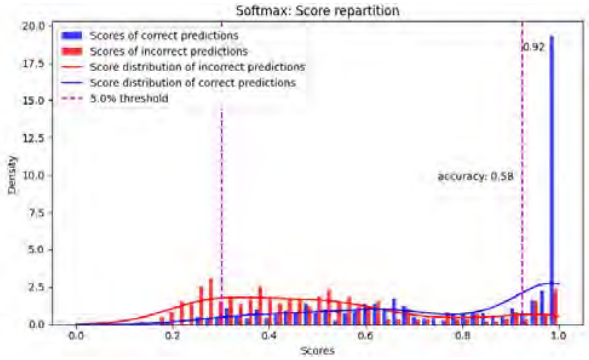


Fig 4: Softmax repartition for correct and incorrect predictions. According to this graph, if the softmax is between 0.92 and 1 it is more likely the prediction is true. Besides that, we can't say anything about the prediction

incorrect predictions with scores greater than 0.39 (only 5%) and very few correct predictions have a score lower than 0.08. Meaning the high density overlap area is between 0.08 and 0.39. We can then apply the idea described in section 3.2 to empirically define automatic decision thresholds. Indeed, according to the validation set, a prediction with a score higher than 0.39 is correct 95% of the time, a prediction with a score lower than 0.08 is likely incorrect. On the other hand, applying the same logic to the softmax distributions does not give much information. The high density overlap area is very wide, see Fig 4 (from 0.3 to 0.92) which reflects the lower quality of the softmax scores.

OOD experiments

In order to evaluate the behaviour of a confidence score w.r.t OOD data, we propose 3 different approaches. The first one is to create a validation dataset composed of every image of the initial validation set but rotated for a given or random angle (only valid if the data was not augmented by rotations during training). The second one, is to take the negative value of each pixel. The third one is to employ a close but different dataset. So we will use a dataset composed of 10 classes from XVIEW (Lam 2018) which were not considered on the employed training dataset. In the first two cases, a good confidence score should shift the correct and incorrect distributions towards low confidence scores because even if an image is correctly predicted, the model should be less confident in its prediction since it never saw an image like that, but the ratio of correct/incorrect probabilities for every value of the confidence score should remain the same. On the third case, a good confidence score should give a low confidence score for every sample. Those data can't be classified by our underlying model because it does not have the corresponding knowledge. No prediction should have a high confidence score because every prediction on that dataset will be misclassified.

Fig5 and Fig 6 present the correct and incorrect distributions of the negative images of the original validation dataset for CP-

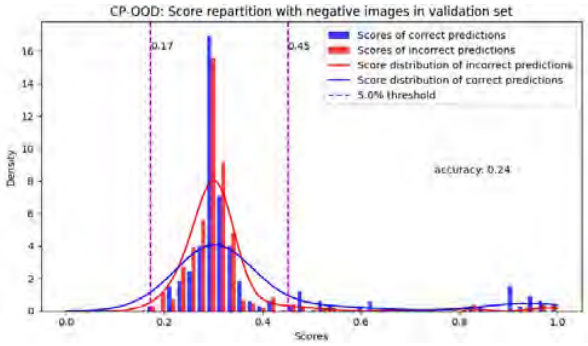


Fig 5: Correct and incorrect prediction distributions for CP-OOD on the negative validation dataset

OOD and Softmax. On Fig 5, we can see that almost every score, correct and incorrect is concentrated in a low range between 0.17 and 0.45. which is the desired behavior, since images are far from the training distribution.

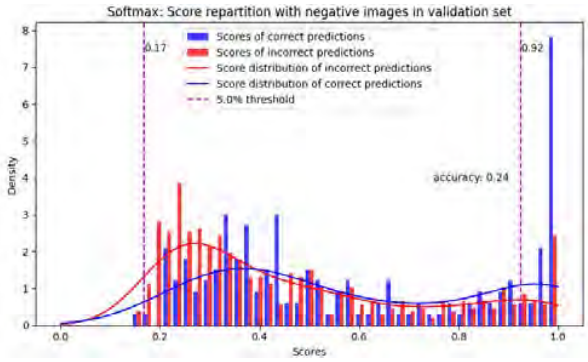


Fig 6: Correct and incorrect prediction distributions for softmax on the negative validation dataset

When it comes to softmax, the same behavior than previously is observed. The high density area almost goes from 0 to 1 which is far from the ideal score.

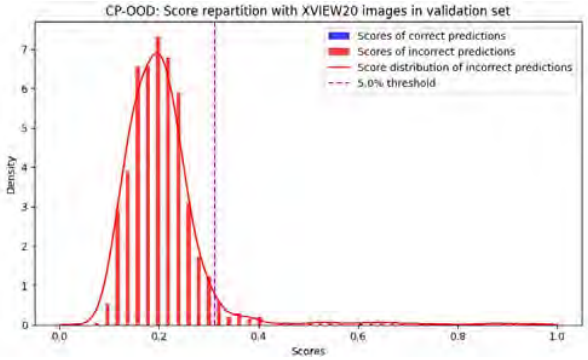


Fig 7: Incorrect prediction distribution for CP-OOD on dataset the underlying model was not trained on.

Fig 7 shows that CP-OOD behave as hoped on images the underlying model can't correctly classify (because the true classes aren't part of the training set). Every scores are very low, meaningful of the low confidence one should have in those predictions.

We also analyze the confidence scores from the probability perspective. Surprisingly, on XVIEW (Lam 2018), the softmax yield close to theoretical probability distribution which mean

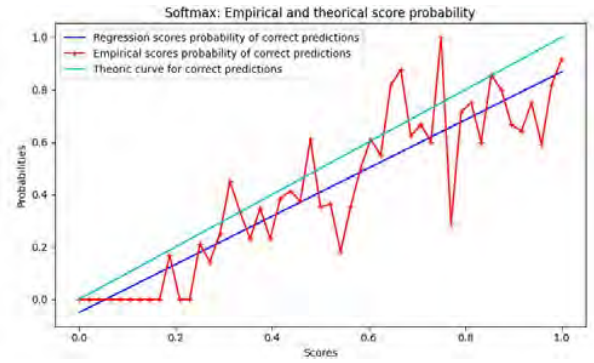


Fig 8: Softmax score probability on XView in normal settings.

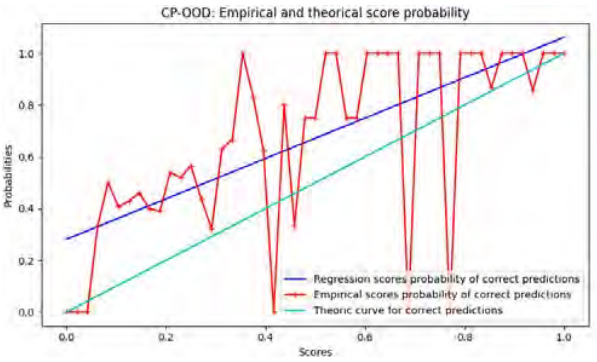


Fig 9: CP-OOD confidence probability of correct and incorrect predictions on XView in normal settings.

softmax is representative to the probability of a prediction to be correct. This can be drawn from Fig 8 with the low distance between the regression of the empirical points and the theoretical curve. On the other hand, CP-OOD regression curve, see Fig 9, is further to the theoretical one meaningful of a poorer calibration in the regard of probability. This is explained by the separation power. Indeed, the two criteria to evaluate a confidence score cannot be achieved at the same time. When developing a confidence score, or deciding which confidence score is the more suited for a given application, a tradeoff between separation power and probability calibration has to be done.

5. Conclusion

In this work, we proposed a method and some metrics to qualify confidence scores performances and different robustness to compare confidence scores. Our work is independent of the model and the dataset allowing our framework to be applied to any image classification problem. To validate our work, we compared the softmax output and few confidence scores based on conformal prediction on XVIEW (Lam 2018). CP-based confidence scores yield better performances according to our metrics in every settings of our method (as expected). Even if presented metrics highlight conformal predictions-based confidence score performances, it does not show the existing gap between softmax and those confidence scores as good as graphs do, new metrics still need to be introduced to have a better understanding of confidence scores. More comparisons with other existing confidence scores, such as ABC score (Jha 2019) with the use of other datasets such as MNIST (LeCun 1998) or CIFAR (Krizhevsky 2009), would be interesting to proceed.

Bibliography

Angelopoulos A. N., Bates, S. 2021. "A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification." *arXiv preprint arXiv:2107.07511.*

Berjaoui, A., Boyer, J.P., Adam, J.L. 2021. "Data based AI confidence evaluation."

Chen, T., Navratil, J., Iyengar, V., and Shanmugam, K. (2019). 2019. "Confidence scoring using whitebox meta-models with linear classifier probes." *arXiv preprint arXiv:1805.05396.*

Cohen, G., Afshar, S., Tapson, J., van Schaik, A. 2017. "EMNIST: Extending MNIST to handwritten letters." *doi: 10.1109/IJCNN.2017.7966217.*

Corbière, C., Thome, N., Saporta, A., Vu, T.-H., Cord, M., and Pérez, P. 2020. "Confidence estimation via auxiliary models." *arXiv preprint arXiv:2012.06508.*

Dormann, C. 2019. "Calibration of probability predictions from machine-learning and statistical models." https://onlinelibrary.wiley.com/doi/epdf/10.1111/geb.13070.

Goodfellow, I. J., Shlens, J., and Szegedy, C. 2014. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572.*

Hendrycks, D. and Gimpel, K. 2016. "A baseline for detecting misclassified and out-of-distribution examples in neural networks." *arXiv preprint arXiv:1610.02136.*

Jha, S., Raj, S., Fernandes, S., Jha, S. K., Jha, S., Jalaian, B., Verma, G., and Swami, A. 2019. " Attribution-based confidence metric for deep neural networks."

Krizhevsky, A. 2009. "Learning Multiple Layers of Features from Tiny Images."

Lam, D., Kuzma, R., McGee, K., Dooley, S., Laielli, M., Klaric, M., Bulatov, Y., McCord, B. 2018. "xView: Objects in Context in Overhead Imagery." *arXiv preprint arXiv:1802.07856.*

LeCun, Y., Cortes, C., Burges, C. J.C. 1998. "The MNIST dataset." *http://yann.lecun.com/exdb/mnist/.*

Mandelbaum, A. and Weinshall, D. 2017. "Distance based confidence score for neural network classifiers." *arXiv preprint arXiv:1709.09844.*

Vovk, V., Gammerman, A., and Shafer, G. 2005. *Algorithmic Learning in a Random World.* Springer.

Vuk, M., Curk, T. 2006. "ROC Curve, Lift Chart and Calibration Plot." https://mz.mf.uni-lj.si/article/view/44/41.

Zeni, G., Fontana, M., Vantini, S. 2020. "Conformal Prediction: a Unified Review of Theory and New Challenges." *arXiv preprint arXiv:2005.07972.*

Multi-Category Classification with Semantic Projection and Semantic Regularization

Arthur Ledaguenel [*1,2] **Céline Hudelot** [1] **Mostepha Khouadjia** [2]

Abstract

Neuro-symbolic techniques are growing increasingly popular to embed or infuse symbolic knowledge into deep neural networks. In this paper, we compare – in terms of accuracy, semantic robustness and complexity – two methods for binary multilabel classification under semantic constraints: Semantic Projection (SP) - a generalization of the HEX-graph methodology introduced in (Deng et al., 2014) – and Semantic Regularization (SR) – with the semantic loss defined in (Deng et al., 2014).

1. Introduction

The field of **Artificial Intelligence** (AI) has historically been divided into two families of approaches : **symbolic** techniques, which process logical variables through reasoning algorithms, and **statistical learning** techniques, which derive parametric models from huge amounts of data. In the past decade, **artificial neural networks** (ANNs), a sub-field of statistical learning, have taken the AI scene by storm and imposed themselves as state-of-the-art techniques in many tasks. Recently though, many researchers have pointed out the weaknesses of ANNs, especially in scarce data regimes, and advocated for the development of hybrid systems, often called **neuro-symbolic** techniques, involving both symbolic and neural components, to embed or infuse symbolic knowledge into ANNs and improve their accuracy, robustness or impose certain constraints on their outputs. In the past few years, many different neuro-symbolic techniques have been proposed in the literature. They vary in the way symbolic knowledge is represented and how it is embedded or infused in the neural network.

In this paper, we compare two methods for binary multilabel classification under semantic constraints : Semantic Projection (SP) - a generalization of the HEX-graph methodology introduced in (Deng et al., 2014) - and Semantic Regularization (SR) - defined in (Xu et al., 2018). Our main contribution is to develop a joint formalism to compare both techniques in terms of accuracy, semantic robustness and computational complexity in a principled way, and highlight common key quantities that both techniques compute and rely on. This re-framing leads to the conclusion that SP is superior to SR at inference time *by design*, and suggests that an intermediary loss functional between the two techniques could be more efficient.

[*]Equal contribution [1]MICS, CentraleSupelec, Saclay, France [2]IRT SystemX, Saclay, France. Correspondence to: Arthur Ledaguenel <arthur.ledaguenel@centralesupelec.fr>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

2. Formalism

2.1. Standard neural techniques

Our study tackles with the task of **binary multilabel classification** : a deep neural network $\mathcal{M}_\theta(X)$ is trained on a dataset $(X^i, y^i)_{1\leq i\leq n}$ with $X^i \in \mathbb{R}^d$ and $y^i \in \{0,1\}^k$ to predict an instantiation over a set of categories $\{Y_j\}_{1\leq j\leq k}$. When no structure is assumed, the dimensionality of the problem is exponential with the number of categories k and the task becomes quickly intractable for large target spaces.

To circumvent this limit, usually one of two *structural priors* is enforced :

1. **Strictly mutually exclusive categories** : this means that no two categories can be activated at the same time, and turns the problem into a single multi-valued variable classification task (i.e. $y^i \in [\![1,k]\!]$ instead of $y^i \in \{0,1\}^k$).
   This is usually implemented by applying a *softmax layer* on the last activation scores produced by the network.

2. **Independent multilabel regression** : this means that all combinations of variables are possible and that the probability of a category being activated depends only on the activation score of that category.
   This is implemented by applying a *sigmoid layer* on the last activation scores produced by the network.

This paper explores a formalism that enables to enforce a

more complex range of *structural priors* compared to the two aforementioned techniques.

2.2. Neuro-symbolic techniques

Both techniques build on top of an existing neural network $\mathcal{M}_\theta$ of which the final pooling or activation layer (such as *softmax* or *sigmoid*) has been removed. The model then produces a vector of activation scores $\mathbf{a}_\theta \in \mathbb{R}^k$, upon which each technique applies additional computations.

From this score vector, any technique must :

> **Training** : compute a loss functional to be optimized through gradient descent given labeled samples (and/or unlabeled samples in a semi-supervised setting).
>
> **Inference** : predict an instantiation which corresponds to the mode (ie. the most probable instantiation) of a certain probability distribution on the instantiation space $\{0,1\}^k$.

Remark 2.2.1. It is important here to notice that the system **does not need to compute the full distribution** it enforces, but solely certain values relative to it (such as the mode for inference, and the probability of a certain instantiation for the negative log-likelyhood loss).

Example 2.2.2. We can for instance get back to the case of independent multilabel regression.
Given label scores $\mathbf{a} := (a_i, ..., a_k) \in \mathbb{R}^k$, one can define the independent multilabel unnormalized distribution as :

$$\mathbf{E}_{\mathcal{I}(\mathbf{a})} : \{0,1\}^k \to [0,1], \mathbf{y} \mapsto \prod_{1 \leq i \leq k} e^{a_i . y_i}$$

The standard independent multilabel probability distribution is then obtained after normalization :

$$\mathbf{P}_{\mathcal{I}(\mathbf{a})}(\mathbf{y}) := \frac{\mathbf{E}_{\mathcal{I}(\mathbf{a})}}{\mathsf{Z}_{\mathcal{I}(\mathbf{a})}}$$

with $\mathsf{Z}_{\mathcal{I}(\mathbf{a})} := \sum_{\mathbf{y}} \mathbf{E}_{\mathcal{I}(\mathbf{a})}(\mathbf{y})$ the partition function.

The loss functional computed given a labeled sample $(X, \mathbf{y}_{true})$ is the standard **negative log-likelihood** :

$$\mathcal{L}_{\mathcal{I}}(\theta, X, \mathbf{y}_{true}) = -\log \mathbf{P}_{\mathcal{I}(\mathbf{a}_\theta)}(\mathbf{y}_{true})$$

with $\mathbf{a}_\theta := \mathcal{M}_\theta(X)$ the scores produced by the model.

Remark 2.2.3. Independence ensures that *global normalization* (normalizing the whole distribution of dimension 2^k) and *local normalization* (normalizing the marginal distributions for each variable of dimension 2, resulting in a total dimension of k) are *equivalent*, which is the property exploited through the *sigmoid* layer to make computations tractable.

However both Semantic Projection (SP) and Semantic Regularization (SR) differ from traditional techniques (ie. *independent multilabel regression*) by allowing us to take into account **semantic constraints** over the output categories $\mathbf{Y}$ to enforce a specific probability distribution and/or loss functional.

In SR, defined in (Xu et al., 2018), semantic constraints are specified as a **propositional formula** α over the categories $(Y_1, ..., Y_k)$. We note $\mathbf{y} \models \alpha$ if instantiation $\mathbf{y}$ respects constraints encoded by α.

Example 2.2.4. If $\alpha = (Y_1 \vee \neg Y_2)$, then instantiations of the form $\mathbf{y} = (0, 1, ...)$ are **invalid**.

In (Deng et al., 2014), semantic constraints are encoded by means of a **HEX-graph** that stipulates which categories are subsumed to others and which categories are disjoint (or mutually exclusive).

A HEX-graph H can easily be associated to a propositional formula α_H enforcing the same semantic constraints, but most propositional formulas can't be encoded into a HEX-graph . This implies that the **expressiveness** of HEX-graphs is strictly inferior to that of propositional logic. In this paper we introduce Semantic Projection, which is a formal generalization of the HEX-graph methodology to arbitrary semantic constraints expressed in propositional logic.

To properly define the two techniques, we must define two basic operations on distributions : **normalization** and **semantic projection**.

Definition 2.2.5 (Normalization)**.** Transforming an unbounded, positive, and non-null distribution into a probability distribution.

Given a $\mathbf{E} : \{0,1\}^k \to \mathbb{R}^+$ such that $\mathsf{Z}(\mathbf{E}) > 0$, we will note :

$$\overline{\mathbf{E}} := \frac{\mathbf{E}}{\mathsf{Z}(\mathbf{E})}$$

with Z is the partition function.

Definition 2.2.6 (Semantic projection)**.** Projecting a distribution $\mathbf{E}$ on the space of valid instantiations according to α. We will note :

$$\mathbf{E}^\alpha := \mathbf{E}.\mathbb{1}[\mathbf{y} \models \alpha]$$

with $\mathbb{1}[z] := \begin{cases} 1 & \text{if z true} \\ 0 & \text{otherwise} \end{cases}$ the indicator function.

2.3. Semantic Regularization

Definition 2.3.1 (Semantic loss)**.** Given a set of categories $\mathbf{Y} = \{Y_1, ..., Y_k\}$, α a propositional formula over $\mathbf{Y}$ and a vector of scores $\mathbf{a}$, we define the **semantic loss** between α and $\mathbf{a}$ as:

$$\mathcal{L}^s(\alpha, \mathbf{a}) := -\log \sum_{\mathbf{y} \models \alpha} \prod_{i:\mathbf{y} \models Y_i} p_i \prod_{i:\mathbf{y} \models \neg Y_i} (1 - p_i)$$

$$= -\log(\mathsf{Z}[(\overline{\mathbf{E}}_{\mathcal{I}(\mathbf{a})})^\alpha])$$

$$= -\log(\mathsf{Z}(\mathbf{E}^\alpha_{\mathcal{I}(\mathbf{a})})) + \log(\mathsf{Z}(\mathbf{E}_{\mathcal{I}(\mathbf{a})}))$$

with $p_i := \frac{1}{1+e^{-a_i}}$.

Remark 2.3.2. The first formulation is the one introduced by the authors in (Xu et al., 2018), whereas the second and third are seen through the lens of our formalism. Their equivalence is shown in (Ledaguenel et al., 2022).
We can notice the expression of p_i which results from the *sigmoid function* applied to the scores a_i, and correspond to the local normalization step (see 2.2.3).
This loss term can be framed as performing a step of *positive/negative sampling* on valid/invalid instantiations : valid ones are *pushed up* and invalid ones are *pushed down* in the vocabulary of *energy-based learning*. Finally, an important remark is that this loss functional does not depend in any way on the true label. This property gives the ability to perform **semi-supervised** learning by backpropagating this loss on unlabeled samples.

Semantic Regularization simply consists of adding this term (with a regularization coefficient) to the standard independent multilabel loss functional defined in 2.2.2 :

$$\mathcal{L}^\alpha_{\mathcal{R}(\lambda)}(\theta, X, \mathbf{y}_{true}) := \mathcal{L}_{\mathcal{I}}(\theta, X, \mathbf{y}_{true}) + \lambda.\mathcal{L}^s(\alpha, \mathbf{a}_\theta)$$

This does not alter the probability distribution over instantiations enforced by the model, ie. the prediction during inference is still the mode of $\mathbf{P}_{\mathcal{I}(\mathbf{a})}$.

2.4. Semantic Projection

Definition 2.4.1 (Projected Independent Multilabel Distribution)**.** Given activation scores $\mathbf{a} := (a_i, ..., a_k) \in \mathbb{R}^k$ and a satisfiable propositional formula α, one can define the **projected independent multilabel probability distribution** as :

$$\mathbf{P}^\alpha_{\mathcal{I}(\mathbf{a})} = \overline{\mathbf{E}^\alpha_{\mathcal{I}(\mathbf{a})}}$$

Remark 2.4.2. Given a HEX-graph H and its equivalent α_H, $\mathbf{P}^{\alpha_H}_{\mathcal{I}(\mathbf{a})}$ is equivalent to the conditional probability $Pr(y|x)$ defined in (Deng et al., 2014) with $\mathbf{a} = f(x; w)$. The original definition and its equivalence with ours are shown in (Ledaguenel et al., 2022).

Semantic Projection computes the mode of the projected independent multilabel probability distribution during inference.

The loss functional computed during training, given a labeled sample $(X, \mathbf{y}_{true})$, is then the standard **negative log-likelihood** loss over the distribution $\mathbf{P}^\alpha_{\mathcal{I}(\mathbf{a})}$:

$$\mathcal{L}^\alpha_\Pi(\theta, X, \mathbf{y}_{true}) = -\log \mathbf{P}^\alpha_{\mathcal{I}(\mathbf{a}_\theta)}(\mathbf{y}_{true})$$
$$= -\log \mathbf{E}_{\mathcal{I}(\mathbf{a}_\theta)}(\mathbf{y}_{true}) + \log(\mathsf{Z}(\mathbf{E}^\alpha_{\mathcal{I}(\mathbf{a})}))$$

with $\mathbf{a}_\theta = \mathcal{M}_\theta(X)$

3. Observations

We present in this section several theoritical observations concerning Semantic Projection and Semantic Regularization.

First, it follows *by design* that predictions from SP are always consistent with the semantic constraints :

Proposition 3.0.1. *Given a parametrization θ, a satisfiable propositional formula α and a sample $X \in \mathbb{R}^d$:*

$$\arg\max_{\mathbf{y} \in \{0,1\}^k} \mathbf{P}^\alpha_{\mathcal{I}(\mathbf{a}_\theta)}(\mathbf{y}) \models \alpha$$

with $\mathbf{a}_\theta = \mathcal{M}_\theta(X)$

Besides, SP assigns higher probability than the standard independent multilabel probability distribution (enforced by both the standard technique and SR) on valid samples :

Proposition 3.0.2. *Given a parametrization θ, a propositional formula α and a valid sample $(X, \mathbf{y}) \in \mathbb{R}^d \times \{0, 1\}^k$ with $\mathbf{y} \models \alpha$:*

$$\mathbf{P}^\alpha_{\mathcal{I}(\mathbf{a}_\theta)}(\mathbf{y}) \geq \mathbf{P}_{\mathcal{I}(\mathbf{a}_\theta)}(\mathbf{y})$$

with $\mathbf{a}_\theta = \mathcal{M}_\theta(X)$

This implies that whatever loss functional is enforced by the system, for a given parametrization of the base neural network $\mathcal{M}_\theta$, inference with SP will always be more accurate than with SR.

Finally, we show in (Ledaguenel et al., 2022) that loss functionals of SP and SR are tightly linked :

Proposition 3.0.3.

$$\mathcal{L}^\alpha_\Pi = \mathcal{L}_{\mathcal{I}} - \mathcal{L}^s(\alpha, \mathbf{a}) = \mathcal{L}^\alpha_{\mathcal{R}(-1)}$$

This points out that to reach the same goal on almost identical architectures, the two techniques add opposite additional terms to the base loss functional. This counter-intuitive divergence is justified by the additional computations performed for inference after the neural network in the case of SP, which can be seen as a *parameter-free* extension of the architecture. However, this may indicate that a more optimal path lies between these two extremes, which led us to

Table 1. Properties of each technique

TECHNIQUE	PROB. DIST.	CONSISTENCY	LOSS	SEMI-SUPERVISED
STD	$\mathbf{P}_{\mathcal{I}(\mathbf{a})}$	✗	$\mathcal{L}_{\mathcal{I}}$	✗
SR	$\mathbf{P}_{\mathcal{I}(\mathbf{a})}$	✗	$\mathcal{L}_{\mathcal{I}} + \lambda.\mathcal{L}^s(\alpha, \mathbf{a})$	✓
SP	$\overline{\mathbf{E}^{\alpha}_{\mathcal{I}(\mathbf{a})}}$	✓	$\mathcal{L}_{\mathcal{I}} - \mathcal{L}^s(\alpha, \mathbf{a}_\theta)$	✗
RSP	$\overline{\mathbf{E}^{\alpha}_{\mathcal{I}(\mathbf{a})}}$	✓	$\mathcal{L}_{\mathcal{I}} - (1-\lambda).\mathcal{L}^s(\alpha, \mathbf{a}_\theta)$	✓

propose an hybridization of both techniques, called Regularized Semantic Projection (RSP), with the same probability distribution as SP and the loss functional:

$$\mathcal{L}^{\alpha}_{\Pi+\mathcal{R}(\lambda)} := \mathcal{L}_{\mathcal{I}} - (1-\lambda).\mathcal{L}^s(\alpha, \mathbf{a})$$

Remark 3.0.4. Table 1 summarizes key properties for each technique. The PROB. DIST. column specifies the mathematical expression of the probability distribution from which the mode will be predicted at inference time. CONSISTENCY indicates for which techniques the output will be consistent with the semantic constraints α *by design*. LOSS shows the loss functional optimized by each technique. Finally, SEMI-SUPERVISED points out which techniques can be used in a semi-supervised setting, where some of the input samples are unlabeled.

4. Efficient exact computations

As mentioned at the beginning of the paper, manipulating the projected independent multilabel unnormalized distribution $\mathbf{E}^{\alpha}_{\mathcal{I}(\mathbf{a})}$ in a naive way by operating on its individual values – hence performing computations on the space of instantiations – becomes quickly intractable as k grows (complexity in $\mathcal{O}(2^k)$).

Hence, to compute the aforementioned quantities (eg. $\arg\max \mathbf{P}^{\alpha}_{\mathcal{I}(\mathbf{a})}(\mathbf{y})$ or $\mathsf{Z}(\mathbf{E}^{\alpha}_{\mathcal{I}(\mathbf{a})})$ for instance), one needs to design a specific computational scheme, exploiting similar factorization properties about $\mathbf{E}^{\alpha}_{\mathcal{I}(\mathbf{a})}$ that we mentioned concerning $\mathbf{E}_{\mathcal{I}(\mathbf{a})}$ in 2.2.3.

(Deng et al., 2014) implements a custom computational scheme, derived from the **message passing algorithm on junction trees**, that takes advantage of both the factorization of $\mathbf{E}^{\alpha}_{\mathcal{I}(\mathbf{a})}$ into potentials and the sparsity of those potentials implied by the semantic constraints encoded in the HEX-graph. The **sum-product scheme** is utilized to compute the partition function of the distribution (and so the loss functional of the model), and the **max-product scheme** is utilized to compute the mode of the distribution (and so the prediction of the model).

Observations made earlier highlight that the semantic loss from (Xu et al., 2018) can be computed using the same key quantities (specifically $\mathsf{Z}(\mathbf{E}^{\alpha}_{\mathcal{I}(\mathbf{a})})$), which implies that the sum-product scheme can be utilized to perform Semantic Regularization. This results in an implementation very close to that of an **arithmetic circuit** used in (Xu et al., 2018).

(Deng et al., 2014) demonstrates that the complexity of the online computations for both algorithms (sum-product and max-product) is exponential in the max of two properties of the HEX-graph called the **tree-width** and the **maximum overlap**, but linear in the size of the graph (ie. the number k of categories considered). Therefore, for certain families of graphs, exact inference is tractable.

5. Future work

The algorithms developed in (Deng et al., 2014) can be generalized to process all propositional formulas. However, the complexity of the calculations can quickly explode. We are currently working on developing such a system and studying how complexity bounds (eg. tree-width and maximum overlap) evolve when applied to general propositional formulas.

Acknowledgments

This research work has been carried out under the leadership of the Technological Research Institute SystemX, and therefore granted with public funds within the scope of the French Program "Investissements d'Avenir".

References

Deng, J. et al. Large-scale object classification using label relation graphs, 2014. URL http://link.springer.com/10.1007/978-3-319-10590-1_4.

Ledaguenel, A. et al. Multi-category classification with semantic projection and semantic regularization. IRT SystemX Technical report, 2022.

Xu, J. et al. A semantic loss function for deep learning with symbolic knowledge. volume 12, pp. 8752–8760. International Machine Learning Society (IMLS), 2018.

INTERPRETABILITY OF DEEP LEARNING MODELS FOR VISUAL DEFECT DETECTION: A PRELIMINARY STUDY

Mehdi Elion, Sonia Tabti, Julien Budynek
FieldBox.ai, Bordeaux, France
{melion, stabti, jbudynek}@fieldbox.ai

ABSTRACT

How relevant are interpretability methods designed for deep learning models in the context of visual defect detection? Beyond their actual output, to what extent can these methods be used in a production environment? We study and evaluate interpretability methods for convolutional neural networks (CNN) and vision transformers (ViT) on image classification datasets designed for defect detection.

1 Introduction

Computer vision models based on deep learning have been increasingly successful in numerous fields, including industrial applications such as quality control and visual inspection. However, such models suffer from a lack of trust from end users as they are often considered as "black boxes" with not enough insights regarding their decision process. Hence, the need for model interpretability methods has concomitantly grown, not only to facilitate user trust and adoption, but also because such techniques can help detect model biases, validate the relevance of the decision processes underlying deep learning models and therefore make a first step towards AI certification.

1.1 Brief literature search on interpretability methods for computer vision deep learning models

To meet the need for model interpretability, several methods have been developed to provide users with visual insights pertaining to model predictions. Perturbation-based methods such as LIME [1], SHAP [2] or occlusion sensitivity [3] are model-agnostic and provide interesting insights, but they usually are computationally expensive, and perturbed data can be outside of the training distribution. On the other end, methods based on gradient back-propagation such as Integrated Gradients [4] or GradCAM (class activation maps) [5] are widely used and usually work well with CNNs. However, such methods are not necessarily adapted to alternate architectures such as Vision Transformers. Some methods, eg: attention-rollout [6], exploit the attention mechanism in ViT architectures in order to visualize which features they can extract from input images. However this method is not class-dependent, making it difficult to use for defect detection. Several methods have thus been developed specifically to better interpret ViT predictions [7], for instance gradient-attention-rollout, which is a class-dependant improvement of attention-rollout, or layer-wise relevance propagation (LRP), based on Deep Taylor Decomposition [8], which has proved to be a suitable choice for ViTs.

1.2 Problem Statement

Live defect detection applied to pictures of products taken at regular time intervals on a production line is a common industrial use case. In this study, we approach it through an image classification task. More precisely, we will focus on two formulations for this problem. Binary classification on the one hand, to detect whether or not there is a defect on an image of a product. Multi-class classification on the other hand, to categorize among several defect types which one appears on a product image. In this study, CNN and ViT classifiers will be compared in terms of classification metrics, computation efficiency and effectiveness of some of the most suitable interpretability frameworks for them.

2 Approach and implementation

In this section, we describe the above-mentioned deep learning models and their respective interpretability methods selected for this preliminary study. Then, we provide training and implementation details.

2.1 Models and interpretability methods description

The selected deep image classification networks for this study are VGG16 [9] (pretrained on ImageNet) for CNN classifiers, and ViT-small16 [10] (pretrained on ImageNet using DINO [11]) for ViT-based ones. The VGG16 architecture provides a good balance in terms of classification performance, computation efficiency and adaptability to many interpretability methods. In this study, the interpretabilty frameworks compared for the CNN model are Occlusion sensitivity and GradCAM. Occlusion sensitivity is a perturbation-based method that is model agnostic. It occludes iteratively image regions to assess how the CNN's confidence is affected. GradCAM is a gradient-based method. It uses the feature maps produced by the last convolutional layer to understand which regions of an image were relevant to the CNN.

Vision Transformers is the second type of classification model selected for this study as it has shown impressive results in multiple computer vision applications with high robustness to various types of perturbations (eg: image occlusion, domain shift) [12], which is valuable in an industrial context. The simplest interpretability method to exploit the ViTs' multi-head self-attention mechanism is the attention-rollout method which combines the attention maps from all the heads. As a result, this method is not class-specific, which can be an issue when one wants to inspect an interpretability map for a specific class to understand a model's decisions and errors. More formally, the attention-rollout boils down to the following equation:

$$\hat{\mathbf{A}}^{(b)} = I + \mathbb{E}_h \mathbf{A}^{(b)}, \quad \text{rollout} = \hat{\mathbf{A}}^{(1)} \cdot \hat{\mathbf{A}}^{(2)} \cdot ... \cdot \hat{\mathbf{A}}^{(B)} \tag{1}$$

where $\mathbf{A}^{(b)}$ is the attention map, $b = \{1, ..., B\}$ the transformer block index, $\mathbb{E}_h$ the mean across "heads" dimension and $(\cdot)$ the matrix multiplication.

The layer-wise relevance propagation (LRP) [7] provides class-specific maps. It computes relevance scores based on the Deep Taylor Decomposition principle for each attention head in each layer of a Transformer model. Then, it back-propagates these relevancy scores through the layers. In short, the output of the method, noted $\mathbf{C}$, is a combination of weighted attention relevance and is computed as follows:

$$\bar{\mathbf{A}}^{(b)} = I + \mathbb{E}_h(\nabla \mathbf{A}^{(b)} \odot R^{(n_b)})^+, \quad \mathbf{C} = \bar{\mathbf{A}}^{(1)} \cdot \bar{\mathbf{A}}^{(2)} \cdot ... \cdot \bar{\mathbf{A}}^{(B)} \tag{2}$$

where $\nabla \mathbf{A}^{(b)}$ is the gradient of the attention map, $R^{(n_b)}$ is the layer's relevance with respect to a target class (see [7] for more details), $\odot$ is the Hadamard product and $(.)^+$ denotes the positive part function.

2.2 Training procedure and implementation details

In order to obtain classifiers that are specifically trained on our target tasks, we use transfer learning and fine-tuning as it allows us to reach good results while using a reasonable amount of resources and time. More precisely, the transfer learning phase consists in removing the classifier head that was specific to the source task, replacing it with a new one that is specific to our target task and training it while leaving original backbone weights frozen. Then, during the fine-tuning phase, backbone weights are partially or totally unfrozen before launching a second training phase. Models are trained using gradient descent with Cross Entropy as a training criterion.

3 Experimental study

3.1 Datasets

The models and the interpretability methods implemented are evaluated on two image classification datasets. The first one is the casting defect dataset available on Kaggle [13], which contains a total of 7348 images of size 300×300, labeled as showing a defective (*def-front*) or non-defective (*ok-front*) product. It is divided into training and test sets. The training set contains 3758 defective images and 2875 non-defective images, while the test set contains 453 defective images and 262 non-defective images. The second one is the NEU-DET dataset [14], which contains 1800 images showing six types of surface defects of a hot-rolled steel strip, which are Crazing (Cr), Inclusion (In), Patches (Pa), Pitted Surface (PS), Rolled-in Scale (RS), and Scratches (Sc). Each type of sample has 300 grayscale images of size 200×200. Note that, for each dataset, images were resized to 224×224 before being used as model inputs, and we also keep 20% of the training set for validation in order to avoid overfitting.

3.2 Results

First, classification metrics listed on table 1 show that combining transfer learning and fine-tuning to train a classifier yields successful results, especially on such clean datasets where defects are easily identified. It should be noted that ViTs perform slightly better than VGG16 for most metrics and both datasets. Note that, the training procedure being very successful on NEU-DET, both models end-up misclassifying the same unique test image, hence the identical metrics.

	CNN			ViT		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
Casting defect	98.74	99.01	99.55	99.58	99.66	99.33
NEU_DET	99.72	99.73	99.72	99.72	99.73	99.72

Table 1: Classification metrics obtained on test sets. For the casting defect dataset, metrics were computed by considering defective products as positive cases. For NEU-DET, shown metrics were averaged over all classes.

Second, table 2 shows different relevant time measurements for each deep learning classifier and each interpretability method, namely: inference time and time necessary to generate interpretability heatmaps for one image. The hardware used is an NVIDIA GeForce RTX 2080 Ti GPU with 11 Go of RAM. It is clear that generating those heatmaps is computationally more intensive than a simple inference, by a factor of about three to ten, indifferently on both datasets. For the CNN classifier, GradCAM is twice as fast as occlusion sensitivity. For the ViT classifier, rollout and LRP have roughly the same performance. And, inference with ViT is twice as fast as inference with CNN.

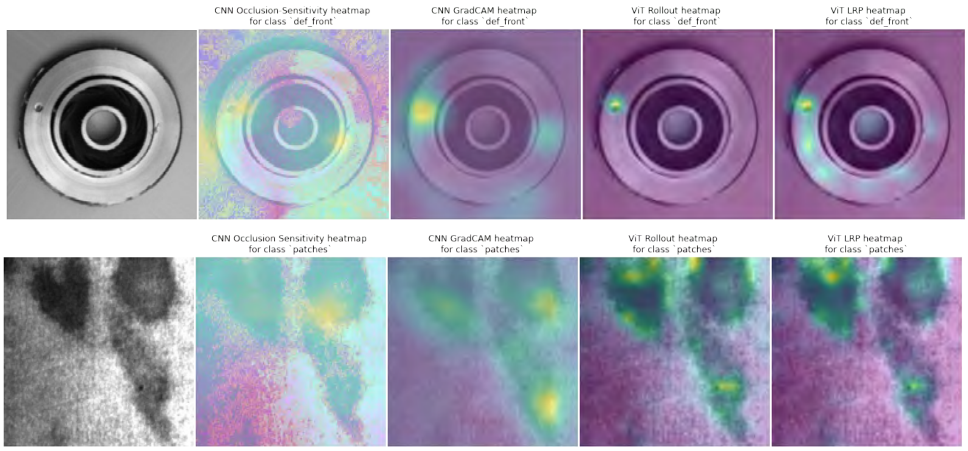


Figure 1: Example interpretability maps computed for defective products from the casting defect dataset (top line) and from the NEU-DET dataset (bottom line). The first image is the raw picture, then the columns show heatmaps for CNN Occlusion, CNN GradCAM, ViT attention-rollout and ViT LRP.

Finally, based on some visual examples, we evaluate the quality of interpretability maps output by the models. Figure 1 shows such examples on correctly classified samples. For class-specific methods, the interpretability map of the predicted class is shown. One can observe that occlusion sensitivity provides less accurate results than other methods. Indeed GradCAM, attention rollout and LRP heatmaps tend to highlight tighter areas in the image. However, we note that, on the casting defect example, GradCAM seems more "exhaustive" in terms of highlighted class-related areas, while LRP seems more selective than GradCAM. Attention rollout, on the other hand, by design, doesn't highlight class-related areas but salient elements in the image, which, in that case, can be defects themselves but in other cases can be harmful to model interpretability.

We also see the benefit of using interpretability to better understand classification errors. For instance, figure 2 shows an example on the casting dataset where the classifier seems to wrongly consider a product as defective because of particular lighting conditions. An example is shown as well on the NEU-DET dataset where an inclusion defect is confused with a scratch. It is understandable since some scratches are similar to inclusions.

4 Conclusion

To conclude this preliminary study, combining the measurements in tables 1 and 2 with the qualitative analysis of the interpretability maps, we recommend practitioners the use of GradCAM over occlusion sensitivity for CNNs and LRP over attention-rollout for ViTs. However, if the computational efficiency required once a defect detection solution is deployed on a production line is strong, one might consider using ViTs and an interpretability method as an offline tool

Figure 2: Example interpretability maps computed on misclassified images, first from the casting defect dataset (left side: raw picture, then heatmaps for CNN Occlusion and CNN GradCAM), and from the NEU-DET dataset (right side: raw picture, then heatmap from ViT LRP).

	CNN			ViT		
	Forward-pass	Occlusion	GradCAM	Inference	Rollout	LRP
Casting defect	51.0 ± 1.6	160.0 ± 9.4	89.2 ± 4.7	24.7 ± 8.2	291.2 ± 27.3	266.4 ± 26.1
NEU_DET	58.2 ± 13.0	181.0 ± 44.0	92.3 ± 7.8	25.6 ± 5.0	249.2 ± 23.6	246.3 ± 17.2

Table 2: Computation times (in ms). Provided numbers correspond to mean and standard deviation of elapsed time during single-image operations: forward-pass or computation of intepretability maps.

to monitor the model's predictions behavior. Future works will include a deeper comparative study. For instance, other sizes of ViTs could be used and other interpretability methods tested. Also, a quantitative analysis of the relevance of interpretability maps using datasets with segmentation maps could be done.

References

[1] M.T Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[2] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[3] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[4] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks, 2017.

[5] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, oct 2019.

[6] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*, 2020.

[7] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791, June 2021.

[8] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognition*, 65:211–222, May 2017.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021.

[12] Muhammad M Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.

[13] Ravirajsinh Dabhi. Casting product image data for quality inspection, 2020.

[14] Kechen Song and Yunhui Yan. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science*, 285:858–864, November 2013.

Deep Neural Networks Abstraction using An Interval Weights Based Approach

Fateh Boudardara [1], Abderraouf Boussif[1], Mohamed Ghazel[2], and Pierre-Jean Meyer[2]

[1]Technological Research Institute Railenium, Valenciennes, France
[2]Univ Gustave Eiffel, COSYS-ESTAS, Villeneuve d'Ascq, France

Abstract

In this work, we present a Neural Network (NN) abstraction approach to reduce the state-space (number of nodes) of NN towards solving the non-scalability of NN formal verification approaches. The main idea consists in merging neurons on the NN layers in order to build an abstract model that over-approximates the original one. Concretely, the outgoing weights of the abstract network are computed as the sum of the absolute value of the weights on the original one, while the incoming weights are intervals determined based on the signs of the outgoing and the incoming weights of the original model.

1 Introduction

Due to the tremendous success of deep neural networks (DNNs), they are increasingly deployed in safety-critical systems, such as autonomous cars and trains. However, these systems must meet some specific safety requirements before their deployment. Therefore, many concerns about the safety of DNNs have been raised recently. In fact, recent studies demonstrated the vulnerability of DNNs [1], thus the domain of neural networks verification are becoming more popular and attractive. Several formal verification methods are adjusted and applied to check some properties on DNNs, such as safety and robustness. Originally, the verification problem of DNNs was transformed to an optimization problem and solved using Mixed-Integer Linear Programming and SAT/SMT solvers [2, 3]. Many other methods were developed for instance abstract interpretation [4], reachability [5] and others.

Unfortunately, the developed techniques cannot scale to verify large models because of the high complexity of DNNs. Model reduction methods that are considered as abstraction methods and consist of reducing the size of the model while preserving some relevant behaviors [6, 7, 8], are seen as a promising remedy to the problem of scalability of the existing NN verification methods. A model reduction approach ensures that whenever the property holds on the reduced model, it must hold on the original. In this paper we present a method that is based on converting the original NN to an interval NN (INN). The reduced model is constructed by taking the interval hull of the incoming weights and the sum of the outgoing weights in such a way that the outputs of the original network are always included in those of the abstract one. The presented method supports both *Tanh*-NN and *Relu*-NN. A succinct presentation of the proposed approach and the preliminary obtained results are presented here below; further details about the approach and the experiments are presented in [9].

A neural network is a sequence of connected layers. The first layer is the input layer, followed by one or more hidden layers and an output layer. Each neuron s_{ij} in S_i [1] of a hidden layer receives data from its predecessor layer, calculate its activated value using Equation 1, and forward the result to its successor layer.

$$v(s_{ij}) = \alpha(\sum_{s \in S_{i-1}} w(s, s_{ij}) \times v(s) + b_{s_{ij}}) \quad (1)$$

In Equation 1, $w(s, s_{ij})$ is the weight of the edge connecting $s \in S_{i-1}$ to $s_{ij} \in S_i$, $b_{s_{ij}}$ is the bias of the node s_{ij}, and α is a predefined activation function. Our method supports *Relu*-NN ($\alpha(x) = Relu(x) = max(0, x)$) and *Tanh*-NN ($\alpha(x) = Tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$).

(a) The original network

(b) The abstract network

Figure 1: An example explaining the main idea of the proposed approach.

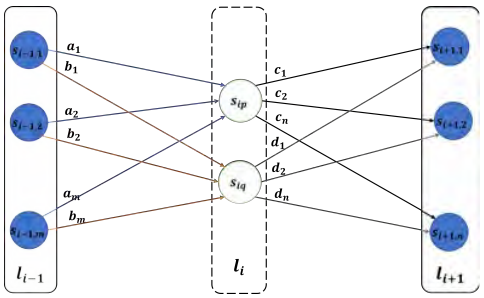
2 Proposed model reduction method

Model reduction for NNs, as a sub-category of NN abstraction, is a concept of reducing the size of NNs by merging some neurons while guaranteeing that the original model N satisfies the property P whenever this property is satisfied by the abstract model $\bar{N}$, i.e., $\bar{N} \models P \implies N \models P$.

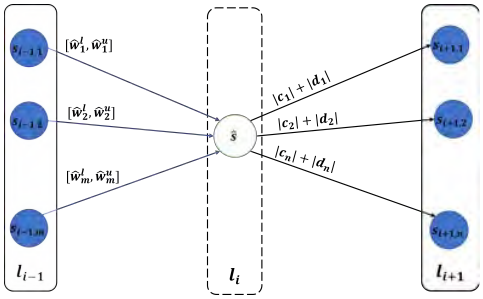
The broad idea of our method is to merge neurons of hidden (intermediate) layers and compute the incoming weights of the abstract node as the convex interval hull of its incoming weights before abstraction multiplied by the sign of its outgoing weights. On the other hand, the outgoing weights of the abstract node are computed as the sum of the absolute value of its corresponding outgoing weights on the original network. Figure 1 illustrates the main idea approach. Notice that *sign* is a function defined as follow: $sign : \mathbb{R} \to \{-1, 1\}$

$$sign(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

2.1 Model reduction for NN with *Tanh* activation function



(a) A sub neural network containing three hidden layers, we want to merge the two nodes of layer l_i.



(b) The sub neural network after abstraction. Here $\hat{w}_k^l, \hat{w}_k^u : 1 \leq k \leq m$ are the weights calculated using formula (3).

Figure 2: An illustration of our abstraction method applied on a hidden layer l_i. The model on the right is the abstraction of the one on the left, where the node $\hat{s}$ is obtained upon merging s_{ip} and s_{iq}.

For simplicity and without lose of generality, let consider the network in Figure 2a, and assume that we want to merge the two nodes s_{ip} and s_{ip}. The obtained abstract network is presented in Figure 2b, where $\hat{s}$ is abstract node after merging s_{ip} and s_{ip}. The incoming weights of $\hat{s}$ have the form of intervals and they are calculated as follows:

$$\begin{cases} \hat{w}_k^l = \min\limits_{1 \leq j \leq n} \{sign(c_j)\, a_k, sign(d_j)\, b_k\} \\ \hat{w}_k^u = \max\limits_{1 \leq j \leq n} \{sign(c_j)\, a_k, sign(d_j)\, b_k\} \end{cases} \tag{3}$$

and its outgoing weights are the sum of the absolute value of the corresponding outgoing weights of s_{ip} and s_{ip}. Algorithm 1 summarizes the essential steps of the model reduction for neural networks with *Tanh* (*Tanh*-NN).

[1] S_i is the set of neurons of layer l_i, and $s_{ij} \in S_i$ is the j^{th} neuron of S_i

Algorithm 1 Proposed model reduction procedure for *Tanh*-NN

1: create a node $\hat{s}$
2: select s_{ip} and s_{iq}
3: calculate the incoming weights to $\hat{s}$ using Equation 3
4: calculate the outgoing weights: $\hat{w}(\hat{s}, s_{i+1,j}) = |c_j| + |d_j|$
5: replace s_{ip} and s_{iq} with $\hat{s}$

2.2 Model reduction for NN with *Relu* activation function

The *Relu* function is a piece-wise linear function, it eliminates the negative values (set them to zero) and returns only positive values. This particularity prevents the application of the model reduction method present in Algorithm 1 on *Relu*-NNs. Algorithm 2 depicts the update of Algorithm 1 to support *Relu*-NNs, where c_j^* (resp. d_j^*) presented in line 4 in Algorithm 2 is the outgoing weight c_j (resp. d_j) such that $sign(c_j*)\, a_k = \min\limits_{1\leq j\leq n} \{sign(c_j)\, a_k\}$ (resp. $sign(d_j*)\, b_k = \min\limits_{1\leq j\leq n} \{sign(d_j)\, b_k\}$).

Algorithm 2 Proposed model reduction procedure for *Relu*-NN

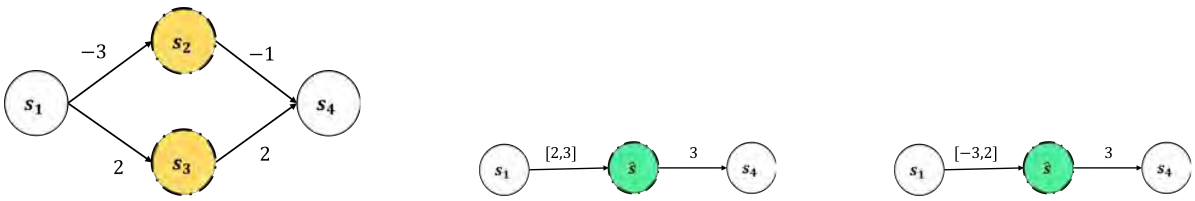
1: create a node $\hat{s}$
2: select s_{ip} and s_{iq}
3: **for** outgoing weight of s_{ip} and s_{iq} **do**
4: calculate c_j^* and d_j^*
5: **end for**
6: calculate the incoming weights to $\hat{s}$ using Algorithm 3
7: calculate the outgoing weights: $\hat{w}(\hat{s}, s_{i+1,j}) = |c_j| + |d_j|$
8: replace s_{ip} and s_{iq} by $\hat{s}$

Algorithm 3 Computation of the incoming weights for *Relu*-NN

1: **if** $sign(a_k) \neq sign(c_j^*)$ or $sign(b_k) \neq sign(d_j^*)$ **then**
2: Use Equation 3
3: **else if** $sign(a_k) = sign(c_j^*)$ and $sign(b_k) = sign(d_j^*)$ **then**
4: Use Equation 4
5: **end if**

$$\begin{cases} \hat{w}_k^l = \min\{a_k, b_k\} \\ \hat{w}_k^u = \max\limits_{1\leq j\leq n} \{sign(c_j)\, a_k, sign(d_j)\, b_k\} \end{cases} \tag{4}$$

Figure 3 shows an example of merging two neurons s_2 and s_3 of the original network presented in Figure 3a. In the case the network uses the *Tanh* activation function, its abstract network is the network in Figure 3b obtained by applying Algorithm 1. If we suppose that the original network (Figure 3a) is a *Relu*-NN, Algorithm 2 is applied and the corresponding abstract network is presented in Figure 3c.



(a) The original network

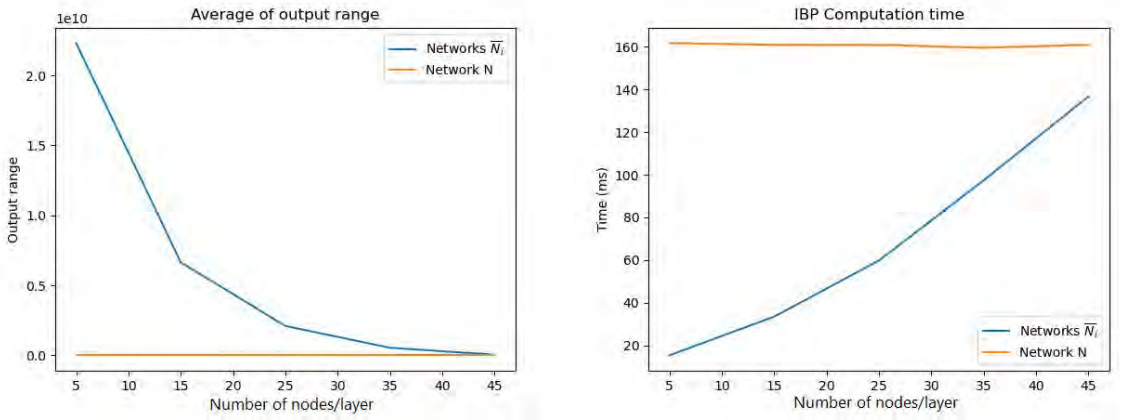
(b) The abstract model for *Tanh*-NN

(c) The abstract model for *Relu*-NN

Figure 3: An example of the abstraction method applied on two neurons s_2 and s_3 of a hidden layer l_i.

We implemented Algorithm 1 and 2 as a Python framework and we conducted a series of experiments on the ACAS Xu benchmark [2]. For the output range computation we considered the property ϕ_5 as defined in [2]. We

examined the performance of our approach by varying the size of layers of the abstract network $(5, 15, 25, 35, 45)$. The selection of neurons to be merged is performed randomly, and for each abstract network we calculated the abstraction time, the output range using Interval Bound Propagation (IBP) algorithm [5] and IBP computation time. We compared the average output range and the IBP computation time over 50 random runs for each abstract model with the results obtained on the original model as shown in Figure 4a and 4b.



(a) Output range

(b) IBP computation time

Figure 4: Comparison of the output range and the IBP computation time on the original network and different abstract networks.

The obtained results showed that there is a trade-off between the total number of abstract nodes and the precision of the obtained abstract model. Having more nodes in the abstract network increases its precision, and also its IBP computation time. Notice that IBP is one of the fastest verification methods, and yet its computation time is significantly higher compared to the abstraction time of our approach (its is not provided to shortage of space and it will be presented on the poster).

References

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint*, 2013.

[2] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International conference on computer aided verification*, pages 97–117. Springer, 2017.

[3] Souradeep Dutta, Susmit Jha, Sriram Sankaranarayanan, and Ashish Tiwari. Output range analysis for deep feedforward neural networks. In *Proc. 10th NASA Formal Methods*, pages 121–138, 2018.

[4] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2018.

[5] Weiming Xiang, Hoang-Dung Tran, Xiaodong Yang, and Taylor T Johnson. Reachable set estimation for neural network control systems: A simulation-guided approach. *IEEE Transactions on Neural Networks and Learning Systems*, 32(5):1821–1830, 2020.

[6] Pavithra Prabhakar and Zahra Rahimi Afzal. Abstraction based output range analysis for neural networks. *arXiv preprint*, 2020.

[7] Yizhak Yisrael Elboher, Justin Gottschlich, and Guy Katz. An abstraction-based framework for neural network verification. In *International Conference on Computer Aided Verification*, pages 43–65. Springer, 2020.

[8] Pranav Ashok, Vahid Hashemi, Jan Křetínskỳ, and Stefanie Mohr. Deepabstract: Neural network abstraction for accelerating verification. In *International Symposium on Automated Technology for Verification and Analysis*, pages 92–107. Springer, 2020.

[9] Fateh Boudardara, Abderraouf Boussif, Pierre-Jean Meyer, and Mohamed Ghazel. Interval weight-based abstraction for neural network verification. In *Fifth International Workshop on Artificial Intelligence Safety Engineering (Accepted)*, pages 1–16, 2022.

Detecting Outliers in Few-Shot-Learning Support Sets

1. Authors

- **Tarek Ayed**, Data Scientist at Sicara
  tarek.ayed@sicara.com
- **Etienne Bennequin**, PhD Candidate at Sicara and CentraleSupélec
  etienneb@sicara.com
- **Antoine Toubhans**, Head of Science at Sicara
  antoinet@sicara.com

Abstract

Few-shot-learning is a machine learning research area where a model infers a task it has not been trained on. In Computer Vision classification, this means classifying samples among classes the model has not seen during training, and given only a few examples of each class, typically less than 5. The quality of these few labeled samples (i.e. the support set) is highly impactful on the performance of the model. In particular, the presence of outliers in the support set quickly degrades the model's ability to learn the classification task [1] [2]. We propose a method to detect these outliers using KNN and Isolation Forest and an already trained FSL backbone. We achieve more than 91.3% AUROC on CUB [3] and 89.3% on MiniImageNet [4] for 5-shot tasks. We also show that this method scales reliably to 10-shot and 500-shot setups and that these results are highly reliant on the quality of the used backbone.

Introduction

In machine learning and deep learning, classification models are classically trained to predict one of a number of predetermined classes. Few-shot-learning (FSL) [5] is a set of techniques aiming at training models capable of classifying classes they have never encountered in the training set. The prediction is only based on a few examples of each class (the support set). This is highly useful in a number of use-cases where there is not sufficient time and/or sufficient training data to retrain a model whenever a new class appears, such as character recognition, voice recognition, object classification, etc.

However, in real-world setups, the support set is frequently altered and updated by humans and can thus contain errors. One of those errors is to assign a sample to the wrong class. This is called an outlier. The aim of this work will be to explore the idea of detecting these outliers in the support set, in order to prevent them from causing a drop in performance.

Robust Few-Shot-Learning

Few-shot models have to rely on a few labeled examples to generalize and classify unknown images. This means that the quality of the few examples given to the model in an FSL task can have a high influence on its performance. This has been documented in [1] and [2] where it is clear that the few-shot learner's performance falls linearly as outliers are added to its support set.

This deterioration in FSL models gave rise to *Robust Few-Shot Learning* (RFSL) which is a research problem introduced by [2]. The goal is to build models that are resilient to the presence of outliers in their support sets.

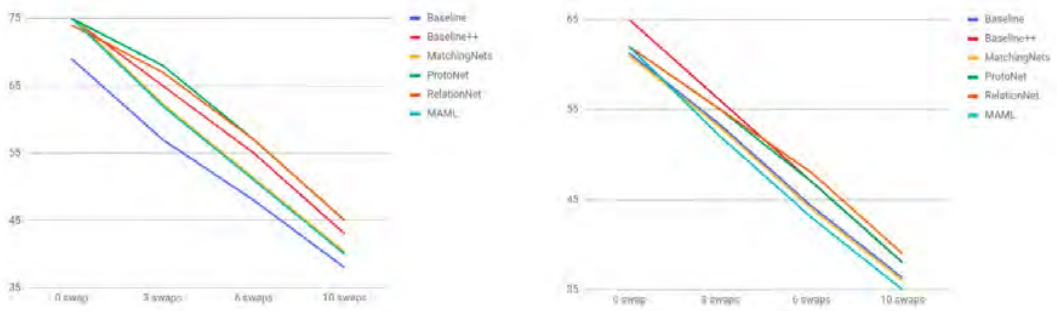


Figure 1. Evolution of a few-shot-learning model classification test accuracy when adding outliers to the support set. CU-Birds [3] dataset on the left, MiniImageNet [4] [6] on the right. Figure from [1].

The authors make a distinction between Representation Outliers (ROs) and Label Outliers (LOs). Images of the first type are correctly labeled, but are noisy samples, whereas those of the second type are those that have been wrongly labeled. In our work, we chose to only focus on Label Outliers.

The approach used by [2] is thus to build a model that is robust to the presence of outliers in the support set, rather than find methods to remove these outliers. We take the complementary approach that looks for ways to detect and remove these outliers from the support set.

Method

In this work, our proposed method is to apply outlier detection methods, such as isolation forest [7] or K-nearest neighbors (KNN) [8], in the embedding space, in order to accurately predict which elements in a support set class are outliers. An outline of this proposed method is described in figure 2.

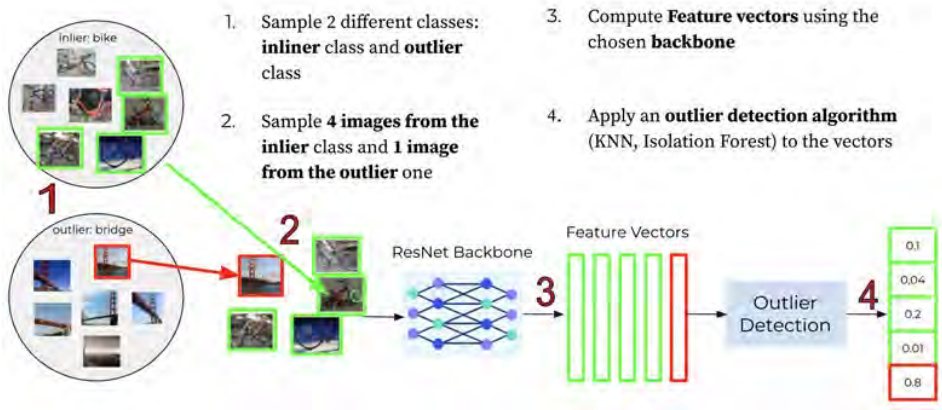


Figure 2. The general outline of the proposed method from image sampling to outlier detection predictions.

In other words, we only apply outlier detection to one class at a time. Features of each image of a class are computed using a trained backbone. Then an outlier detection method is applied to the feature vectors of the images of the class and each image is given an *outlier score*. Finally, by thresholding this prediction, we can output the prediction as to whether an image is an outlier or not.

Experiments

Datasets used We chose to rely on three different datasets, that are usually used in few-shot-learning experiments, as they all have more than 100 classes.

1. **CU-Bids** [3] is a dataset of birds photographs containing 6,033 images split across 200 classes.
2. **CIFAR-100** [9] is a dataset of 60,000 32 by 32 colour images split across 100 classes containing 600 images each.
3. **MiniImageNet** [6] is a subset of ImageNet [6] containing 100 classes and 600 84x84 colour images in each class.

Outlier generation procedure The way we generate artificial outliers in our datasets is to sample a class of $k-1$ samples of the same class and 1 image from another class, as shown in figure 2. Thus a sample contains a batch of images from the same class, along with a mislabelled image that comes from another class of the dataset.

Both inlier and outlier classes are drawn randomly. The $k-1$ inliers and the outlier are also sampled randomly from their respective classes.

Results Our main results are presented in table 1 for CUB [3], and in table 2 for Mini-ImageNet [4].

Outlier Detection Method	**Backbone**	**5-shot**		**10-shot**	
		AUROC	Pr. at 80% R.	AUROC	Pr. at 80% R.
KNN	ResNet18	89.8	**66.0**	94.0	**60.9**
Isolation Forest	(*Pre-trained*)	**91.3**	58.7	**95.1**	53.3
KNN	ResNet50	85.8	53.3	93.2	58.0
Isolation Forest	(*Pre-trained*)	89.9	58.2	**95.1**	54.3
KNN	ResNet18	81.7	46.8	88.2	33.4
Isolation Forest		89.0	55.4	92.5	42.7
KNN	ResNet50	80.5	41.6	92.3	49.4
Isolation Forest		86.3	50.0	93.0	45.8

Table 1. Outlier Detection performance in terms of AUROC and Precision at 80% Recall of KNN and Isolation Forest, applied to CU-Birds in 5-shot and 10-shot setups (with 1 outlier in each class for both), and using different ResNet backbones, some of which were pre-trained on ImageNet, then fine-tuned for CUB. The others have been directly trained on CUB from scratch.

Results on Mini-ImageNet

Outlier Detection Method	**Backbone**	**5-shot**		**10-shot**	
		AUROC	Pr. at 80% R.	AUROC	Pr. at 80% R.
KNN	ResNet18	80.5	41.0	86.5	27.0
Isolation Forest		**89.3**	**54.3**	89.7	35.4
KNN	ResNet50	79.5	40.0	86.7	35.7
Isolation Forest		85.0	44.3	**90.4**	**35.9**

Table 2. Outlier Detection performance in terms of AUROC and Precision at 80% Recall of KNN and Isolation Forest, applied to Mini-ImageNet in 5-shot and 10-shot setups (with 1 outlier in each class for both).

2. Conclusions

In this work, we show that classical outlier detection algorithms such as KNN and Isolation Forest can successfully be applied to FSL support sets to detect mislabeled samples, and this can be done accurately enough so that it is useful in real-world setups.

Our method generally achieves $> 90\%$ AUROC and $> 50\%$ Precision at 80% recall on three datasets (CU-Birds, CIFAR-100 and Mini-ImageNet) and on 5-shot and 10-shot settings. We also show that these methods can also scale to 500-shot samples and that the backbone used to compute feature vectors has a great influence on the final performance.

It could be used as an iterative way of mining potential outliers in the support sets and having them checked by a human operator so that the quality of the task's support sets gradually increases.

References

[1] E. Bennequin, "Meta-learning algorithms for few-shot computer vision," 2019. [Online]. Available: https://arxiv.org/abs/1909.13579

[2] J. Lu, S. Jin, J. Liang, and C. Zhang, "Robust few-shot learning for user-provided data," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[3] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep., 2010.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009.

[5] M. Fink, "Object classification from a single example utilizing class relevance metrics," in *Advances in Neural Information Processing Systems*, L. Saul, Y. Weiss, and L. Bottou, Eds. MIT Press, 2005.

[6] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," 2017. [Online]. Available: https://arxiv.org/abs/1606.04080

[7] F. T. Liu, K. M. Ting, and Z. hua Zhou, "Isolation forest," in *In ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining. IEEE Computer Society*, 2008.

[8] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *SIGMOD Rec.*, 2000.

[9] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

CASUAL: Case-based Reasoning using Unsupervised Part Learning

Romain Xu-Darme[1,2], Georges Quénot[2], Zakaria Chihani[1], Marie-Christine Rousset[2]
[1]*Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France*
romain.xu-darme,zakaria.chihani(at)cea.fr
[2]*Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France*
georges.quenot,marie-christine.rousset(at)imag.fr

***Abstract*—In this contribution, we present CASUAL (Case-based Reasoning using Unsupervised Part Learning), a preliminary work on an interpretable model for fine-grained visual classification. It uses an unsupervised part learning algorithm to perform a semantic realignment of the latent representation of images in order to 1) simplify the process of learning class prototypes during training 2) reduce the number of comparison between latent vectors during inference.**

With the growing need for trust in decision-making processes using machine-learning, the field of explainable artificial intelligence (XAI) has been rapidly increasing. Recent years have notably seen the development of a plethora of methods covering a wide spectrum of usages, from *post-hoc* explanations of opaque systems to models that are inherently transparent (leading to the concept of *interpretability by design*). In the latter case, the model uses semantic features in an intelligible manner to produce the decision, hence ensuring that the user can build an accurate mental picture reflecting the model behavior [1]. In the particular case of computer vision tasks (*e.g.,* image classification), models using case-based reasoning [2]–[6] now show accuracy results on par with their non-interpretable counterparts. The process itself consists in solving new instances of a given problem using comparisons with previously encountered examples, and is comparable to the form of cognitive operation applied by human beings in domains such as case law. In practice, during learning, the objective of the model is to extract a set of references points (*a.k.a., prototypes*) from the training set. These prototypes are used as comparison points with the test instance during inference to compute the decision. In the particular case of image classification for instance, prototypes are usually images (or parts of images) from the training set - chosen as representative of their respective class - and the process of classification is based on the visual similarity between the image under test and the set of all class representatives. However, defining a relevant visual similarity metric between images directly remains a challenge, due to the lack of semantic value of the individual pixels: for example, two images differing only by a small geometric transformation (*e.g.,* rotation, shift) would not be considered as "similar" using an euclidean distance between pixel values. Instead, modern approaches (see Fig. 1) perform image comparison in the latent space of a deep Convolutional Neural Network (CNN), using abstract representations that are, hopefully, for robust to variations (with works like [6] even quantifying the similarity in terms on hue, contrast, shape, texture, etc). However, in practice, architectures such as ProtoPNet [3], ProtoPShare [4] or ProtoTree [5] look for discriminative clusters without knowing beforehand which points are actually relevant for the downstream task (classification). In particular, non discriminative points (*e.g.,* corresponding to the background) can clutter the latent space and may have a negative impact on the learning process. This may explain why architectures such as [3], [4] use the object bounding boxes, when available. Moreover, during inference, every single point of the latent representation of the test image is compared to all prototypes, leading to a high inference time (*e.g.,* 196k comparisons for a 14×14 latent representation and 1000 prototypes).

In this work, we propose a new model, called CASUAL (Case-based reasoning using Unsupervised Part Learning), which performs a preliminary *semantic realignment* of the features vectors before prototype extraction, in the specific context of fine-grained recognition. Instead of finding clusters inside of a convolutional space shared among all points of the latent representation of the image, we first identify relevant points - corresponding to certain parts of the object - within this representation, using the unsupervised algorithm presented in [7] (for which a patent is currently pending). As illustrated in Fig. 2, this semantic realignment of feature vectors:

1) removes feature vectors corresponding to the background of the image, and the need for bounding boxes.
2) maps each training image into multiple latent spaces, reducing the complexity of the clustering process.

As a consequence, our CASUAL model learns one latent space per part, and *part prototypes* inside of these latent spaces. In practice, as shown in Fig. 2, this also simplifies the inference process by reducing the number of comparisons between the test image and the prototypes by an order of magnitude (prototype comparisons take place within each separated part latent space).

In order to validate our approach, in the coming months we aim at applying this architecture to the Caltech-UCSD Birds 200 dataset [8] and to provide accuracy results on the corresponding fine-grained classification task.

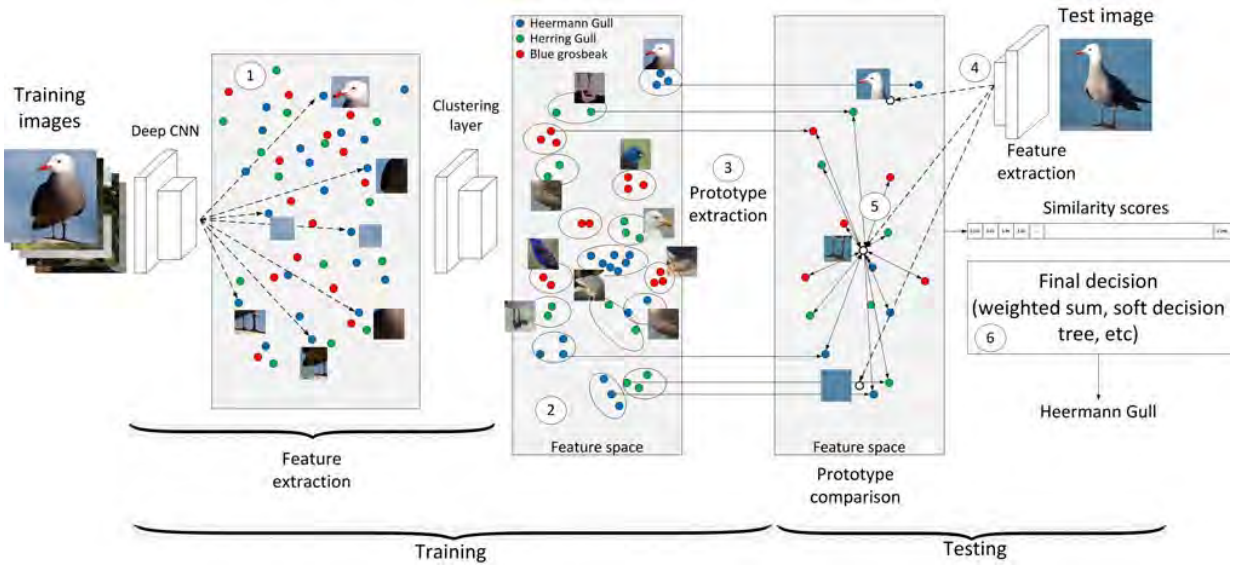


Fig. 1: State-of-art case-based reasoning models. 1) Each training image is converted into a set of points (feature vectors) in the latent space of a deep CNN, each point corresponding to a region of the image. 2) During training, the system learns to project these points (using additional convolutional layers) such that there exist discriminative clusters corresponding to the different categories of objects. 3) Prototypes are extracted as parts of training images closest to each cluster centroid. 4) During inference, all points (white) of the latent representation of the test image are compared (5) to all prototypes and similarity scores aggregated before the final decision (6).

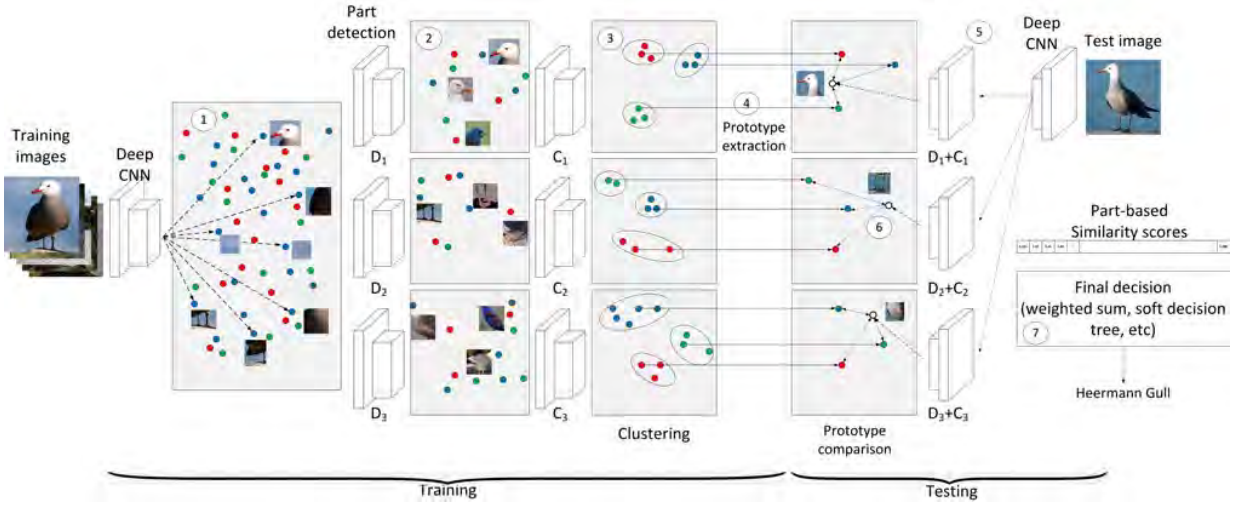


Fig. 2: Our proposal for the CASUAL architecture. 1) Similar to related works, each training image is first converted into a set of points in the latent space of a deep CNN. 2) Then, using the algorithm described in [7], we select points (layers D_i) corresponding to the same part of the object (*e.g.,* "head","legs","belly"), before applying clustering layers C_i *per part* (3), leading to one latent space per part. 4) Part-prototypes are extracted. 5) During inference, we first project the test image into a set of points (one per latent space corresponding to a given part). 6) Each part point (white) of the latent representation of the test image is compared to all corresponding part-prototypes and similarity scores aggregated before the final decision (7).

REFERENCES

[1] Z. C. Lipton, "The mythos of model interpretability," *Communications of the ACM*, vol. 61, pp. 36 – 43, 2018.

[2] O. Li, H. Liu, C. Chen, and C. Rudin, "Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network that Explains Its Predictions," *arXiv:1710.04806 [cs, stat]*, Nov. 2017, arXiv: 1710.04806. [Online]. Available: http://arxiv.org/abs/1710.04806

[3] C. Chen, O. Li, C. Tao, A. J. Barnett, J. Su, and C. Rudin, "*This* looks like *That*: Deep learning for interpretable image recognition," *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, p. 8930–8941, 2019.

[4] D. Rymarczyk, L. Struski, J. Tabor, and B. Zieliński, "ProtoPShare: Prototype Sharing for Interpretable Image Classification and Similarity Discovery," *[preprint] arXiv:2011.14340 [cs]*, Nov. 2020. [Online]. Available: http://arxiv.org/abs/2011.14340

[5] M. Nauta, R. van Bree, and C. Seifert, "Neural Prototype Trees for Interpretable Fine-grained Image Recognition," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14 928–14 938, 2021.

[6] M. Nauta, A. Jutte, J. Provoost, and C. Seifert, "This Looks Like That, Because ... Explaining Prototypes for Interpretable Image Recognition," *arXiv:2011.02863 [cs]*, Mar. 2021, arXiv: 2011.02863. [Online]. Available: http://arxiv.org/abs/2011.02863

[7] R. Xu-Darme, G. Quénot, Z. Chihani, and M.-C. Rousset, "PARTICUL: Part Identification with Confidence measure using Unsupervised Learning," Jun. 2022, accepted at XAIE: 2nd Workshop on Explainable and Ethical AI – ICPR 2022. [Online]. Available: https://hal-cea.archives-ouvertes.fr/cea-03703962

[8] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona, "Caltech-UCSD Birds 200," California Institute of Technology, Tech. Rep. CNS-TR-2010-001, 2010.

Village Bringing trust from ODD to Data (posters)

Introduction to the themes of the village

Flora Dellinger, Morayo Adedjouma

For a safe integration of AI components into critical systems, an evolution of classical engineering processes is required. In particular, new concepts and new activities need to be integrated, such as the ODD (Operational Design Domain) and the data lifecycle.

First, the ODD is a new concept, coming from the automotive sector, describing the "Operating conditions under which a given automated driving system (ADS) is designed to properly operate, including but not limited to environmental, geographical, and time-of-day restrictions, roadway types, speed range etc." (SAE J3016, 2021). Its objective is to define the limits where the ADS is valid and thus confine the scope of the safety case, as well as the validation.

On the other hand, AI components are trained and evaluated on datasets. Using relevant data of quality is necessary to develop and validate AI models that satisfy use case requirements and expected system performances. This is particularly true for critical systems that puts an emphasis on safety and avoidance of failures. However, obtaining a proper dataset for a use case is challenging in itself. Datasets need to be statistically representative of the operational domain, meaning the situations encountered by the system in operating conditions. The system performance is also related to its responsiveness to any complex and environmental situations the systems may face, at the frontier of the operational domain and beyond. Then, one should also validate the system robustness against such situations.

We find then here a strong interest to use an ODD as an input to specify which data to acquire for training, testing and validating Machine Learning models. However, the ODD concept was primarily defined for safe-by-design purpose. It is intended to be refined to reach its final maturity level through the overall system development cycle for defining a satisfactory operating system. While its usage within the safety, design and V&V activities was well elaborated in the literature (SAE J3016, 2021), this declination for a data and machine learning perspectives, while of an obvious interest, is not straightforward.

Finally, datasets are often seen as fixed databases, and expected to be perfect. However, in reality, datasets are constantly evolving, containing several batches of data acquired at different moments and under various conditions. They also contain some imprecisions, annotation mistakes or anomalies, and need to be filtered and preprocessed before being used for training an AI model.

This village provides a focus on the interaction between ODD and data lifecycle. First, an ODD Engineering process (Bohn et al.) is introduced, as well as some perspectives for ODD usage in a data and ML monitoring (Adedjouma et al.). Then the declination from ODD to Dataset specification is addressed with an outline of Trustworthiness aspects for Data Engineering in AI (Langlois et al.), associated with the first experiments of a PhD thesis (LeCoz et al.). We provide an overview of methodologies to build trustworthy datasets for AI. Some practical solutions are featured like a Data platform (Braud et al.) and the use of synthetic datasets (Leroy et al.). Finally, some research approaches tackle the challenges of leveraging unlabeled data for time series (Antoni et al., Ngole Mboula) and images (Poka Toukam et al.).

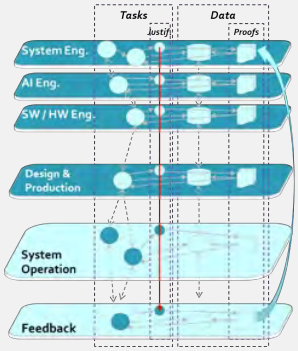
Bringing trust from ODD to Data

Towards Trustworthiness for Data Engineering in AI

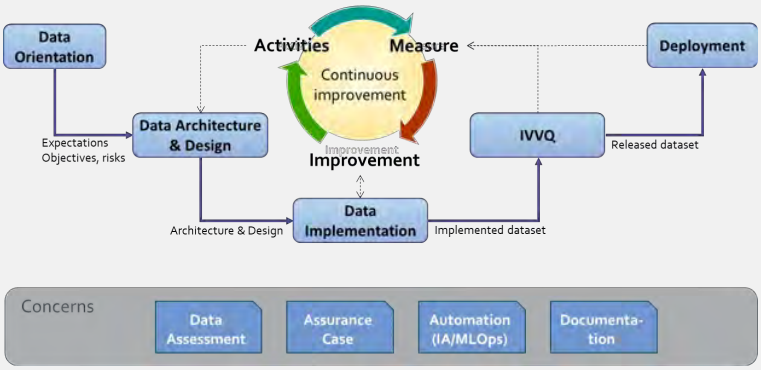
Benoît Langlois[4], Jean-Luc Adam[3], Xavier Baril[1], Eric Feuilleaubois[5], Faouzi Adjed[2], Flora Dellinger[5]

(1) Airbus, (2) IRT SystemX, (3) Renault, (4) Thales, (5) Valeo

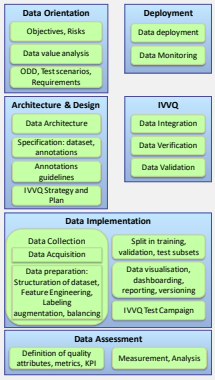
Towards Trustworthiness by ...
A Formalization of the Data Lifecyle for Complex and Critical Systems in AI



Complex system dataflow



Proposition of a Development Process for Data Engineering in AI

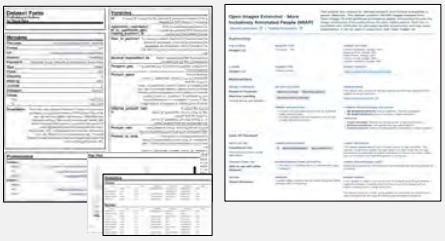


Identification of 70+ Data Activities

Towards Trustworthiness by ...
Documenting Data(sets)

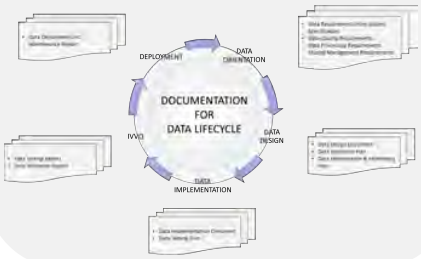
" ... Despite the importance of data to machine learning, there is currently no standardized process for documenting machine learning datasets ..." "Datasheets for datasets", Timnit Gebru et al

« Information sheet » to Increase **Transparency** as a first step

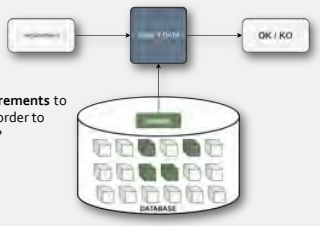


« Information Sheets » illustrations from literature : « Data Nutrition Label » and « Data Cards »

Towards Trustworthiness by ...
Documenting Data Engineering activities



Towards Trustworthiness with ...
a language for dataset specification



How to define **requirements** to specify a dataset in order to enable **verification** ?

Textual requirements are often ambiguous. ➡ Need for a **language** !

1. Define one table per requirement/data characteristic

Data characteristics can be:
- Categorical variables
- Quantitative variables

mapped into categories.

Rain	Count
YES	2000
NO	2000

List all possible values of the data characteristics with associated count target.

Speed	Count
[1, 25]	2000
[26, 50]	2000

2. Combine requirements

Speed	Rain	Count
[1, 25]	YES	1000
[1, 25]	NO	1000
[26, 50]	YES	1000
[26, 50]	NO	1000

Both requirements satisfied independently can lead to unbalanced dataset

Challenge for dataset specification: balance between perfect characteristics distribution (can lead to "waste" data) and unbalanced dataset.... => Language must allow to express **TOLERANCES**

Towards Trustworthiness with ...
Metrics for data assessment

In the data engineering activities, the data metrics provide a quantitative assessment of the adequacy of the datasets for the ML/AI task to be learned. The following metrics will be considered:

- **Representativeness** defines how the dataset represents the original population
- **Diversity** defines how the variety is respected. It guarantees the respect of probabilistic or deterministic distribution of the data
- **Completeness** describes the proportion of the missing information in a given dataset
- **Coverage** introduces the notion of wrapping a given class or situation learned by the model. It guarantees the proper execution of the model in the covered situation
- **Corner Cases** represent ambiguous data where the model fails. An estimation of their proportion and influence is needed.

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Bringing trust from ODD to Data

Expression and validation of an operational domain using extreme examples for computer vision applications

Adrien LE COZ[1,2,3], Stéphane HERBIN[2], Faouzi ADJED[1]

[1] IRT SystemX, [2] ONERA, [3] Université Paris-Saclay

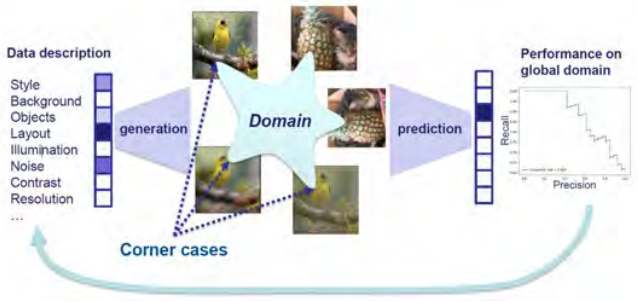
Context

- Deep Learning models revolutionized computer vision (image classification, object detection, …) but their predictions can't always be trusted (out-of-distribution data, adversarial examples, ambiguous data, ...).
- Need to determine the **operational domain** of a model: the conditions upon which its predictions can be considered reliable.

Objectives

- Define an operational domain based on extreme examples,
- Study its application and usefulness to:
  - evaluate Deep Learning models,
  - explain models,
  - monitor models during deployment,
  - manipulate the compromise data coverage / performance, ...

Approach



Global approach:

- Use a controllable generative model to manipulate images and generate extreme examples (corner cases).
- Those corner cases define the limits of the operational domain of a classifier.
- Link the classifier performance to the data description.

Technical details:

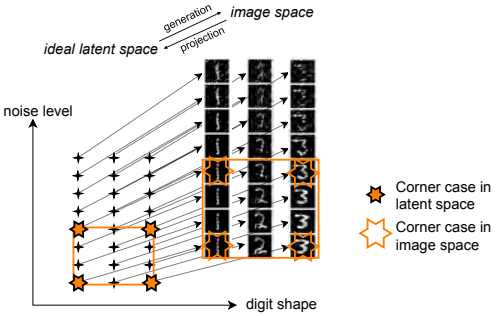
- Focus on multiclass image classification.
- Use the generative model StyleGAN2 [3] known for:
  - generation of high-quality images,
  - a disentangled latent input space allowing image manipulation [5].
- Use MNIST dataset (handwritten digits):
  - start with a simple dataset to validate the approach,
  - add blur and noise as visual attributes that influence classifier performance [4].

Results

- Find the visual attributes that most influence the classifier performance: mainly noise, blur, contrast, and shape.
- Generate corner cases by manipulating images along those attributes until classifier prediction changes.

- The latent space of the generative model allows image manipulation by moving into the directions that degrade the classifier prediction [1, 2].

Future work



- Validate the complete pipeline on a first use case (corrupted MNIST).
- Scale to more complex data.
- Explore the various applications of an operational domain.

References

- [1] Adrien Le Coz, Stéphane Herbin, and Faouzi Adjed. Leveraging generative models to characterize the failure conditions of image classifiers, 2022. *[Accepted at workshop AISafety 2022, IJCAI-ECAI-22]*
- [2] Oran Lang, Yossi Gandelsman, Michal Yarom, Yoav Wald, Gal Elidan, Avinatan Hassidim, William T. Freeman, Phillip Isola, Amir Globerson, Michal Irani, and Inbar Mosseri. Explaining in style: Training a gan to explain a classifier in stylespace, 2021.
- [3] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data, 2020.
- [4] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019.
- [5] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation, 2020.

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Self-supervised Learning for Anomaly Detection on Time Series

Olivier ANTONI and Marielle MALFANTE

CEA Grenoble

Context

Apply an innovative deep learning self-supervised method for anomaly detection on an unlabeled dataset.

Keywords

Time series, self-supervised learning, transfer learning, data representation, predictive models, 1D-CNN, anomaly detection.

Air Liquide use case: Efficiency Monitoring System (Data Quality)

The use case considered is a set of time series, without any annotation (unlabeled dataset for anomaly detection). Various sensors are used to record time series of several physical measurements (pressure, temperature), during a year. The sampling rate is not constant.

The method exposed in this study deals with univariate time series regularly sampled. This study is based on the temperature sensor System_3-TI1223.PV which was selected for its apparent stability in nominal regime.

The preprocessing step to build the working dataset includes:
- Linear interpolation to obtain a regularly sampled time series (1 sample per min),
- Normalization to have a centered and reduced working dataset,
- Windowing with a sliding window of 420 samples (7 hours) and a stride of 5 minutes.

Method

The proposed approach is a two-step method illustrated in the schematic:

- First a representation of the data is learned using a self-supervised approach. In this configuration, we use a pretext task of prediction to train a 1D-CNN:
  - The network architecture, composed of [3 x Conv1D(number of filters: 32, kernel size: 3) + 1 x MaxPooling1D(pool size: 2)] repeated 3 times, is chosen for its ability to extract a robust and general representation of the data.
  - Each data composed of 420 samples is split: the first N1 = 280 samples are used as input, the last N2 = 140 samples (ground truth) are used as output for the prediction task. The network is trained on this prediction task using ground truth data.
- In a second step the neural network is used to detect anomalies: an anomaly is raised each time the value of the prediction metric is above a fixed threshold.

Results

The training of the predictive neural network is done on the January data (8 845 data) and the validation on the February data (8 269 data). Data from other months of the year (about 42 000 data each month) are used to test the method.

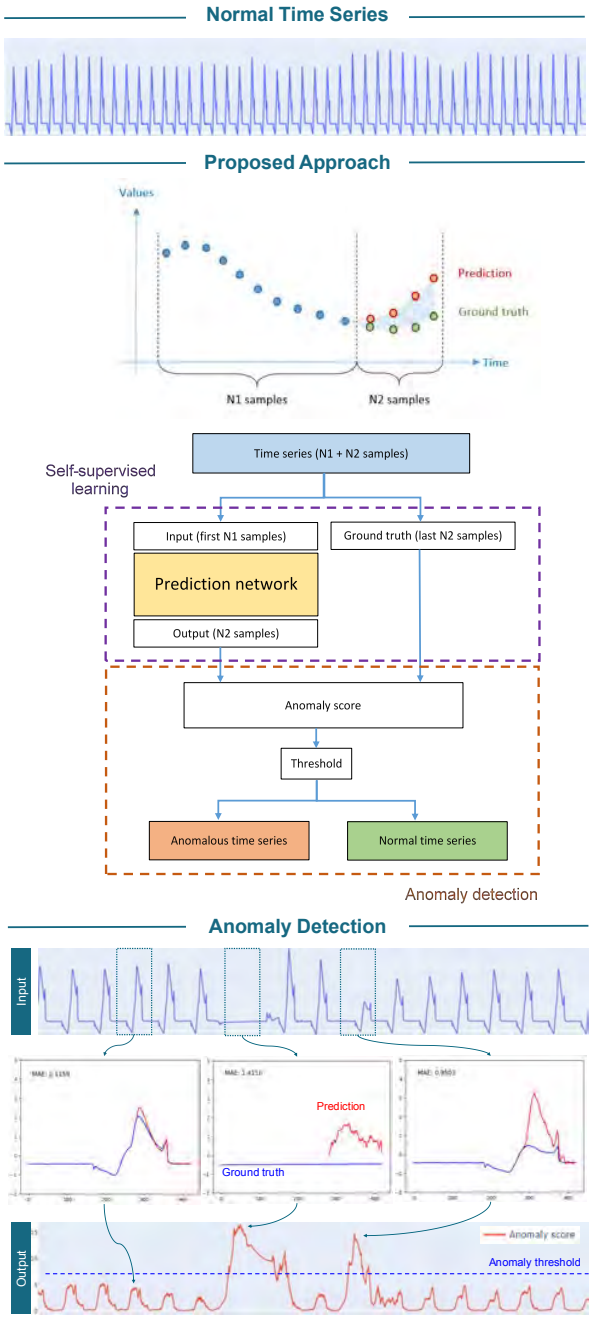
The anomaly score is equal to the value of the prediction metric (MAE) divided by the 99th percentile of the metric values calculated on the training set (p_{99} = 0.1123).

The anomaly threshold is chosen such that the anomaly score is slightly higher than the value for which the signal variations of the test set are considered normal by an expert.

The proposed method detects 142 anomaly segments over the full year. These anomalies are mainly due to process anomalies or changes in the production regime. Examples of network prediction results for nominal and anomalous data, along with the resulting anomaly scores, are displayed on the plots.

Prospects and future work

Future work on this use case includes using an alternative pretext task and applying the method to multivariate or multimodal time series for more robust predictions.



Bringing trust from ODD to Data

Anomaly Detection on Vibratory Sensors with Perceivers

Laurence GUILLON(1), Amélie BOSCA(2), Michel POUJOL(2)

(1) Naval Group Toulon, (2) Sopra Steria Aix/Toulon

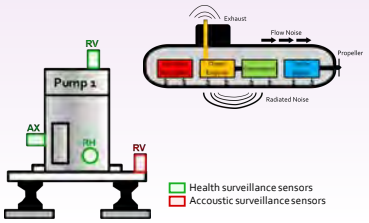
Introduction

Ships (surface vessels or submarines) are equipped with thousands of sensors of different types. Among these, **vibration sensors** are distinguished by the fact that they must meet 2 needs:

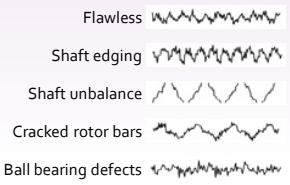
- to detect or prevent breakdowns,
- to ensure that the acoustic stealth of the ships has not deteriorated.

Set up all these sensors might be **expensive** and **difficult to maintain**. This study explore the way to drastically reduce their number without compromising on these 2 types of requirements.

For this study, a pump similar to that installed on some ships was installed on this test bench with 4 vibration sensors, just as on board.

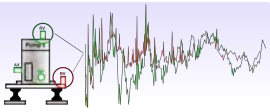


Operating states



Problem

Is it possible with some AI model to detect and classify anomalies just as it is when using the 3 health surveillance sensors, but **using only the single acoustic surveillance sensor** that is mounted on the base?



Perceivers

Perceivers are specific types of **Transformers**. Just like Transformers, they are essentially a stack of main blocks using 3 types of conceptual representations of their inputs that evolve over the blocks:

- **Queries** (Q) = current representations of learned useful concepts,
- **Keys** (K) = current representations of input information considering all its learned angles of interpretation,
- **Values** (V) = current representations of input information considering all the learned useful concepts it may contain.

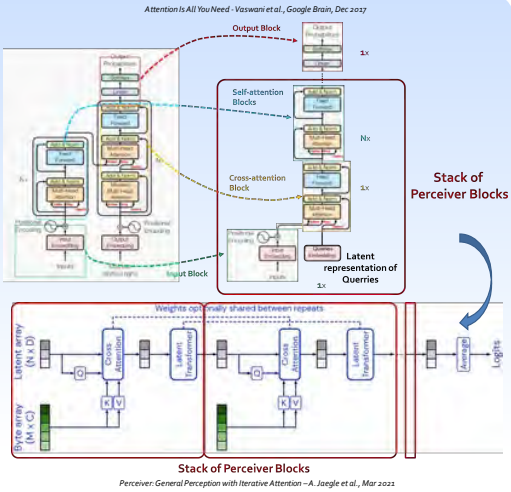
They differ from Transformers principally by the fact that each of their main blocks, uses a **cross-attention** sub-block instead of a (masked) self-attention one. This offers the **possibility to represent the useful concepts (Queries) in a latent space which can be much smaller than the space of representation of the inputs** from which they are extracted.

Thanks to this, the **computational complexity of Perceivers is no longer quadratic** and therefore, they can handle **inputs of much larger sizes than Transformers**.

This **decoupling of Queries and input representations** offers **2 other important advantages** in the learning process. It offers **a lot of flexibility and versatility**:

- not only in **handling, possibly simultaneously, very heterogeneous inputs** both in terms of their nature (**images, time series, texts, videos…**), structure and shape,
- but also, in the **pre-processing** of these inputs since it allows to do a **full Feature Learning**.

In this case of full feature learning, the latent representation of Queries in the first Perceiver block is initialized at random. But the **Perceivers also allow to initialize this latent space with some other kind of encoders** (CNN, LSTM, …) and/or to do **more or less Feature Engineering**.



Experiment

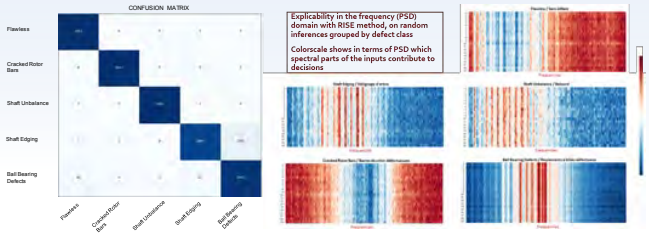
Perceivers have been tested on a **very challenging Use Case**:

- **Very high sampling frequency** (typically > 20 kHz),
- **Very unstable and fluctuating signal**,
- **Most of the data is noise** and useful information is really buried in that noise,
- **Signal to noise ratio may be very low** for some of that useful information to be extracted,
- **No prior Feature Engineering** (except resampling), **only Feature Learning**,
- 4 kinds of generally **hard-to-detect defects:** shaft edging, shaft unbalance, cracked rotor bars, ball bearing defects

Results

Perceivers proved:

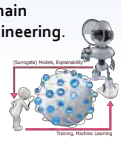
- to be **very effective in detecting anomalies** with a **fast and efficient learning**,
- to have **very well learned** how to detect these anomalies,
- to be able to extract:
  - **very accurate and consistent patterns for each type of anomaly**,
  - **very distinct patterns between each type of anomaly.**



Perspectives

There are many other use cases for which this type of solution could also be valuable (e.g., use cases with **multimodal anomaly detection** needs, **mixing of very heterogenous data**, …).
Perceivers offer other perspectives, like:

- **Explainability by Design** (using cross-attention weights),
- **Domain Adaptation** (e.g., from test bench data/models to « true » data/models),
- **Semi-supervised learning**,
- Feature Learning for **better domain understanding and system engineering**.



Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Bringing trust from ODD to Data

Robustness using fairness: problem formulation

Evgenii Chzhen[1], Mohamed Hebiri[2], Jean-Michel Loubes[3], Gayane Taturyan[2,3,4]

(1) Université Paris-Saclay, CNRS, LMO; (2) Université Gustave Eiffel, LAMA; (3) Université Toulouse III - Paul Sabatier, ANITI; (4) IRT SystemX

Introduction

- Data-driven algorithms can inherit biases that are present in the data, degrading their performance.
- A prediction algorithm may exhibit different behaviors for different groups of individuals.
- **Question:** Can enforcing fairness make the prediction model more robust?

Model

- $\mathcal{X} = \mathbb{R}^d$ - input space, $\mathcal{Y} = \mathbb{R}^d$ - label, $\mathcal{S} = \{-1,1\}$ - sensitive attribute
- (X, S, Y) are drawn from $\mathbb{P}$ on $(\mathcal{X}, \mathcal{S}, \mathcal{Y})$
- $h: \mathcal{Z} \to \mathcal{Y}$ - measurable function (predictor)
  - $\mathcal{Z} = \mathcal{X} \times \mathcal{S},\ Z = (X, S)$ - aware case
  - $\mathcal{Z} = \mathcal{X},\ Z = X$ - unaware case
- $l: \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ - loss function

Definition (Risk Parity) A predictor $h: \mathcal{Z} \to \mathcal{Y}$ satisfies Risk Parity (RP) w.r.t distribution $\mathbb{P}$ if $\mathbb{E}_P[\, l(h(Z), Y) \mid S = s\,] = \mathbb{E}_P[l(h(Z), Y) \mid S = s'\,]$ for all $s, s' \in \mathcal{S}$.

- $h^* \in \underset{h:\, \mathcal{Z} \to \mathcal{Y}}{\operatorname{argmin}} \mathbb{E}_P[l(h(Z), Y)]$ - (unfair) optimal predictor
- $h_f^* \in \underset{h_f: \mathcal{Z} \to \mathcal{Y}}{\operatorname{argmin}} \{\mathbb{E}_P[l(h_f(Z), Y)] : h_f \text{ satisfies RP}\}$ – fair optimal predictor

Challenges

Which one of the above predictors is better depending on the test distribution $\mathbb{P}'$?
It is shown (1) that

- $\mathbb{P}$ and $\mathbb{P}'$ are close - h^* is still a good for $\mathbb{P}'$
- Majority ($S = 1$) becomes minority ($S = -1$) in $\mathbb{P}'$ - h_f^* is better

Open questions:

- What is the precise expression of h_f^* ?
- What distance measure must be used to control the changes in distribution?

Proposition

Solve the following optimization problem

$$h_f^* \in \underset{h_f: \mathcal{Z} \to \mathcal{Y}}{\operatorname{argmin}} \{\, \mathbb{E}_P[l(h_f(Z), Y)] : \mathbb{E}_P[\, l(h_f(Z), Y) \mid S = -1\,] = \mathbb{E}_P[\, l(h_f(Z), Y) \mid S = 1\,]\}$$

for binary classification, i.e. $l(h_f(Z), Y) = \mathbb{I}(h_f(Z) \neq Y)$, and when

1. $\mathcal{Z} = \mathcal{X} \times \mathcal{S},\ Z = (X, S)$
2. $\mathcal{Z} = \mathcal{X},\ Z = X$

Future Work

- Extending the results from a binary sensitive attribute to a multi-valued one.
- Extending the results to continuous sensitive attribute.
- Finding an optimal distance measure to control the changes in distribution.

Expected application

The derived fair predictor can be applied on Thales LAS Aerial Photograph Interpretation / EuroSAT use case dataset, to solve the Highway/River binary classification problem, where a small percentage of images (~3%) has a certain blue-veiled property.

References

[1] Maity, S., Mukherjee, D., Yurochkin, M., and Sun, Y. (2021). Does enforcing fairness mitigate biases caused by subpopulation shift?

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Sparsity based anomaly detection framework

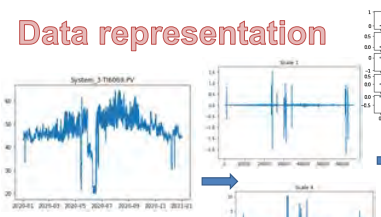
Fred NGOLE MBOULA

CEA Paris Saclay

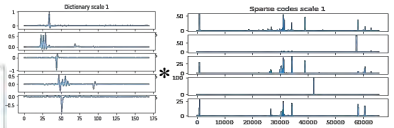
Introduction

- Modern industrial plants produce increasingly complex datasets by monitoring of thousands of assets, to the end of identifying past anomalies or future failures and/or optimizing underlying processes.
- These data typically showcase a high degree of non stationarity related to the complexity of the underlying processes, and hence a wide variability of temporal structures, with limited statistical a priori on the time series.
- In this setting, we propose a general model free anomaly detection framework, aimed at dealing with complex signal structures while making mild assumptions on their behaviors.
- We illustrate the proposed approach on a industrial dataset made of temperatures recorded on different assets of complex plant over a year.

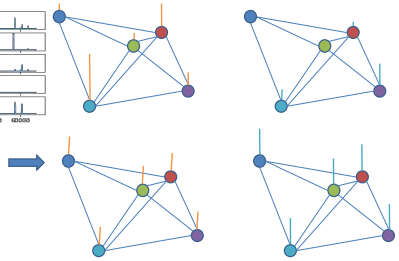
Data representation



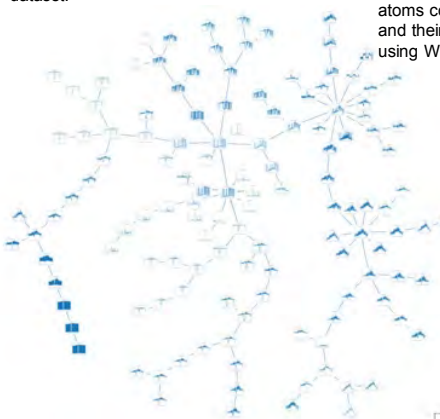
a) Wavelet decomposition



b) Convolutional dictionary learning on the wavelets coefficients



Temperatures dataset presented as a graph whose adjacency matrix is derived from the pairwise dissimilarities calculated on the dataset.

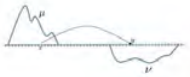


Globally anomalous signal identification based on pairwise time series dissimilarity matrix

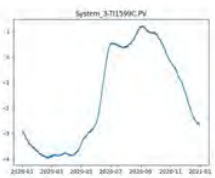
The univariate anomaly detection simply consists in applying the Isolation Forest method to the samples sets made on the sparse codes of the time series considered at each scale, endowing the detection spaces with the graph based metric defined in the top right box.

Time series dissimilarity metric

We build a dissimilarity metric between the original time series based on their atoms correlations at different scales and their contributions to signal energy, using Wasserstein distances.



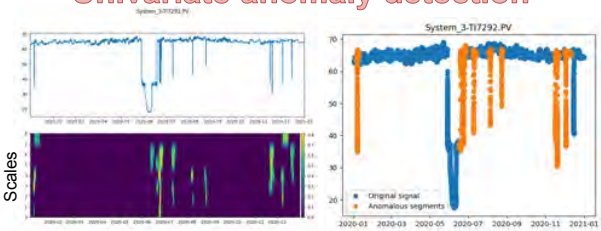
Collaborative anomaly detection consists in searching anomalies jointly for structurally similar time series, by merging their respective sets of sparse codes at each scale, which provides a statistically more complete basis for identifying normality and thus reducing false positives



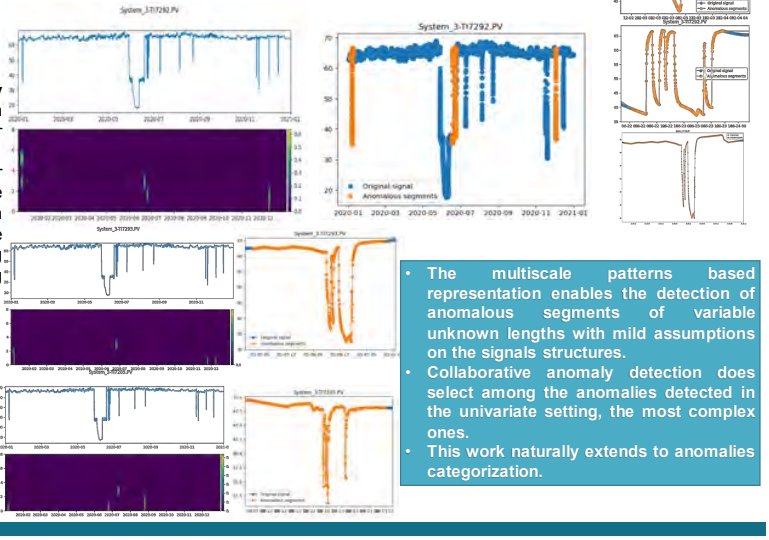
Metric in the sparse codes space

- Euclidean distance does not take the correlations between the atoms into account. We consider a fully connected weighted graph representing the atoms and their pairwise distances.
- Each sparse code can be taken as the initial condition of the diffusion equation on this graph.
- We choose the distance between two sparse codes as the L_2 distance between the diffusion equation solution they generate on the graph.

Univariate anomaly detection



Collaborative anomaly detection



- The multiscale patterns based representation enables the detection of anomalous segments of variable unknown lengths with mild assumptions on the signals structures.
- Collaborative anomaly detection does select among the anomalies detected in the univariate setting, the most complex ones.
- This work naturally extends to anomalies categorization.

Bringing trust from ODD to Data

Leveraging unlabeled data to improve active learning for trustworthy data selection and annotation

Fritz Poka Toukam[1] , Nicolas Granger[1] , Oriane Siméoni[2], Angélique Loesch[1]

(1) CEA-List, Université Paris-Saclay; (2) Valeo

Active learning for object detection on industrial use cases

- To perform well, classical deep learning techniques rely on large collections of samples, fully annotated by hand. It is a very expensive and fastidious task.
- **Active Learning (AL) iteratively selects batches of data for annotation** (acquisition function) and retrains the model to optimally assign the annotation budget.
- From an industrial point of view, we propose to apply AL:
  - On a realistic use-case: detection on images from driving scenes
  - With **Yolov5 detector** : fast, data efficient, widely deployed in production
- Our previous work [1] in Confiance.AI in 2021 shows the performance of traditional active learning strategies on this scenario

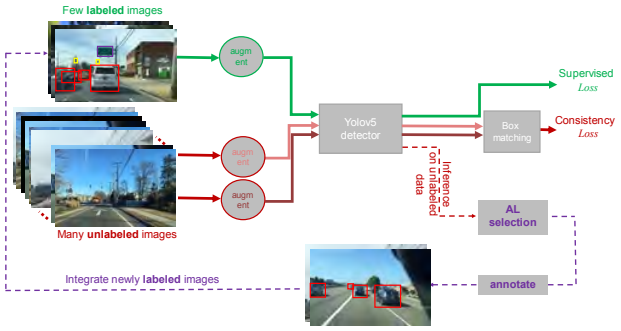


Figure 2 : Overview of the proposed AL pipeline which incorporates the consistency metric and unlabeled samples during training.

Consistency loss to leverage unlabeled data

- By definition, a lot of unlabeled data is available in AL scenario, but **traditional AL techniques ignore unlabeled data** during representation learning
- The **consistency metric** between two randomly augmented views of an image has been introduced recently as a **self-supervised objective for AL model training** [2], and as a **criterion in the acquisition function** [2,3]
- For object detection, the consistency is based on the **matching of predicted boxes** between two augmented views (figure 1)
- **Improvements to this consistency metric** are proposed and illustrated in figure 2:
  - using **Generalized IOU loss** instead of L1 as the regression objective
  - using **stronger augmentations** from the Yolov5 [4] pipeline

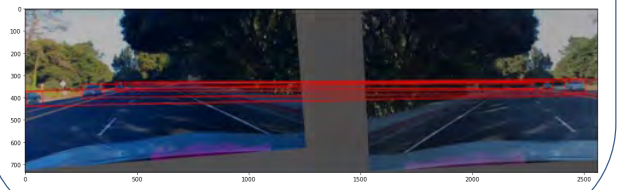


Figure 1 : Box-matching over two augmented views of the same image

Results and future work

- Initial experiments of our approach (Figure 2) on BDD100k show consistent gains of self-supervision in initial and later cycles (Figure 3)

Future experiments :

- Complement box regression consistency [2] with classification objective
- Use **consistency loss as a score** for the acquisition function [2,3].
- Refine box-matching heuristic to accomodate strong yolov5 augmentations
- Application to the Valeo Use Case

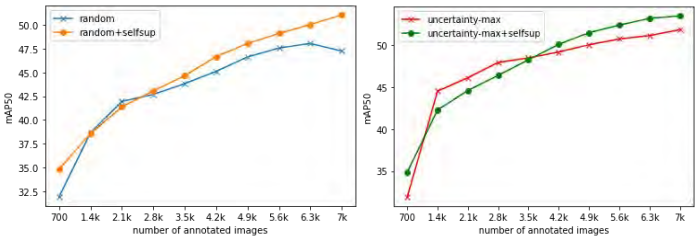


Figure 3 : Results of active learning trainings with and without self-supervision on BDD100k (average of 3 runs with different seed images), using random (left) and max-uncertainty acquisition function (right)

Conclusion

- Promising results
- Well tuned for detection task
- Minimal disruption of detection models and AL pipelines

References

[1] Fritz Poka Toukam, Thomas Dalgaty, Hedi Ben-Younes, Nicolas Granger, Spyros Gidaris, Camille Dupont, Oriane Simeoni. Is active learning better than random selection for real-word tasks ? In the Confiance.ai Days 2021 Poster Booklet. Confiance.ai Days 2021, Oct 2021, Toulouse, France. 2021. hal-03687605
[2] I. Elezi, Z. Yu, A. Anandkumar, L. Leal-Taixe, and J. M. Alvarez. Not all labels are equal:Rationalizing the labeling costs for training object detection. InProceedings of the IEEE/CVFConference on Computer Vision and Pattern Recognition (CVPR), 2021.
[3] W. Yu, S. Zhu, T. Yang, C. Chen, and M. Liu. Consistency-based active learning for objectdetection.CoRR, abs/2103.10374, 2021
[4] G. Jocher. Ultralytics /yolov5.

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Bringing trust from ODD to Data

Proposition of an ODD engineering process

Christophe Bohn (IRT SystemX), Kévin Mantissa (IRT SystemX), Gabriel Burtin (IRT SystemX)

Methods and Tools for Operational Design Domain Project

ODD is dedicated to a **system-level feature** with **automation capabilities**

ODD is a **chosen restriction** of the Operational Domain by **system designers** in order to **ensure feature usage**

ODD is a **part of** the **system specification** dedicated to **external conditions** on which development efforts will be focused

Operational Design Domain (ODD) definition by the SAE J3016 :

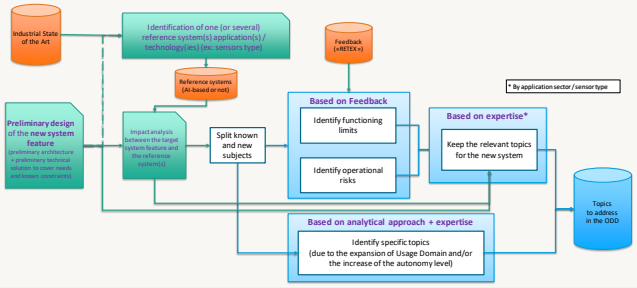
"Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics"

ODD is defined for a **specific design** of a system-level feature and its **technical limitations**

ODD was initially created by the **autonomous automotive** industry

ODD shall be **understandable** for the **end users** and the **certification authorities**

STEP 1 – Identification of topics



Inputs:
- Preliminary Design of the new system feature which the ODD has to address
- Impact analysis between new system feature and reference systems (reference for applications or technologies)
- Related industrial State of the Art allows to identify potential limitations and risks of the new system feature

Activities:
- Segregation of new subjects for analytical approaches and known subjects addressed by reference system knowledge
- An analytical approach on new subjects allows to identify specific topics to address in the ODD. It requires system expertise
- Functional limits and operational risks are identified in feedback of reference systems. Only topics relevant for the new system feature are kept by system experts

Outputs:
- The previous identified topics are collected to be addressed in the next step of the ODD process

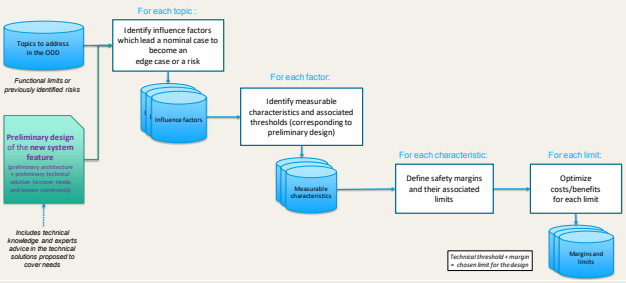
STEP 2 – Identification of characteristics

Inputs:
- Topics to address in the ODD identified in Step 1
- Preliminary design of the new system feature is required in this analysis

Activities:

Step 2 is applied on each topic one by one
- For each topic, influence factors are identified in relation with the new system feature
- For each factor, measurable characteristics have to be considered with their thresholds for engineering activities
- For each characteristic, a safety margin is taken into account and allows to define the acceptable limits
- A local optimization is done between development costs and usage benefits to confirm or extend limits

Outputs:
- Influence factors with their measurable characteristics and optimized limits



Example 1:
- **Topics to address:** Cruising in its lane
- **Associated influence factors:** Capability to perceive the lane markings
- **Associated measurable characteristics:** Distance that can be covered without lane detection + lane markings readability on the ground
- **Associated margins & limits:** Guarantees lane marking, except for roadworks areas.
- **Optimized costs/benefits:** Multiple lane markings, colored in roadworks areas

Example 2:
- **Topics to address:** Entering/exiting a tunnel
- **Associated influence factors:** Quick variation of luminosity
- **Associated measurable characteristics:** Variation of luminosity/temporality
- **Associated margins & limits:** No anticipation possible leading to an exclusion of tunnels from the ODD
- **Optimized costs/benefits:** Setting up luminosity sensors with V2X communication for each tunnel is too expensive

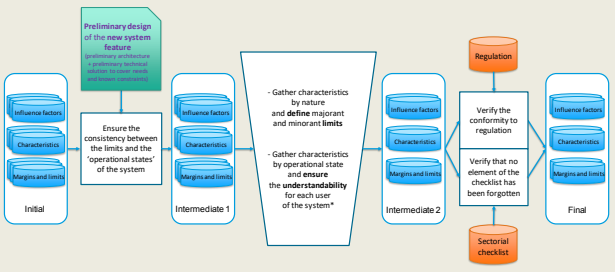
STEP 3 – System Consolidation

Inputs:
- Influence factors with their measurable characteristics and optimized limits
- Preliminary design of the new system feature
- Regulation and sectorial checklist

Activities:

Step 3 aims to consolidate all the topics together
- From an initial set of characteristics and limits of all topics, ensure consistency with the operational states of the system
- Characteristics are gathered by nature for all topics to identify majorant and minorant limits
- Characteristics are gathered by operational states to grant understandability for system users
- Check the exhaustiveness of ODD characteristics thanks to sectorial checklists and verify the conformity with regulation

Outputs:
- Preliminary ODD describing influence factors with their characteristics and relevant limits



** Assumption : the system is composed of only one autonomous feature/service for this ODD.*

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

ODD usages in a data and ML monitoring perspectives

Project: Methods and Tools for Operational Design Domain

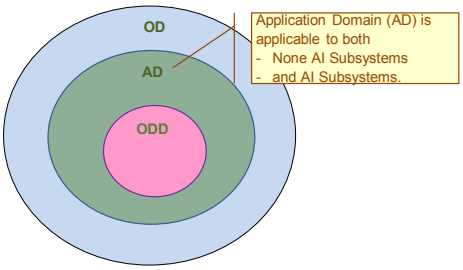
Georges Jamous (Airbus Protect), Morayo Adedjouma (CEA)

ODD Description

The Operational Design Domain (ODD) was originally defined for Driving Automation Systems for On-Road Motor Vehicles, by SAE J3016 (*). The ODD term definition is evolving to consider ML modeling for multiple operational domains (road vehicles, aeronautic, etc.) {SAE AIR6988, EASA, SAE AS6983, etc.} but there is no consensus on a unique ODD definition. ODD aims to describe foreseeable operational conditions an AI-based system will operate within.

(*) SAE J3016 (Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles): "*Operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics*"

ODD Usage perspectives



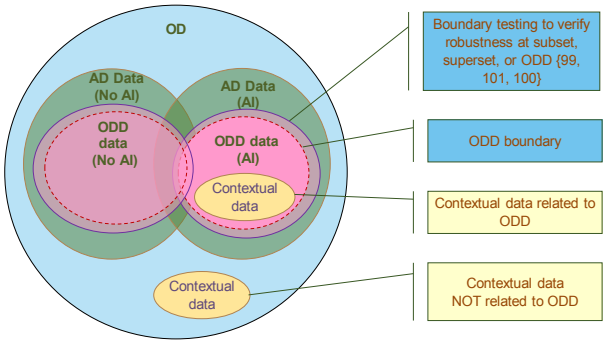
ODD from a Functional and Design perspectives

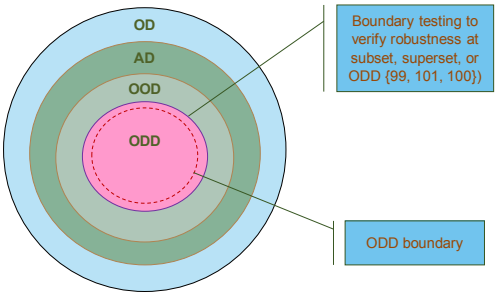
- The Operational Domain (OD) represents the real world in which the subsystem will operate at a given moment of time.
- The Application Domain (AD) describes foreseeable operational conditions a subsystem will operate within. Foreseeable elements in this context must be measurable.
- The ODD is a voluntary restriction of the operational domain (OD) in which the subsystem concerned by AI is intended to operate.

ODD from Data and ML engineering perspectives

Specific information is needed to exploit the ODD depending on the engineering phase:

- for data engineering, additional contextual information is needed on the data nature and the process of collecting them to elaborate the different Training, Validation and Testing datasets. .
- For ML engineering, data for ensuring the AI-system performance at ODD's boundary must be specified.





ODD from Verification, Validation, and Monitoring perspectives

- At design time, the ODD must be verified and validated against the expected properties specified within the system requirements for the AI-system.
- At operation time, the ODD should be monitored to detect any deviation from the AI-system nominal behavior and apply the mitigations measures accordingly. So, the AI-system must also be validated with respect to its robustness against the ODD boundary and OOD (Out Of Domain) data.

Village End-to-end approach for trusted AI systems and V&V (posters)

Introduction to the themes of the village

Guillermo Chaley Gongora, Boris Robert, Cyprien de la Chapelle

To fully support the needs of its industrial members, Confiance.ai shall deliver a consistent end-to-end approach for the development of trusted AI-based systems.

This approach shall be methodological, addressing the full cycle of engineering activities: at system level and at AI component level, from need analysis, specification architecture, design and implementation, to IVVQ and maintenance.

It shall be tooled by a Trustworthy Environment, implementing this end-to-end approach and assisting engineers of various domains (systems engineering, data engineering, AI algorithm engineering, embedded software engineering, etc.)

Such an approach shall allow Confiance.ai members to implement and secure iterations, feedback, consistency during engineering activities, and to build justifications, demonstrate the trustworthiness of their AI-based systems.

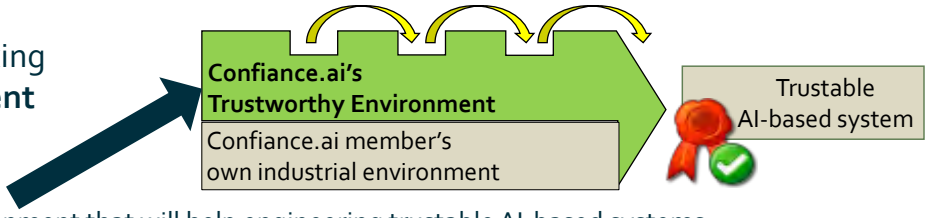
As illustrated by the following posters, Confiance.ai aims at demonstrating:

- How modeling can help getting cohesive engineering workflows and an adequate architecture of the Trustworthy Environment that supports them.
- How a consistent set of engineering methods for trusted AI-based systems can be built, from standards and from methodological contributions developed by Confiance.ai teams in their respective fields.
- How the “classical” Systems Engineering lifecycle shall be accordingly modified/augmented in order to take into account the specificities of AI-based systems.
- How Assurance Cases can help the Verification & Validation of AI-based systems, with trustworthiness being characterized as a set of fundamental properties that shall be proved to be satisfied.
- How trustworthiness properties, or attributes, based on objective criteria and taking into account the multi-dimensional nature of trustworthiness, can be identified, structured, measured and mapped onto engineering processes.
- How the “Trustworthy Environment” delivered by Confiance.ai to its members will assist their AI engineering, by proposing and tuning trustable engineering workflows, accompanying the implementation of engineering processes and monitoring the trustworthiness properties.

Modeling for the description of use and architecture of Confiance.ai's Trustworthy Environment

Boris ROBERT (IRT Saint Exupéry)

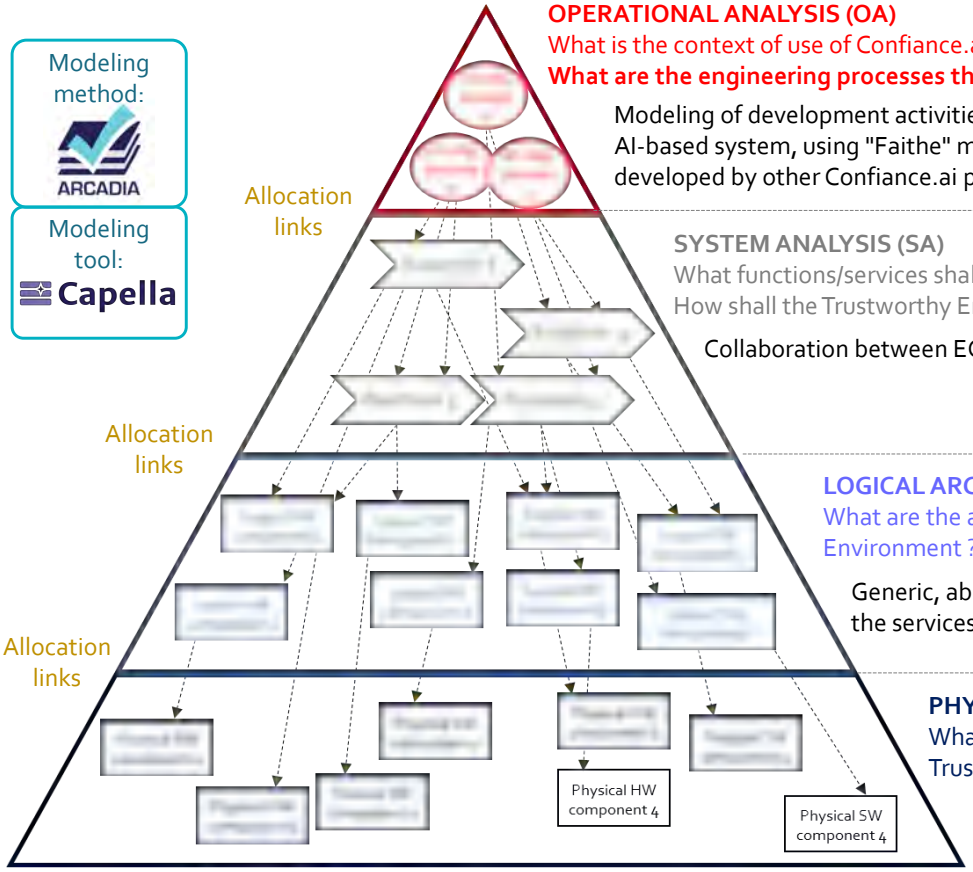
Analysis and architecture modeling, taking **Confiance.ai's Trustworthy Environment** as the System of Interest



The focus of this poster is the Trustworthy Environment that will help engineering trustable AI-based systems.

Modeling method: ARCADIA

Modeling tool: Capella



OPERATIONAL ANALYSIS (OA)

What is the context of use of Confiance.ai's Trustworthy Environment ?

What are the engineering processes that the Trustworthy Environment has to support ?

Modeling of development activities (EC2) and IVVQ activities (EC6) for a trustable AI-based system, using "Faithe" metamodel and taking into account the M&P developed by other Confiance.ai projects EC3/4/5/6/7.

SYSTEM ANALYSIS (SA)

What functions/services shall be performed by the Trustworthy Environment ?

How shall the Trustworthy Environment support the processes described in OA ?

Collaboration between EC2/EC6 (specifier) and EC1 (implementer).

LOGICAL ARCHITECTURE

What are the abstract resources to be used by the Trustworthy Environment ?

Generic, abstract architecture defined by EC1 to implement the services described in LA.

PHYSICAL ARCHITECTURE

What are the concrete resources to be used by the Trustworthy Environment ?

Instantiated architecture defined by EC1, integrating the software components produced by EC3/4/5/7.

Objectives:

- Get cohesive workflows (result of OA) that will be recommended to the users through the companion of the Trustworthy Environment.
- Get a cohesive functional scope of the Trustworthy Environment (result of SA)
- Get a consistent architecture (result of LA/PA) of the Trustworthy Environment that will be delivered by EC1 to industrial partners.

End-to-end approach for trusted AI systems

Capturing and modeling the engineering processes for trustable AI-based systems

Boris ROBERT (IRT Sant Exupéry), Afef AWADID (IRT SystemX)

How building a consistent set of engineering methods that allows to develop trustable AI-based systems ?

1. Capture: provides technical matter

Top-down approach:
capture of a high-level, holistic vision of an engineering process for trustable AI-based systems

Bottom-up approach:
capture of Methods & Processes elaborated by Confiance.ai Projects for specific topics

ISO, SAE INTERNATIONAL, EASA, afnor, ...

Generic vision based on Standards and working groups

can guide the construction of...

can extend, update, detail...

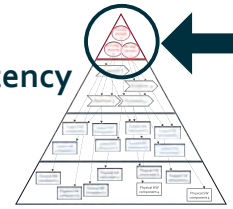
Vision based on Confiance.ai's Action Sheets and Uses Cases

EC3
EC4
EC5
EC6
EC7

- EC3: Characterization & qualification of trustworthy AI
- EC4: Design for trustworthy AI
- EC5: Data engineering for trustworthy AI
- EC6: IVVQ for trustworthy AI
- EC7: Target embedded trustworthy AI

This vision is represented in the Model by using concepts of the Meta-model

2. Modeling: ensures consistency

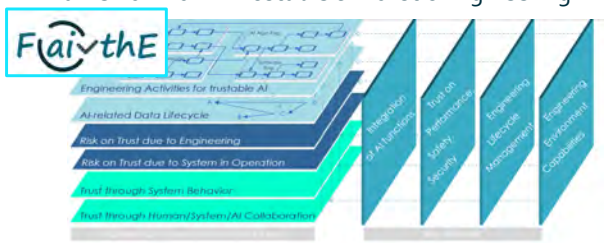


Focus on the "modeling of engineering methods" part

Modeling pyramid described in Poster n°1

This vision is represented in the Model by using concepts of the Meta-model

Framework for AI Trustable & Holistic Engineering



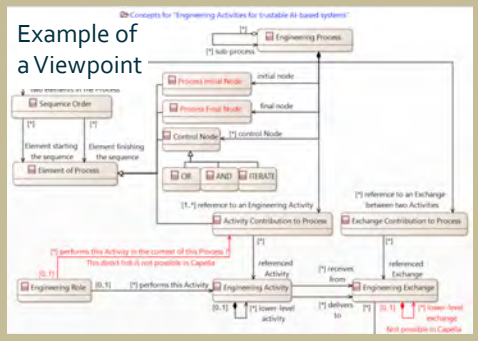
Several crossing Viewpoints to capture the concerns

Meta-model

Implemented in Capella

Meta-model describing all viewpoints of FaithE
+
mapping between concepts of the meta-model and Capella concepts to be used in the Model

Example of a Viewpoint

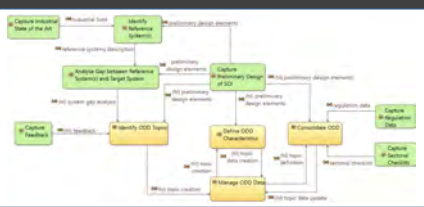


Model

Implemented in Capella

- Model fragment for high-level engineering tasks
- High-level breakdown of specific AI activities, Connections between these activities
- ...

Example of modeled process for ODD (in work)



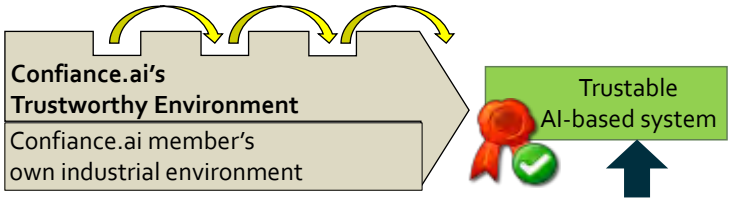
- Model fragment for AI monitoring
- Model fragment for robustness test
- Model fragment for ODD building
- Model fragment for ...

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

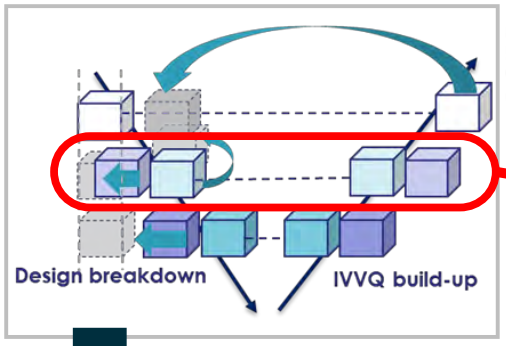
End-to-end method for the engineering of trustable AI-based systems

Boris ROBERT (IRT Saint Exupéry), Xavier LE ROUX (Thales),
Christophe ALIX (Thales)

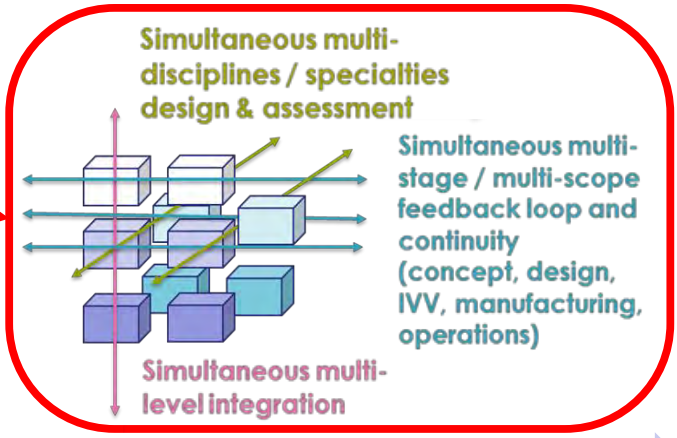
How shall the "classical" Systems Engineering lifecycle be modified/augmented in order to take into account the specificities of AI-based components of a system ?

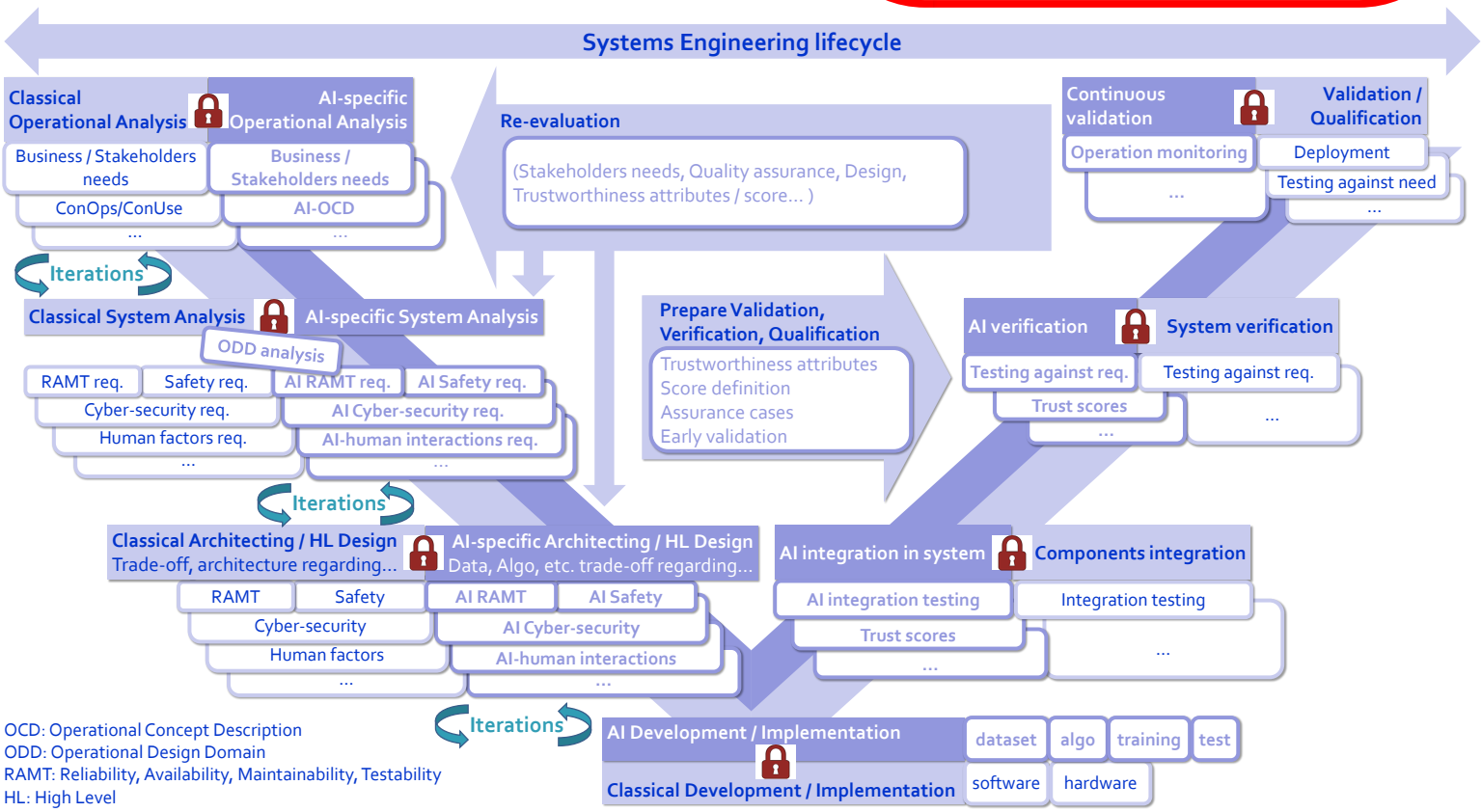


The focus of this poster is a trustable AI-based system developed by a member of Confiance.ai.



Detailed view (**work in progress as described in poster n°2**)





OCD: Operational Concept Description
ODD: Operational Design Domain
RAMT: Reliability, Availability, Maintainability, Testability
HL: High Level

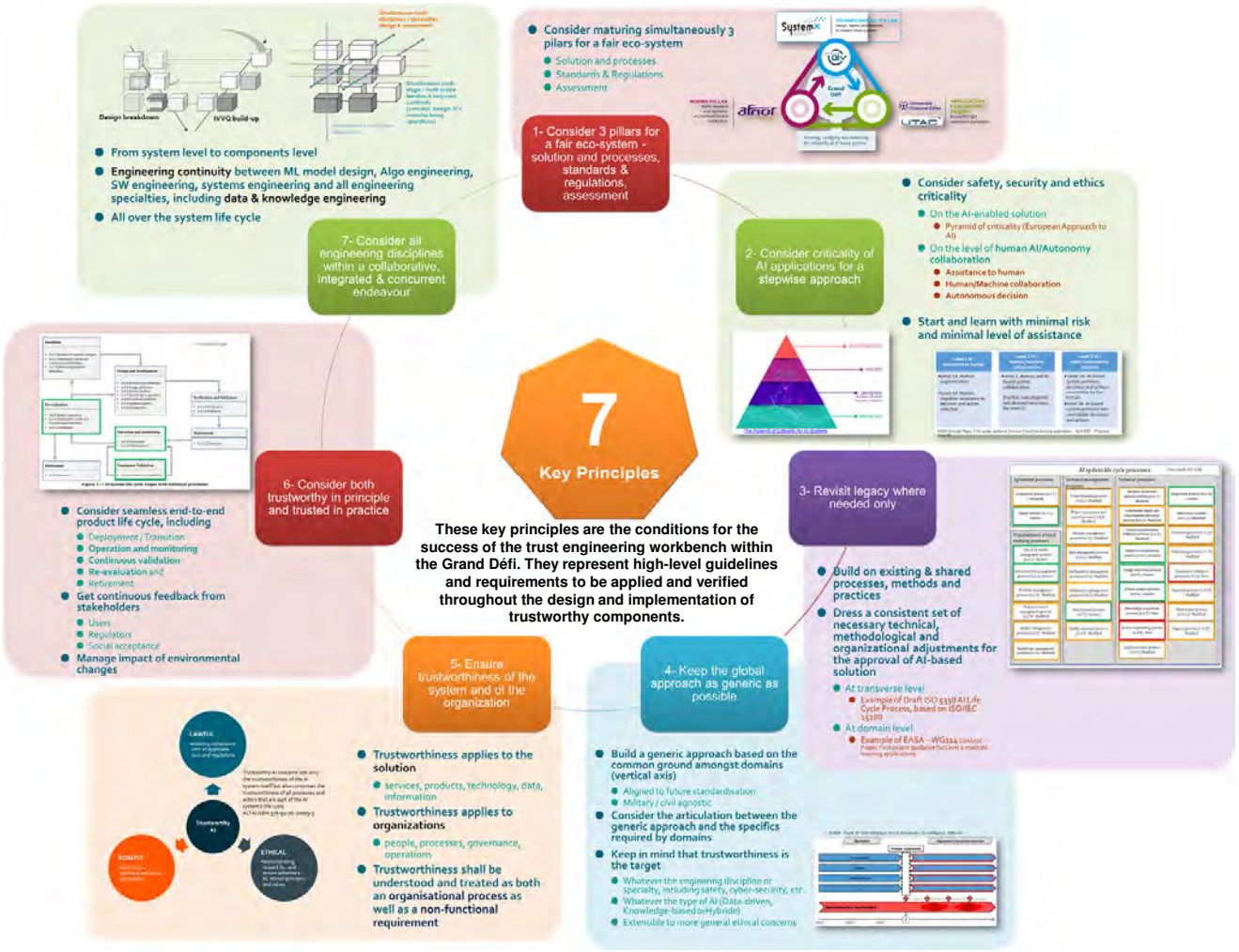
Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

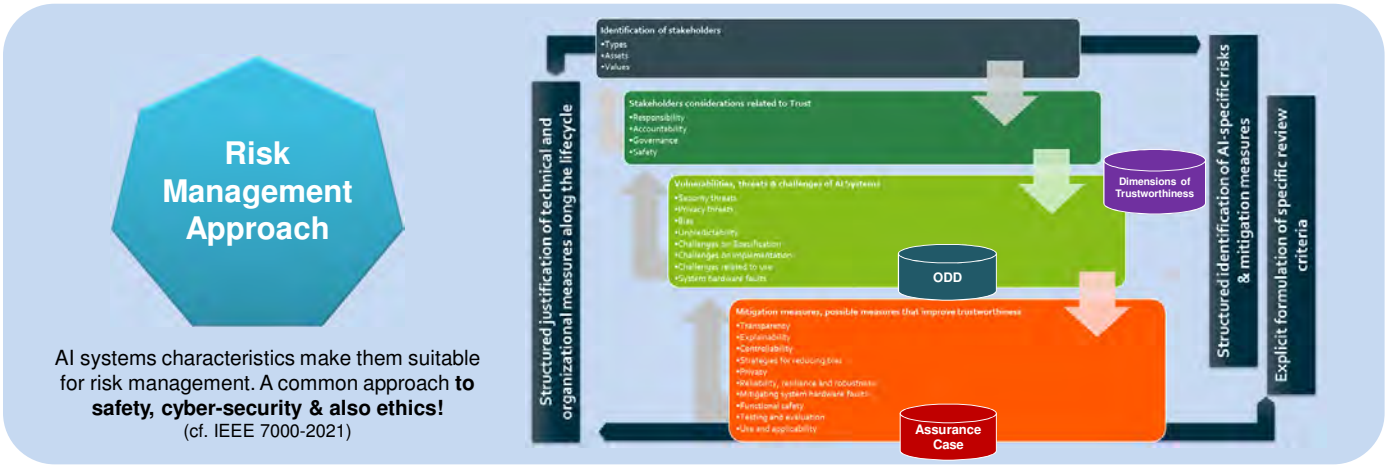
Engineering Trustworthy AI Systems End to End Vision

7 Key Principles & Global Approach

Christophe Alix, Guillermo Chale-Gongora, Jean-Luc Voirin

Thales





Can we assess AI-based system trustworthiness ?

Juliette Mattioli ♣ ; Agnès Delaborde *,♠ ; Henri Sohier ♠

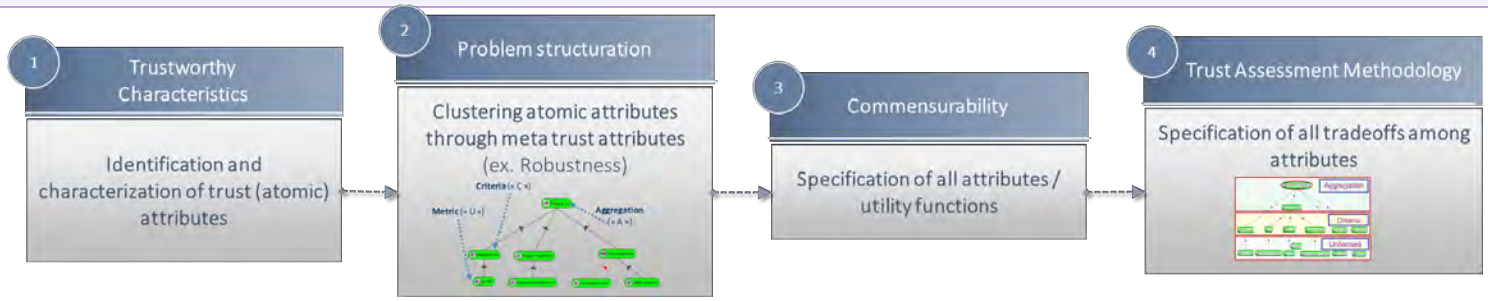
♣ Thales – *LNE – ♠ IRT SystemX

Project: Process and Methods

Rationale

Due to the multi-dimensional nature of trustworthiness, the main issue is to establish objective criteria trustworthiness attributes clearly identified and mapped onto the AI processes and its lifecycle

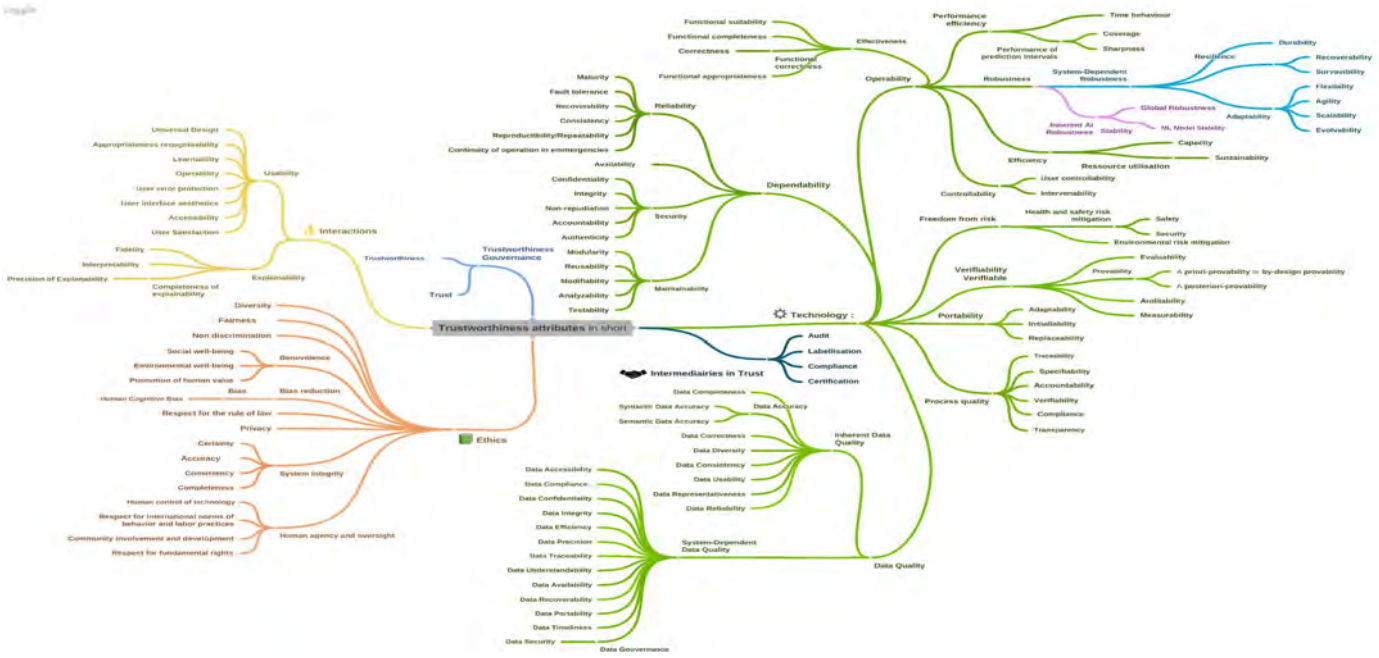
Unified approach on trustworthiness attributes based on Multi-Criteria Decision Aiding



Confiance.ai approach is based on the following steps:

1. Step 1: Definition of the different attributes that constitute trustworthiness
2. Step 2: Structuring of the attributes in a semantic tree allowing a first hierarchy ….
3. Step 3: Identification of metrics, assessment methods or control points for each atomic attribute;
4. Step 4: Definition of an aggregation methodology to capture operational trade-offs and evaluate higher-level attributes.

A first Trustworthiness Attribute Hierarchy



Engineering Dependable AI Systems

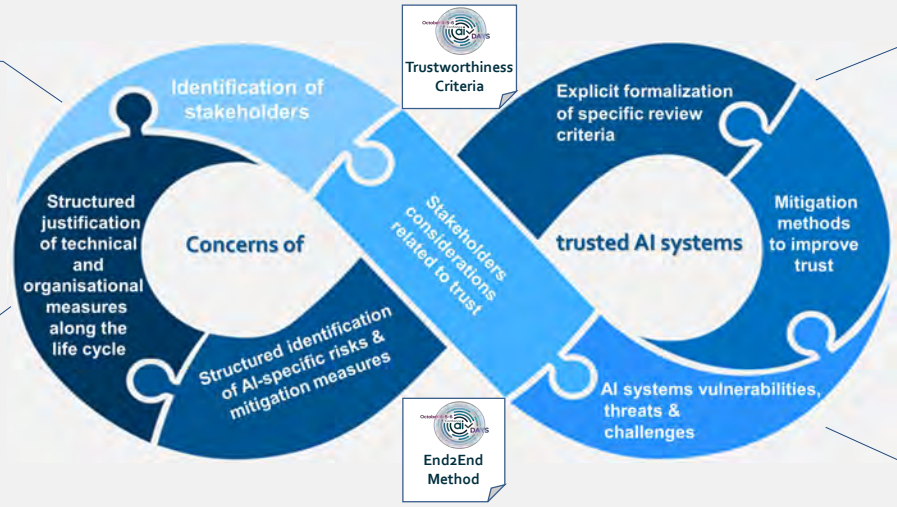
Morayo Adejouma, Christophe Alix, Loic Cantat, Eric Jenn, Juliette Mattioli, Boris Robert, Fabien Tschirhart, Jean-Luc Voirin

What are we trying to achieve?

Enable and support effectively the End2End Vision

7 key principles & Global approach

Build justifications and demonstrate trustworthiness



Implement and secure iterations, feedback and consistency

Assurance cases V&V strategy

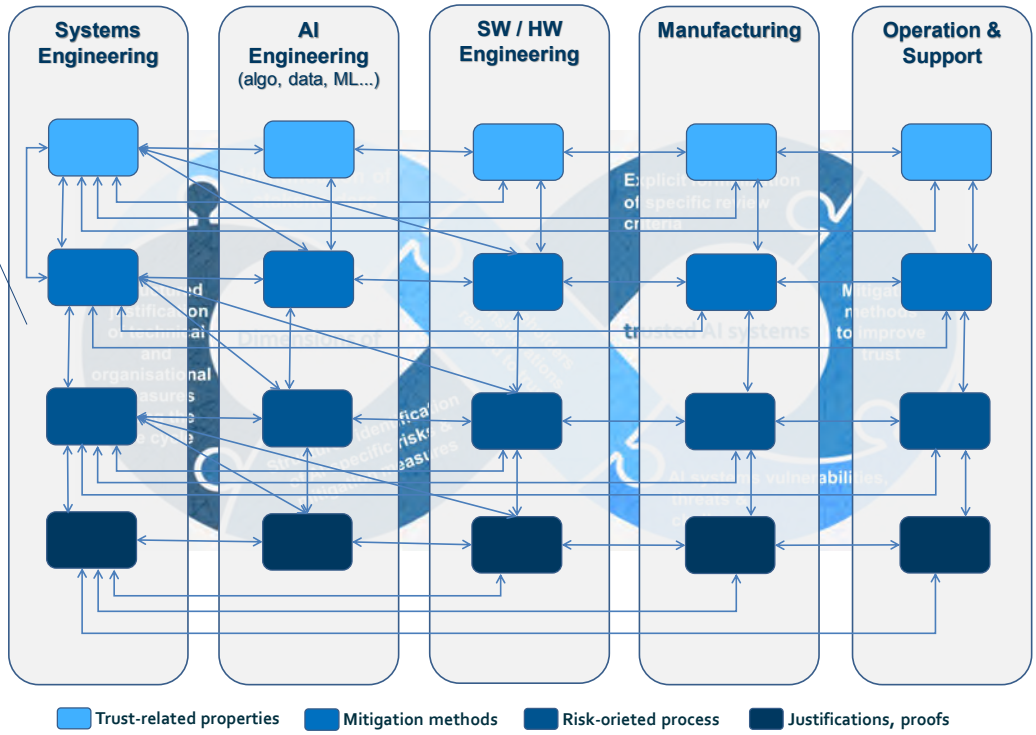
Define, assess, capitalise, analyse and improve measures

How can this be achieved?

Modelling environment

Implementing and securing concurrent and collaborative engineering

End2End use trustworthy environment



Ensuring an efficient feedback with the system in operation

AI Robustness & Monitoring Village

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Assurance Cases and V&V Strategy

Eric JENN [1,2]; Ramon CONEJO [1]; Vincent MUSSOT [1]; Florent CHENEVIER [1,2]

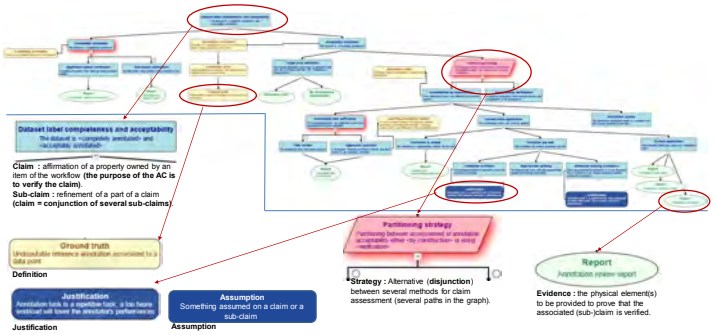
[1] IRT Saint Exupéry, France; [2] Thales, France

Abstract

The verification and validation of AI-based systems is an emerging area, where no widely adopted standard has yet been written. Therefore, a possible approach to ensure the safety of such systems is to characterize trustworthiness as a set of fundamental properties which shall then be proved to be satisfied.

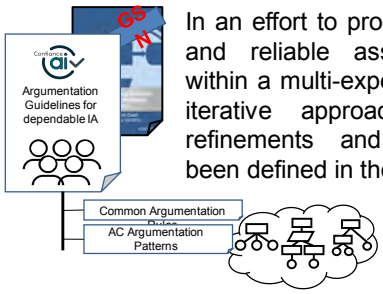
Assurance Cases

In Confiance.AI, we rely on the use of assurance cases to ensure and verify such properties.



Methodology

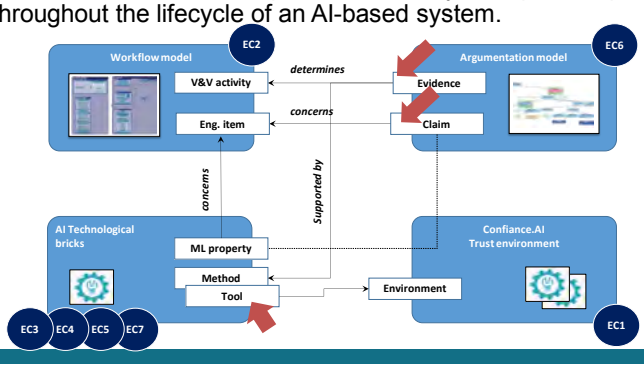
In an effort to produce consistent and reliable assurance cases within a multi-expertise group, an iterative approach based on refinements and reviews has been defined in the process.

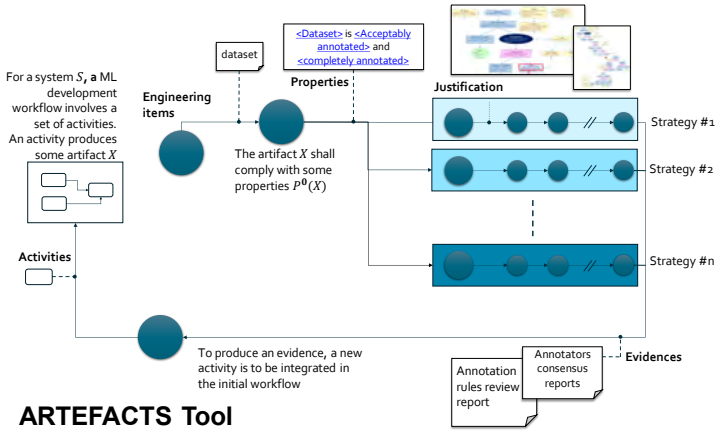


Assurance Case Examples - Data Engineering

Assurance Case and V&V cycle

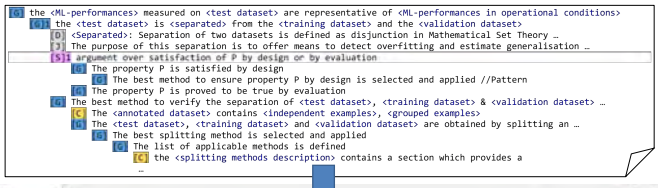
These ACs can provide a complete list of validation and verification activities required to satisfy a specific property throughout the lifecycle of an AI-based system.

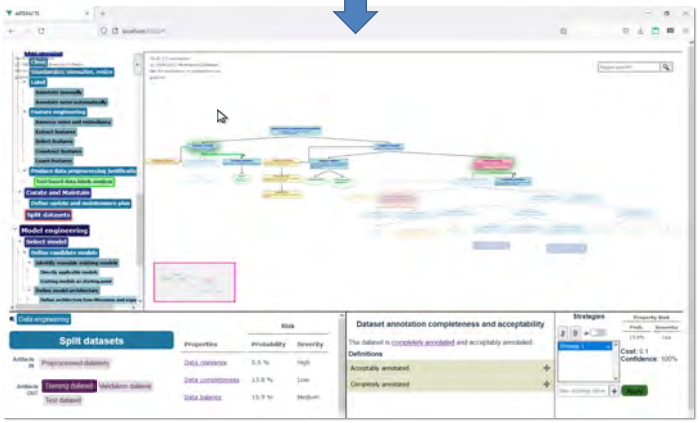




ARTEFACTS Tool

ARTEFACTS is a web application which allows the interaction between the workflow of an AI-based system and the textual ACs defined by the user. It also includes strategies selection, metrics estimation (cost, confidence, risk), and addresses other activities from Confiance.AI.





Conclusion and work in progress

Maturation of assurance cases are iterated over the Data Engineering activities. Once sufficiently evolved, with the expertise of different industry partners, it is expected to generate a first guideline to reliably produce assurance cases for AI-based systems. Furthermore, ARTEFACTS could evolve into a fully functional tool that could ease the completion and review of assurance cases and assist the V&V engineer.

End-2-end use of trustworthy environment

DE LA CHAPELLE Cyprien, FIQUET Ingrid, FOULLIARON Josquin

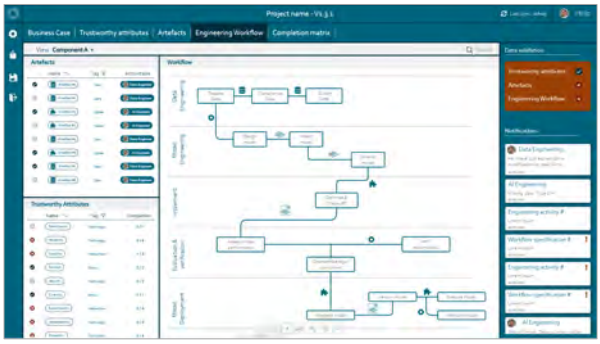
IRT SystemX, Sopra Steria

Design your *confiance.ai* engineering workflows

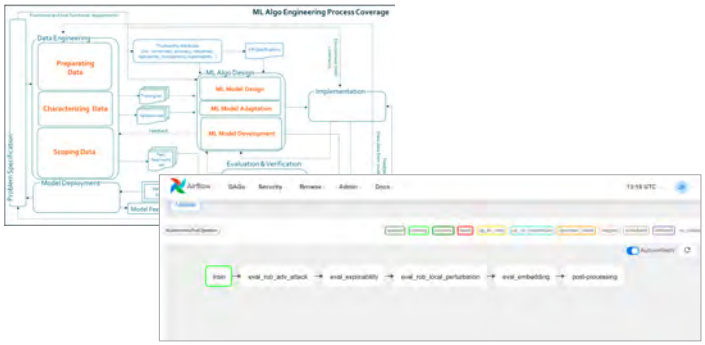
This initial step is fully assisted through the use of the « companion », a user-friendly web application

Features :
- multi-projects and multi-users management system
- collaborative design
- automatic generation of design documentation (business oriented)

User journey
1. business case analysis
2. trust attributes recommendation and configuration
3. artefacts creation and mapping to trust attributes
4. engineering workflows recommendation and configuration
5. checking of coverage mapping and trust attributes traceability



Run your *confiance.ai* engineering processes



This main step is carried out by technical teams to operate trust components within a single environment, share data, experiment and gather results

Features :
- data management and data processing platform
- interoperability services and SDK provided
- integrated IDE and process scheduler
- interactive guides for preparing and running processes
- allow parametrization of post-processing modules for KPI

Monitor your *confiance.ai* properties

Trust components generate results which are ingested in the « supervision panel » database. On top of this database, the « supervision panel » let the user customize its own Dashboards

- Features :
  - library of technical graph for performances assessment
  - library of KPI for trust attributes monitoring
  - assist to dashboards customization
  - availability of audit logs

Posters Village: Explainability tools and processes for understanding

Introduction to the themes of the village
Philippe Dejean

The domain of the explicability of models is one of the foundations of trust. Beyond the study of the usability of existing methods and algorithms for the use cases proposed by the partners of Confiance.AI, explainability is studied in all its forms and everywhere in the processes of data and model definition. We propose an overview of these studies around explainability by means of 9 posters that prepare the next interpretability works:

• *Transversal studies around explainability*
The explainability represent a small part in these work about efficient iterative dataset construction and continuous learning. The objective is to assess the usability and the pertinence of explainability in the different pipelines. Exploitation of synthetic data to improve domain coverage is also studied to evaluate models trained on real data. Four start-ups (Oktal-SE, Golaem, Jolibrain and Cervval) work together to build a dataset, this dataset should be the closest possible to the Scene understanding use case from Valeo. The synthetic dataset should represent a small part of the original dataset, a road circuit in a city with the shape of an 8. The Confiance.ai program will then assess if it is possible to use those synthetic data to evaluate a model trained on real data. A huge advantage is that synthetic data allow us to generate corner cases, or evaluate for example, how far can a pedestrian be detected by the model. The evaluation process has yet to be defined, thus the place of explainability is still unclear. Nevertheless, explainability is a tool to evaluate models, it will thus be used in the aforementioned process. For example, to compare explainability on real and synthetic data, to verify that the model detect the right things on synthetic data and so on.

• *Explainability: Methods and libraries*
The goal of these works is to evaluate existing toolboxes of explainability on the industrial use cases of Confiance IA. During last year, several technics and libraries (Xplique - Deel, Gems.AI - ANITI, XAI360 - IBM, Shap - Microsoft) have been evaluated. Each toolbox will be evaluated in terms of performance with different metrics of explainability, maintnability, adaptability and usability to understand which ones are the more suitable for industrial use case. In the continuity of these works, to determine the usage and the limits of each kind of explainability technics well known in the literature, the toolboxes of Captum (Facebook), Saliency & TCAV (Google), Partial dependence (Global explanation) will be evaluated on various kind of industrial data set: image classification, image detection, time series, tabular prediction, images feature extraction.

• *Regional Explaination for ML Models*
Some work aims to evaluate the technology of the startup AI-Vidence through several actions: participate to the state of the art about knowledge-based AI, test the technology on the use cases, and integrate the tools in Confiance.AI environment. The objectives are to test the startup tools on industrial cases of the program and enrich the explainability library of the program with knowledge-based tools.

• *Counterfactuals-based metrics for the evaluation of image classifiers*
This work was done in collaboration with the 3AI ANITI on new explainability and interpretability methods based on statistical (optimal transport theory) and logical (naturally explainable) approaches. Last work introduces new evaluation metrics for image classification models assessing the bias of a predictor. These metrics leverage causal counterfactuals approximated using Optimal Transport to bring information about where should the classifier focus for its predictions. The effected works consist in:

- Generating relevant counterfactual examples in high dimensional space (image) and in a multiclass context using Optimal Transport and conditional Wasserstein GAN
- Trying alternatives to optimal transport for the computation of counterfactuals (LDDMM, Wasserstein gradient flow)
- Finding two metrics relying on generated counterfactuals and existing feature attribution methods for the detection of spurious correlations in the learning of a classifier

- Leveraging the information provided by these metrics for the training of classifiers

• *Prototype-based models for explainability / Explaining objet detection: the case of Transformers architecture*
It aims to assess neural networks explainable by-design, while focusing its research on the vision field. This work will evaluate the performance of such models compared to classic ones and the explainability of those models compared to post-hoc explainability. Afterward, the pertinent tools will be integrated in the Confiance.AI environment and guidelines will be provided. After a review of the state of the art, two types of models were selected for implementation and/or evaluation: Transformers and Prototypes.

• *Explainable Unsupervised Anomaly Detection for Time Series*
This work is closely related to work on anomaly detection for time series. There are several objectives. First, is to summarize the existent and provide a reading grid for explainability in the anomaly detection field. Then we evaluate existing methods (from literature) and implemented methods through this reading grid and select and implement (or evaluate existing code of) promising anomaly detection methods explainable by-design (in the context of time series). Finally, we evaluate those methods on the compatible use cases, compare results with other implemented methods and present the result in a report and integrate pertinent methods to the Confiance.AI environment and provide associated guidelines.

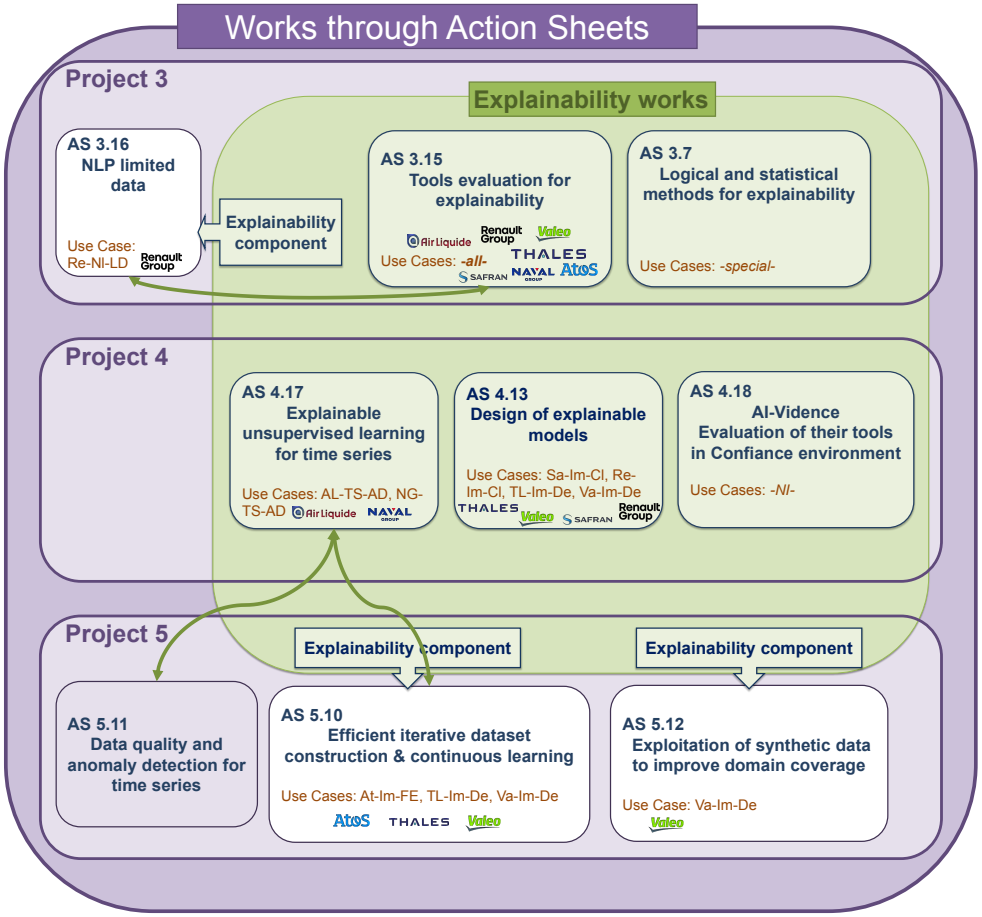
• *Explainability: State of the Art on explainability for NLP*
This year, as several works in Confiance.AI focuse on NLP use case -a new one in Confiance environment-, the explainability on NLP is studied. After an exhaustive state of the art, previous identified libraries and specific methods or toolbox are tested on a NLP model for specific domain while there is few annotated data in this domain.

• *Methodology for Trustworthy Natural Language Process Models with Limited Training Data*
The complete aim of these works around this use case is to improve the trust in NLP system decision. The purpose is to deal with the case of adaptation of a general language model to a specific application domain

Perspectives: From explainability to interpretability
The promising libraries identified during the benchmark have demonstrated their great flexibility to be used on different use cases. However, to choose the right parameters (hyper-parameters of explainability technics), the users shall have knowledge on the explainability technics itself. To help those who have no knowledge on explainability technics to deploy the methods or metrics on their particular use cases, recommendations and analytics tools are studied. The aim is to establish links between the explainability results and the operational context in terms of elements well known by the user to be easily understood by him. The profile of the latter must also be taken into account. This will form the basis of the interpretability work to be carried out next year.

Transversal studies around explainability

Ph. Dejean (1) – Th. Allouche (2) – A. Coppin (3) – C. Gardet (4) – A. Petit (4) – D. Petiteau (1) – A. Poche (1)
(1) IRT-StExupery – (2) ATOS – (3) Naval Group – (4) Sopra Steria



Project 3 - Characterization & qualification of trustworthy AI

Explainability works focus on explaining already trained models via post-hoc methods as well as metrics. Furthermore, this work is a first step toward tools to construct interpretation characteristics in the next batch. Thus, future works will focus on human profiles to adapt the usability of these tools to facilitate interpretation.

Project 4 - Design for Trustworthy AI @ algo, model and systems levels

Explainability works are included in the model design to generate inherently explainable models. It is based directly on real use cases, these works enabled the industrial partners to test all the results obtained in pseudo-operational environments and conditions.

Project 5 - Data, information and knowledge engineering for trusted AI

Explainability works focus on the data and dataset. Combining works on active learning, adaptive learning and transfer learning, these works allow a special attention to be paid to explicability tools through the behavior of data in model adjusting chain.

Use cases treated

THALES THALES LAS Image – Detection (TL-Im-De)	NAVAL GROUP NAVAL GROUP Times Series – Anomaly Detection (NG-TS-AD)	Renault Group RENAULT NLP – Limited Data (Re-NI-LD)	AIRBUS AIRBUS Tabular Data – Regression (Ai-TD-Re)	Air Liquide AIR LIQUIDE Tabular Data – Regression (AL-TD-R)
Valeo VALEO Image – Detection (Va-Im-De)	SAFRAN SAFRAN Image – Classification (Sa-Im-Cl)	RENAULT Image – Classification (Re-Im-Cl)	AtoS ATOS Image – Feature Extraction (At-Im-FE)	AIR LIQUIDE Time Series – Anomaly Detection (AL-TS-AD)

Explainability: Methods and libraries

Antonin Poché

IRT Saint Exupéry

In the literature

Taxonomy

- **When can the method be applied ?**
  - Post-hoc
  - By-design or intrinsic
- **Which kind of models ?**
  - Model-agnostic
  - Model-specific
- **How much information is needed ?**
  - Black-box
  - White-box
- **What is explained ?**
  - Decision (local)
  - Model (global)
  - Data
- **What is the format of the explanation ?**
  - Many different types
  - Cannot be listed, see examples below

Post-hoc methods types

- **Attributions**

'GradCAM' attributions of dog vs cat, from Ramprasaath et al (2019)

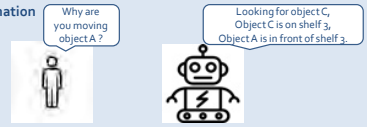
- **Example-based**

'Cole' example on MNIST, from Kenny et Keane (2019)

- **Model surrogate or rule-based**

Represent a neural network by a decision tree with the same behavior

- **Literal or oral explanation**



- **And many other types**

Features visualization from Xplique

Concepts from Xplique

Metrics

- **Expected properties of an explanation:**
  - Fidelity
  - Stability
  - Comprehensibility
  - Generalizability
  - Consistency

In the libraries

The three main libraries

- They have a wide range of methods and cover most of the implemented types of methods.

Xplique	Captum	AIX360
By **IRT Saint Exupéry** For tensorflow and others as black-boxes	By **Meta** For pytorch models	By **IBM** For tensorflow, pytorch and sklean models

- A library's implementation of a method may not be compatible with the model format (tensorflow, pytorch...), while the method is theoretically applicable.

Xplique - Attribution
15 implemented methods

Captum
- Features attribution
16 implemented methods
- Layers attribution
11 implemented methods
- Neurons attribution
9 implemented methods

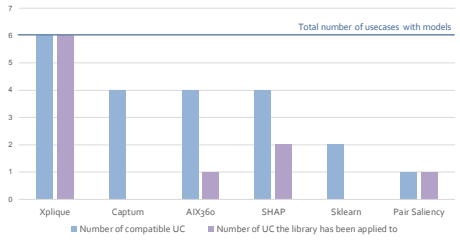
AIX360 -Attribution
2 implemented methods

Xplique - Example-based
Methods in development

Captum – Influential examples
6 implemented methods

AIX360 -Prototypes
1 implemented methods

Overview of tested libraries on the different use cases



AIX360 –Rule based
3 implemented methods

AIX360 – Literal explanation
1 implemented methods

Xplique - Feature visualization
1 implemented method

Xplique - Concepts
2 implemented methods

Captum - Concepts
4 implemented methods

Xplique - Metrics
- 3 Fidelity metrics
- 1 Stability metric
- 1 Representability metric
- 1 Consistencymetric

Captum – Metrics
- 1 Fidelity metric
- 1 Stability metric

AIX360 – Metrics
- 2 Fidelity metric

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec - Saclay, France

Regional Explaination for ML Models

Weituo Dai, David Cortés

Aividence, SoyHuCe

Context

Most business needs and most used algorithms themselves rely on 'segmentation'. Classical ML model explaination tools like SHAP, LIME, do address the local (decision level) or global scales (model level), but none of them approach the 'regional' scale (segment). Understanding by both the Data Scientist (DS) and Business Owner (BO) often relies on that specific scale (be it Operation Regime, Customer segment) still not dealt with.

Keywords

Regional Explaination, segmentation, clustering, SHAP, LIME, Decision Trees, Data Visualisation, Banzhaf, Shapley-Shubick, TreeRank, Ranking Forest.

Method

AntakIA methodology by Aividence aims at gaining a common understanding between the DS and the BO by building explainations of the models at a regional level. The main steps leading to Regional Explaination rely on 'dyadic' steps, implying simultaneously *the Values Space (VS) and Explainations Space (ES)* computed e.g. through Shapley values or other indexes :

- **DYADIC VISUALISATION** : *Visualize the VS and ES datasets at the same time* through dimensions reduction approaches (t-SNE, UMAP, PCA)
- **DYADIC EXPLORATION** : *Explore the simultaneously consistent zones of both spaces* with DS and BO
- **DYADIC SEGMENTATION** : *Define precisely a region, describe as simply as possible each region* in both spaces
- **DYADIC UNDERSTANDING** : *Make sure it makes sense* ! Through mutual explaination and understanding between the DS and BO, with complementary feature-wise analyses.

These steps are to be iterated *until all the VS is addressed*.

Result

On a simple simulated datasets with an explicitly 5-segment biased model (e.g. age below 25, or over 40 and man vs woman, etc.) , we have been able through this dyadic approach to reconstruct the relevant segments learnt by a standard black box Model (XGBoost), considering simultaneously the original values and model explainations.

Air Liquide Use Case

We have been using this dyadic approach for anomaly detection on time series :

- defining a more understandable VS (with signal procesing and ad hoc aggregated features)
- using unsupervised detection anomaly algorithms and SHAP to construct an ES.

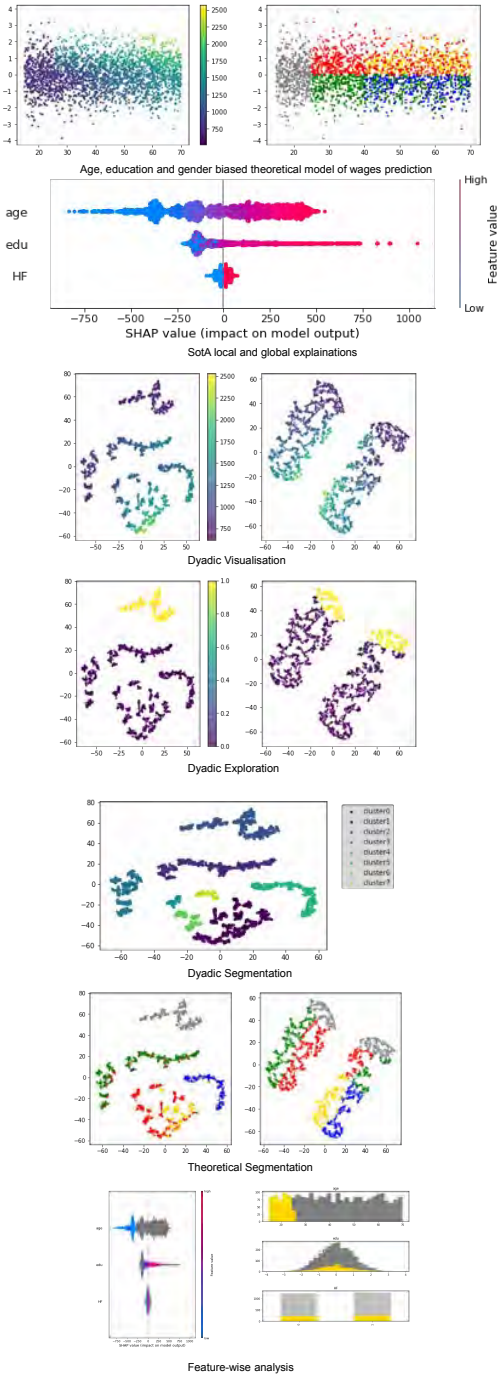
AntakIA is then used to find clusters of time series similar in values and explainations, and to challenge those classifications during reviews with Air Liquide experts, and Data Scientists.

Prospects and future Work

AntakIA methodology also encompasses the building of surrogate models so as to construct Explainable, then hopefully Certifiable AI by design.

We will be analysing the gain of using :

- other decision decision power indexes from games theory (Banzhaf, Shapley-Shubick, …)
- other specific tree-based explainations methods [P. Marquis]
- top performance surrogate models such as TreeRank algorithms [S. Clemençon]



Explainability: Tools and processes for understanding

Counterfactual-Based Metrics for the Evaluation of Image Classifiers

(Project: Logical and statistical methods for explanability)

Yannick PRUDENT[1,2,3] – David VIGOUROUX[1,2,3]

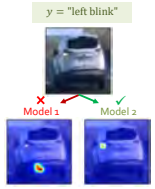
[1] IRT Saint-Exupéry – [2] IRT SystemX – [3] ANITI

Quantitative Evaluation of the Semantic of a Classifier

Idea: We define **two metrics** quantifying the **ability of a classifier to focus on the correct semantic features** when doing its predictions. We assume that **the correct features should correspond to the pixels of the input image that change the most when changing its label** thanks to a **counterfactual generation process**.

Feature Attribution to detect model biases

[Def]: Feature Attribution methods generate **heatmaps** highlighting the regions involved in the decision of a classifier.



Feature Attribution maps to detect **spurious correlations**

Issue: **Model 1** and **Model 2** both predicted the **left blink**, but clearly the Model 1 used irrelevant information for its prediction (**spurious correlations**).

Q: **How to make sure — by a quantitative evaluation — that the classifier is taking its decisions for the good reasons ?**

A:

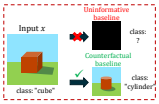
(1) We generate using a GAN a **counterfactual counterpart** of the input image by answering the question: *"What would the input image have looked like if it had been from another class ?"*

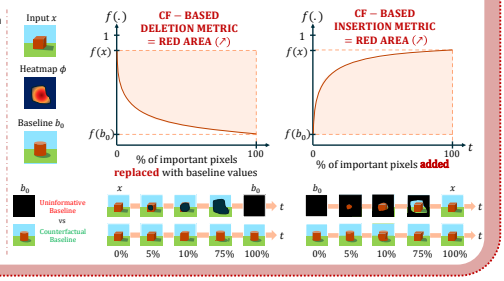
(2) We evaluate the **impact on the classifier's prediction of some perturbations** over **the most important pixels** of the input image (according to the heatmap) **going towards the counterfactual image.**

⇒ This perturbation should have a high impact if the model is focusing on the right features!

Counterfactual-based metrics

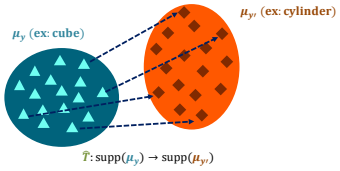
Principle: Usually, feature attribution maps are evaluated doing some **perturbations** of the input image according to the heatmap and going towards a **baseline image expressing "missingness"**. Our metrics follow the same principle but **use a counterfactual image** instead of the usual **uninformative baseline**.





(1) Computation of Counterfactuals using Optimal Transport

Optimal Transport

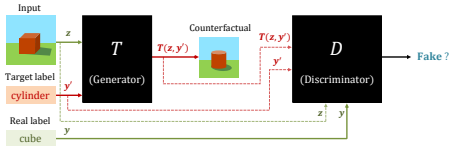


$\hat{T}: \text{supp}(\mu_y) \to \text{supp}(\mu_{y'})$

Monge Optimal Transport's problem:

$$\hat{T} \in \underset{T,\, T\#\mu_y = \mu_{y'}}{\text{argmin}} \left\{ \int_{z \in \text{supp}(\mu_y)} \|z - T(z)\|_2^2 \, d\mu_y \right\}$$

CFGAN

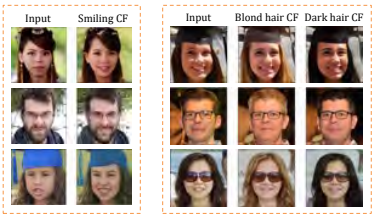


CFGAN Objective (in latent space $\mathcal{Z}$):

$$\min_T \max_D \sum_{y'=1}^{|\mathcal{Y}|} \mathbb{E}\left[D_{y'}(Z) \mid Y = y'\right] - \mathbb{E}\left[D_{y'}\left(T(Z, s(y'))\right)\right]$$
$$- \lambda_{gp} \cdot \mathbb{E}\left[\left(\| \nabla_Z D_{y'}(Z) \|_2 - 1\right)^2\right]$$
$$+ \lambda_{tr} \cdot \mathbb{E}\left[\| T(Z, s(y')) - Z \|_2^2\right]$$

Counterfactual examples

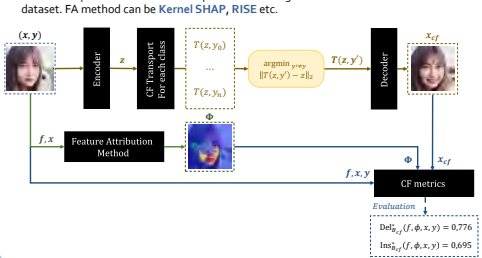
FFHQ dataset (resp. **smile** and **hair color** classes)



(2) Evaluation of Classifier using Counterfactual Metrics

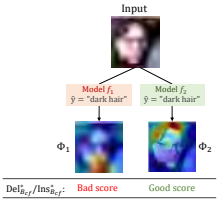
Global Pipeline

Evaluation process overview. Example on the smiling classification task of the FFHQ dataset. FA method can be **Kernel SHAP**, **RISE** etc.



Uses of the CF metrics

1. **Global** or **per sample** evaluation score for **model selection:**



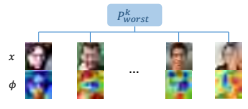
2. **Given a model f**, find the samples for which the **model takes the right decision** but for the **wrong reasons:**

Samples correctly predicted

$$P = \{(x_i, y_i) \in \mathcal{D} \mid f(x_i) = y_i\}$$

Top-k worst samples according to CF Deletion metric

$$P^k_{worst} = \text{argmin}_{P' \subset P, |P'|=k} \sum_{(x,y) \in P'} \text{Del}^{\pi}_{R_{cf}}(f, \phi(x, y), x, y)$$



Future Work

- Integrate a **differentiable version of counterfactual metrics** in the **training loss** of a classifier

⇒ We learn a classification task while ensuring that the classifier is focusing on the right features for its predictions.

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Prototype-based models for explainability

Elodie Guasch

Airbus Protect

Design for Trustworthy AI @ algo, model and systems levels

Prototype-based models: explainability through case-based reasoning

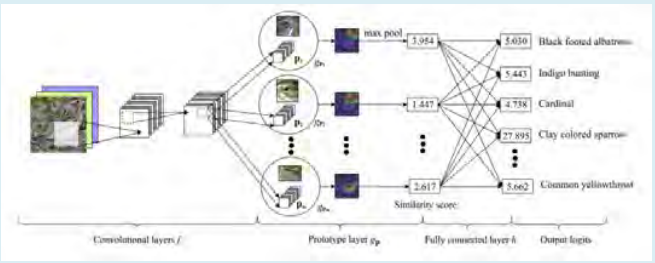
Prototype-based models are part of the so-called "explainable by design" models. Methods using the notion of prototypes provide an explanation of the model's reasoning process by approximating the images to be explained with typical examples, the prototypes, learned from the training dataset. The definition of what a prototype is varies from paper to paper.

These models are used for classification tasks and have been tested on image classification for this study. They have the advantage of providing local and/or global explanations accessible to a wide audience and easily interpretable.

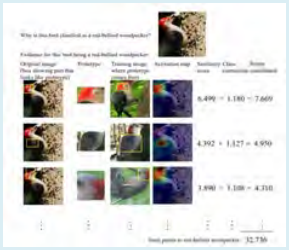
ProtoPNet

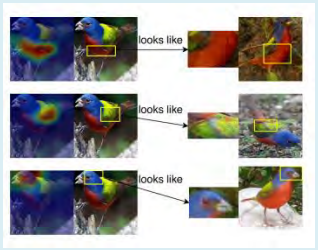
ProtoPNet, an architecture based on a convolutional neural network and a prototypical layer, is introduced by Chan and al in [1]. They define a prototype as a latent representation of a training image patch.

Their model computes a similarity score between learned prototypes and patches from the image being processed, thanks to the prototypical layer. Inference is obtained by processing the smallest distance to patch prototypes with a classifier.



This method provides a **local explanation** for the image being processed by the model.



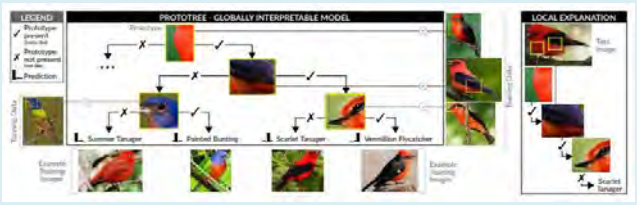


[1] Chaofan Chen et al. "This Looks Like That: Deep Learning for Interpretable Image Recognition". In: arXiv:1806.10574 [cs, stat] (Dec. 28, 2019). arXiv: 1806.10574. URL: http://arxiv.org/abs/1806.10574.

ProtoTree

Nauta and al. defined an architecture called ProtoTree in [2], composed of a convolutional neural network followed by a binary decision tree structure. The notion of a prototype is the same as for ProtoPNet but each learned prototype corresponds to a node of the decision tree.

Inference is obtained by comparing latent features of the test image with learned prototypes at each node, which determine the routing through the tree.



This approach provides both **local and global explanation** for the model reasoning.

[2] Meike Nauta, Ron van Bree, and Christin Seifert. "Neural Prototype Trees for Interpretable Fine-grained Image Recognition". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, pp. 14933–14943

Evaluation of prototype-based models

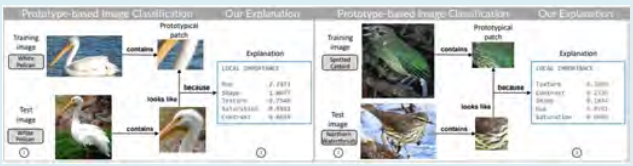
Prototype-based models are mostly qualitatively evaluated regarding explainability.

- **Explainability / performance trade-off :** Both ProtoPNet and ProtoTree present good trade-offs between accuracy and explainability

Base	Baseline	ProtoPNet	ProtoTree
ResNet34	82.3	79.2	82.2

Mean accuracy on bird images of CUB dataset (224x224) from [1] and [2]

- **Exploitation of the explanation :** Learned prototype can be used to detect good and bad behaviors of the model by looking at the areas activated on test images and their similarity with learned prototypes
- **Explanation of learned prototypes :** As shown in [3], prototypes can be further explained through experiments on hue, contrast, shape, saturation and texture. This allows to better understand similarity between image patches and learned prototypes, and to avoid incorrect interpretations.



[3] Meike Nauta, Annemarie Jutte, Jesper Provoost, and Christin Seifert. This looks like that, because ... explaining prototypes for interpretable image recognition, 2020

Explaining object detection : the case of Transformers architecture

Baptiste Abeloos, Stéphane Herbin

ONERA / DTIS, Université Paris-Saclay

Design for Trustworthy AI @ algo, model and systems levels

Introduction

Context: Explanation by design for object detection task

Objective:

1. Explain model prediction for object localization
2. Evaluate the explanation method

Model: Attentional models (DETR)

Object detection with DETR [1]

Object detection sub-tasks:

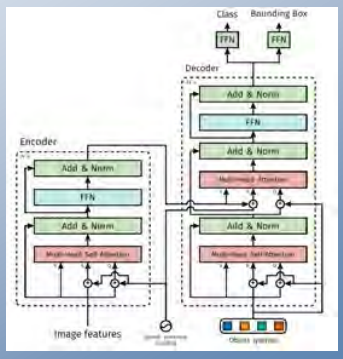
1. Existence
2. Location
3. Category

DETR:

- One stage approach
- Encoder/decoder
- Attentional model

Attentional maps computed as

$$A = \text{softmax}\left(\frac{Q.K^T}{\sqrt{d}}\right)$$



[1] N. Carion et al. "End-to-End Object Detection with Transformers". In: arXiv:2005.12872 [cs, stat] (May. 2020). arXiv:2005.12872. URL: https://arxiv.org/abs/2005.12872

Explanation with transformers [2]

- The nature of the explanation is a heatmap indicating a relevancy score **R** at each image location
- **R** is initialized and updated by a forward pass using a specific attention-based rule for each layer
- At each layer, attention maps are weight-averaged over the heads **h** :

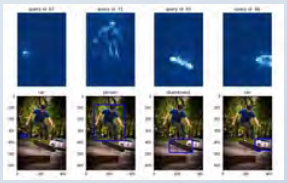
$$\bar{A} = E_h\left[\frac{\partial y}{\partial A} \odot A\right]$$

where y is an output variable and A is an attentional map

- The average $\bar{A}$ is then used to update **R**.

[2] H. Chefer et al. "Generic Attention-model Explainability for Interpreting Bi-Modal and Encoder-Decoder Transformers". In: arXiv:2103.15679 [cs, stat] (March. 2021). arXiv:2103.15679. URL: https://arxiv.org/abs/2103.15679

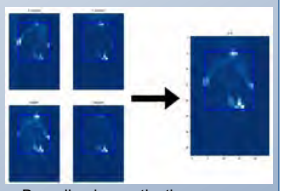
Can we explain the prediction for object location ?



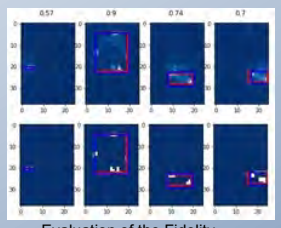
Class activation maps

- Application on the COCO dataset
- Using a public pre-trained DETR model

- Bounding box prediction : prediction of x center, y center, width, height
- The bounding box activation maps are summed to form a single localization activation map



Bounding box activation maps

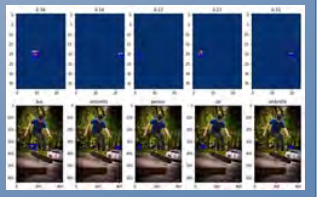


Evaluation of the Fidelity

- Assumption : the ends of the object are more activated when predicting the bounding box coordinates
- Signal selected using Otsu method
- Evaluation: IoU between the minimal bounding box encompassing the signal and the prediction (Fidelity)

Evaluation on the COCO test set

Detection threshold	0,6	0,8	0,9
IoU (Fidelity)	0,46	0,49	0,52



Fidelity decreases for small objects

→ Better confidence in detection implies better confidence in explanation
→ Better explanation score for large detected objects

Perspective and future work:

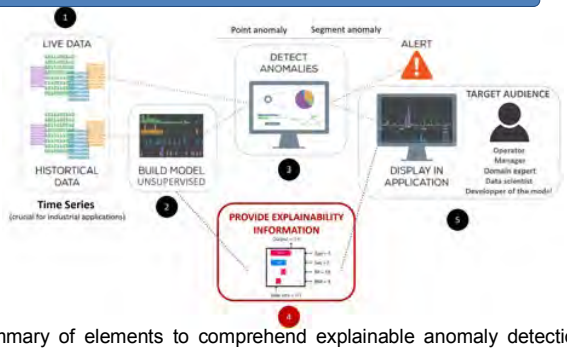
- Can we anticipate good or bad behavior ?
- Application on other dataset/use cases

Explainable Unsupervised Anomaly Detection for Time Series

Maxime Desbois [1], Mathilde Guillemot [2], Antonin Poché[3]

[1]IRT SystemX, [2]Air Liquide, [3]IRT Saint Exupéry

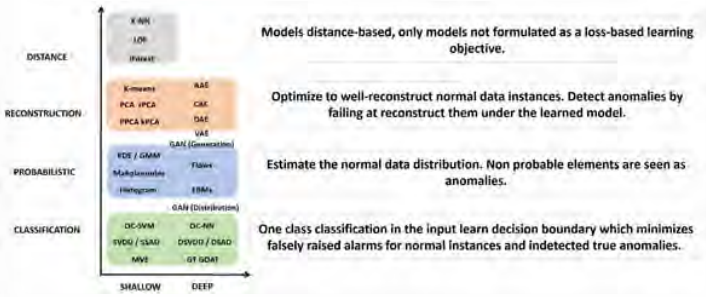
Motivation



A summary of elements to comprehend explainable anomaly detection by design for time series.

Anomaly detection methods

Ruff and al. [1] have proposed a taxonomy for anomaly detection methods along two dimensions : the **model depth** and the **model type**.



Explainability and times series

Explainability **does not have a precise** and universally accepted **definition**. The characterization of explainability is not addressed here but some **important questions** are raised :

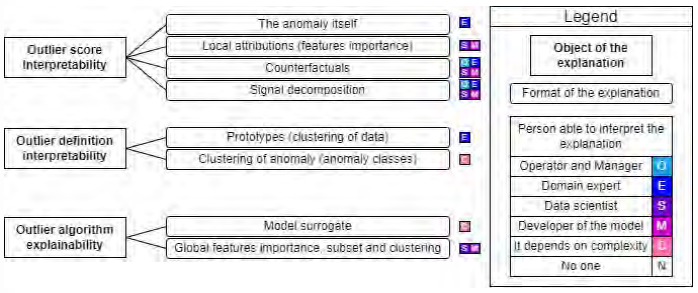
- **What** do we want to explain? The model itself or the outputs?
- **How** do we want to display the explanation?
- **Who** is the target audience?

⇒ There is no generic solution to these questions, answers must be designed according to the use case

Working with **time series brings other challenges** to overcome in comparison with other data format such as images :

- Explanations are harder to understand,
- Experts are always needed,
- Time series representation is still an active research area.

Explanation type and anomaly detection



Explanation are provided to everyone but not **interpretable** by all.

Example of explanation type

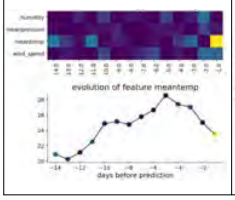
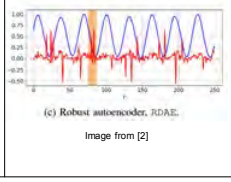
Local attribution	Signal decomposition	Model surrogate
		

Image from [2]

Challenges and perspectives

- Anomaly detection and explainability **performances hard to evaluate** : most industrial use cases are **unlabeled.**
- **Comparison** is problematic : methods and explanation types are too different.
- Time series anomalies and explanations **visualisations** are not trivial.
- A **domain expert** must be highly involved and available in the explainabilitty framework.

For time series, there are **few papers** in the literature on anomaly detection explainable by design. Although some methods seem promising such as **robust autoencoders** [2] or attention based models, adapting **graph attention network** [3] or **transformers** [4].

Bibliography

[1] Lukas Ruff et al. "A Unifying Review of Deep and Shallow Anomaly Detection". In: CoRR abs/2009.11732 (2020). arXiv: 2009.11732. URL: https://arxiv.org/abs/ 2009.11732.

[2] Tung Kieu et al. Robust and Explainable Autoencoders for Unsupervised Time Series Outlier Detection—Extended Version. Number: arXiv:2204.03341 arXiv:2204.03341 [cs]. Apr. 2022. URL: http://arxiv.org/abs/2204.03341 (visited on 06/24/2022).

[3] Hang Zhao et al. "Multivariate Time-Series Anomaly Detection via Graph Attention Network". In: 2020 IEEE International Conference on Data Mining (ICDM). ISSN: 2374- 8486. Nov. 2020, pp. 841–850. DOI: 10.1109/ICDM50108.2020.00093.

[4] Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. Tech. rep. arXiv:2201.07284. arXiv:2201.07284 [cs] type: article. arXiv, May 2022. URL: http://arxiv.org/abs/ 2201.07284 (visited on 06/01/2022).

Explainability : State of the Art on explainability for NLP

Alice PETIT[1], Antoine COPPIN[2], Caroline GARDET[1]

[1] sopra steria [2] NAVAL GROUP

Explainability methods in general seek to explain results or behaviors of a model. *In NLP typically methods aim to find the* ***most important words*** *from which a model made a prediction, or they seek to explain the behavior of a model, i.e.* ***what it has learned or how it processes information***.

Depending on ***how*** *these methods extract elements of explainability and in what goals, they are classified in 4 categories.*
"Local" *if they try to explain a single results*
"Global" *if they try to explain the task independently of any inputs.*
"Post-Hoc" *if they are based on already trained models and require additional operations,*
"Self-Explaining" *if elements for explainability come directly from the model.*

Post-Hoc

Local

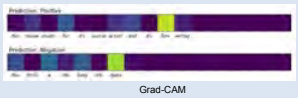
- **Pertubation based methods**: locally perturb an input and simulate the results of the model on these perturbed inputs using a simple and explainable model.

Ex : Leave-one-out, Occlusion, LIME, SHAP, SocRAT, Anchors

Example of results given by SHAP library for sentiment analysis *https://shap.readthedocs.io*

- **Back Propagation methods & Gradient based methods**: propagate a type of information from the last layer to the first layer of a neural network according to a calculation that depends on the method.

Ex : Layerwise Relevance Prop., DeepLIFT. Integrated Gradient, REAT, GradCAM, Gradient-Based Feature Attribution, Gradient X Hidden States, Gradient X Activation



Grad-CAM

- **Rationales based methods**: based on rationales, or significant pieces of sentences, and on the explanations associated with them
Ex : Protodash

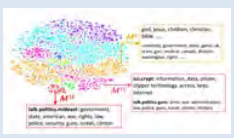
Global

- **Probing :** Based on the simulation with a simpler model of a relation between inputs and outut of a model

Ex : does the model learnt the length of sentences?

- **Profweight :** transfer information from a pre-trained deep neural network that has a high test accuracy to a simpler interpretable model or a very shallow network of low complexity and a priori low test accuracy

- **Taxonomy Induction**

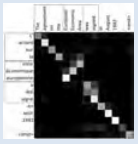


Self-Explaining

Local

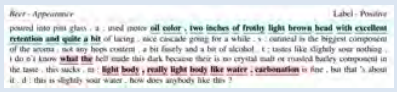
- **Attention based methods**: exclusive to Transformers like models, they are using attention weights to find important words

Ex : Attention weights, Cross Attention weights, Attention X Norm, Gradient X Attention, NICT Kyoto, Dual-stage attention mechanism



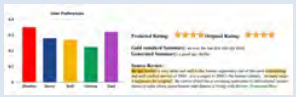
Attention weights for translation task

- **Rationales based methods**: based on rationales, or significant pieces of sentences, and on the explanations associated with them
Human Rationales as Attribution Priors, Interpretable Neural Predictions with Differentiable Binary Variable, TED, CAR, CREX, MARTA

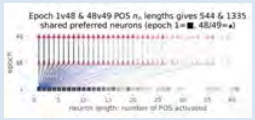


CAR on sentiment analysis and classification task

- **Other**: Epar, EANDC, Multi-hop Inference Explanation Regeneration, Self-explaining structure, J3R, TX-Ray, Transformer Interpretability Beyond Attention Visualization, Cross-Domain Transfer of Generative Explanations, CPM



J3R



TX-RAY

Global

- **Proto-Trex** : Transformers based on prototypes and closest neighbors for explanation

- **Reinforcement Learning**

- **HEIDL**

State $= (w_i, a_{i-1})$	Negated	¬ Negated	$\pi^*(s_i, a_i)$
(this, ¬ Negated)	0.0114	0.0502	¬ Negated
(beautiful, ¬ Negated)	0.0081	0.0779	¬ Negated
(movie, ¬ Negated)	0.0039	0.0506	¬ Negated
(isn't, ¬ Negated)	0.0700	0.0456	Negated
(good, Negated)	0.0578	0.0322	Negated
(but, Negated)	0.0120	0.0365	¬ Negated
(fantastic, ¬ Negated)	−0.0181	0.1708	¬ Negated

Metrics

- **How well an explanation method really represents how the model works? :** Faithfulness, Fidelity, Infidelity, Credibility Score
- **How well an explanation is understandable? :** Comprehensibility, Informativeness
- **How robust/sensitive is an explanation? :** Local Lipschitz, Sensitivity, Stability, Trust Score, Transferability
- **How to go further? :** Trustworthiness, Confidence, Causality

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Explainability: Tools and processes for understanding

Methodology for Trustworthy Natural Language Process Models with Limited Training Data
A State Of The Art

R. Besançon(1), O. Ferret(1), V.Feuillard(2), C. Gardet (3), E. Gherbi(4), L. Schott(4), F.P. Servant(2), J. Tourille(1), A. Uyanik(2)

(1) list (2) RENAULT (3) sopra steria (4) SystemX

Confiance.ai - NLP with limited training data

CONTEXT: Huge NLP Models

GPT3 · BLOOM · DeepMind Gopher · NVIDIA Megatron

- **Always Bigger Transformers Models**: Nowadays, NLP models are based on multitasks transformer models learned from all possible textual information with hundred of billions of parameters.
- **For specific domain with limited training data sets the models are difficult to qualify** even if some techniques are developed for an adaptation (ex: fine-tuning).

OBJECTIVE: Learn a trustworthy NLP Model

- ***How can I develop trustworthy NLP models learned with low data that will generalize on a large amount of new verbatims for getting meaningful insights?***

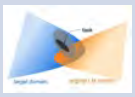


Illustration of data distribution (from [Logeswaran et al., 2019]) Task data is comprised of an observable task distribution sampled from a wider distribution (light grey ellipsis) within an even larger target domain, which is not necessarily one of the domains included in the original LM pretraining domain)

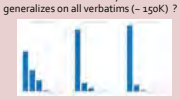
Example (Renault Google Reviews Use case)

– 700 google reviews with annotated aspect based and sentimental information

Can I learn a trustworthy model that generalizes on all verbatims (– 150K) ?

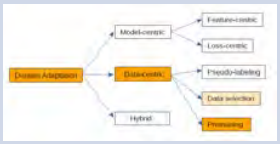
Verbatims

labels



Domain Adaptation

- This is the **adaptation of a Pretrained Language Model (PLM) to a specific domain and/or a specific task without exploiting annotated data.**
- **Different approaches** are possible (see Ramponi and Plank, 2020)



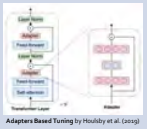
Taxonomy of DA methods (from Ramponi and Plank, 2020)

  - **Domain Adaptative Pretraining** (see Gururangan et al., 2020)
    - **Corpus Based Methods:**
      - continue pretraining with the task dataset: Task Adaptative PreTraining, **TAPT,** or
      - continue pretraining with similar dataset: Domain Adaptative PreTraining **DAPT**
    - **Vocabulary Based Methods**
      Add in-domain vocabulary with initial embeddings and update new embeddings by new pretraining and/or finetuning (regularized) and/or new layers in the network (see Hong et al., 2021 ; Tai et al., 2020, Yao et al., 2021)
  - Most of these methods rely on **Mask Language Models** that **require important computational resources. Adapters,** i.e. new layers in the PLM network **can be used to reduce the computational cost** (see Chronopoulou et al., 2021)
- **Other approach: Domain Knowledge Injection**
  An alternative is to explicitly inject into PLM knowledge coming from a structured Knowledge Base (KB). From a general perspective, the following **stages** need to be considered:
  - **Kind of knowledge** (source, granularity) **and its initial form** (ex: Resource Description Framework or Knowledge Graph),
  - **its exploitation into the PLM,**
  - **When and how the injection are performed** (ex: during/after pretraining or finetuning or inference; ex: by adding to the MLM loss an entity related loss).

A focus will be done on **Domain Adaptation for ABSA tasks** and on **PLM through vocabulary extension**.

Fine Tuning Transformers LM

- This consists in **updating a PLM network considering the task data set** by Google BERT
- **Different strategies** for parameter efficient fine tuning
  - **Fine-Tuning with Compression:**
    identify the parts in the network which removal is less damaging to performance and remove them (ex parameters and/or structure blocks).
  - **Adapters Based Tuning :**
    add a small set of parameters at every layer of the PLM and learn it with the task dataset.



Adapters Based Tuning by Houlsby et al. (2019)

  - **Bias-Term Fine Tuning :**
    limit fine-tuning to the bias terms of the PLM plus the parameters of the task-classification layer.

Adapter-based and BitFit finetuning generally achieve better results when the training set is small (full fine-tuning catches up its performance with more data).

For small dataset there are more optimization instabilities to be considered. Methods for overcoming these issues mainly consist in **applying strategies with multiple starting trials** (see Dodge et al., 2020) or by **considering a regularization term** (see Zhu et al 2020).

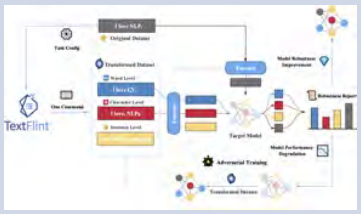
- **Aspect Based Sentiment Analysis (ABSA) with BERT : ABSA requires various tasks**: identification of the **aspect category, opinion term,** and **sentiment polarity**.

Task Finetuning and **Domain Adaptatio**n can be used. Main BERT approaches (adapted to small dataset) are: **BERT-SPC, BERT-ADA, BERT-Lsee, BERT-AEN, BERT-RRC**

A focus will be done on **parameters efficient fine-tuning with Adapters** for classification tasks (including **ABSA**).

Qualification of NLP Models

- The **qualification of a NLP model** can be assessed trough its **performance** (usual classification scores), its **robustness** (the sensitivity to text change) and its **explainability** (highlighting important words which contributes to a task).
- The **performance** of a NLP model for a Task (ex: Classification and/or ABSA Tasks) mainly relies in developing suitable strategies based on **Domain Adaptation** and/or **FineTuning**.
- The concept of **robustness** includes :
  - **Adversarial attacks** (used for image) are being adapted to the context of NLP. It **the attacks** can be at different levels: **embedding, character, word, sentence**.
  - **Model enhancement methods for the defense** :
    - **Adversarial Training**: collect adversarial examples by attacking a target model and fine-tune the model on the augmented dataset.
    - **Certified Robustness Defense**: train the model to provide an upper bound on the worst-case loss of perturbations (Interval Bound Propagation).
    - **Enhance Input Representation** : ex : correction loss mapping embeddings of a misspelled word closed to the embeddings of their correctly variants.



Example of Textflint (https://github.com/textflint/textflint)
A multilingual robustness evaluation platform for natural language processing

- **Methods for explainability** can be divided in two categories,
  - **Post-Hoc**: additional operation after the prediction
  - **Explainable-by-design**: explanation at the prediction

These methods can be **local** or **global**.
We refer to the Action Sheet 3.15.

A focus will be done on **identifying the strategies of attacks allowing the robustness assessment** of NLP models.

- ***Trustworthy strategies according to various typologies of problems will be proposed. These will combine Domain Adaptation and/or FineTuning and/or robust training.***
- ***For trustworthiness comparison, the notion of performance (with usual statistical metrics), of robustness (with adversarial attacks), and explainability will have to be adapted to NLP models.***

ROADMAP

2022												2023											
01	02	03	04	05	06	07	08	09	10	11	12	01	02	03	04	05	06	07	08	09	10	11	12
			State Of The Art																		Report		
								Python Functions i)				Python Functions ii)						Python Functions iii)					
								Typologies de Pblm							Illustrations ii)						Illustrations iii)		

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec - Saclay, France

Confiance.ai · FRANCE 2030

Posters Village: Robust AI and monitoring

Introduction to the themes of the village

Hatem Hajri, Fateh Kaakai

Robustness to outliers is an essential property of AI trustworthiness to ensure that an invalid input data will not lead to an unsafe state of the system. Robustness can be reached “by-design” and it can also be monitored by a specific component, the monitor, running in parallel to the AI model. Therefore, robustness and monitoring are two very related topics in the lifecycle of an AI product. In the Robustness & Monitoring Village, we present methods and tools that are already or will be integrated Confiance.ai environment.

Robustness

The adversarial examples are spectacular illustrations of the lack of robustness of some AI systems. If an invisibly modified picture of a panda can be recognized as a picture of gibbon or if a tagged stop sign is recognized as a 30mph sign (these are two well-known example of adversaria attacks), it is impossible to trust AI. Of course, this kind of attacks is a scientific demonstration of this weakness of data-based AI. They are the result of malicious computations that cannot be generated by pure randomness in the real life. At any rate, it instils the doubt about the general robustness of AI. If the issue of the macroscopic coverage of the operational domain can be solved by gathering a big enough training data base, is it possible to envisage all the microscopic variations around the training examples. In engineering of automatized systems, robustness is a very concrete feature: how does the system behave when itis pulled out of its nominal state? For an AI-based perception system, is it possible to guarantee that the classification will stay the same if the lightning changes? If a panda becomes a gibbon with only a small percentage of changed pixels, will a red traffic light become green if the sky is grey? The panda adversarial example also questioned the meaning of the confidence score that are given with the classification decision. The confidence score in the gibbon classification was much higher than the score of the original panda classification. A wrong classification with a very low score would have been acceptable. It is another highlight of the lack of robustness. It stresses the importance of the estimation of the imprecision on decisions taken. Classical engineered systems are able to evaluate the accuracy margin of their computation. The decision process takes this margin into consideration to adapt its conclusion. To become part of a trustworthy system, AI should be able to evaluate the accuracy of its conclusions.

Online Monitoring

The main objective of the online monitoring of AI models is to detect any deviation of the AI component deployed in operation from the specified expected behavior or from a predefined set of safety operational properties. A product has been developed using AI technologies, and it should demonstrate that the AI model can perform its prediction over its entire Operation Design Domain (ODD) with an accuracy of 99.9 % and that this accuracy is maintained over time in operation. Let’s assume that after a full training phase, the model’s performance does not exceed 99 % of correct predictions, it implies that 10 failures may statistically occur over the reference period (1000 hours) when only one failure would have been tolerated. This situation is unacceptable from a product safety point of view. The deployment of a monitoring component operating in parallel with the AI model (online monitoring as depicted in Figure 1) is a concrete way of managing this type of residual risk induced by a model for which it is not possible or feasible to formally demonstrate the achievement of the performance/safety objectives resulting from the system analyses. Online monitoring is a safety architectural pattern that is well known to operational safety engineers, but it had to be adapted to AI technologies. The work performed in confiance.ai defines an innovative engineering method to develop and verify online monitoring components that combine several monitoring timescales: a monitoring of the AI-based product at present time, a monitoring on a configurable time window of the near past, a monitoring on a configurable time window of the near future. These three (3) monitoring scales complement each other to ensure a high rate of detection of failures that could occur in operational conditions when the AI model is in production.

Monitoring relies on monitoring functions that will detect inconsistence in the inputs and/or product output such as (but not limited to), in the case of images:

- *Standard Defocus (Out of focus) Blur Detection*: In optics, defocus is the aberration in which an image is simply out of focus.
- *Standard Motion Blur Detection*: Motion blur is the apparent streaking of moving objects in a photograph.
- *Standard Brightness Detection*: This function aims at extracting the degree of brightness from an image and raising an alarm if this degree of brightness can impact the prediction of the model on this image.

For the evaluation of the monitor function, we need to assess the ability of the monitoring function to detect anomalies on input images. These anomalies correspond to camera problems encountered by the use case provider.

October 4-5-6 Confiance ai DAYS

Robust AI & Monitoring

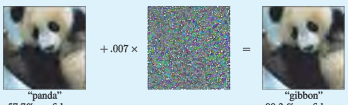
Robustness by design with 1-Lipschitz networks

Corentin FRIEDRICH, Thibaut BOISSIN, Franck MAMALET

IRT Saint-Exupéry

1-Lipschitz neural networks

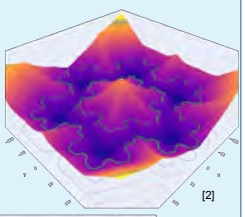
Motivation: standard networks are not robust to adversarial perturbations, i.e. a small perturbation on an input can yield a large change in the output, causing misclassification (Fig. 1).



Adversarial attack [1]

What are 1-Lipschitz networks?

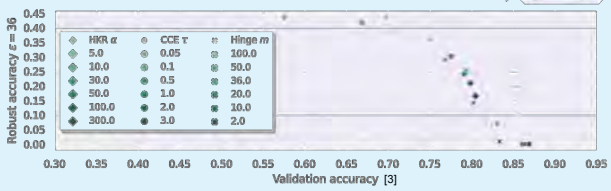
- Lipschitz constant to bound perturbations (Fig. 2):
  $$\|f(x+\varepsilon) - f(x)\| \leq L\|\varepsilon\|$$
- All layers are constrained with Lipschitz constant of $L = 1$



[2]

Why 1-Lipschitz networks?

- Handle **accuracy/robustness trade-off**
  (loss is the master key, see Fig. 3)
- Provide **robustness guarantees** (in L_2 norm)
- **No loss of expressivity** (Fig. 2) compared to standard classification networks



Open-source ready-to-use library

DEEL LIP
LIPSCHITZ KERAS LAYERS

- **Open-source library**, developed by DEEL program, available on PyPI and Github
- Full documentation + examples online https://github.com/deel-ai/deel-lip

- **Easy-to-use** library based on TensorFlow/Keras
  (no prerequisite to train a Lipschitz network)
- Provides **custom Lipschitz layers and losses**
- Post-training export to vanilla TensorFlow networks (**no overhead at inference**)

```python
import tensorflow as tf
from deel.lip.model import Sequential
from deel.lip.layers import (SpectralConv2D,
    SpectralDense, ScaledL2NormPooling2D)
from deel.lip.activations import GroupSort2
from deel.lip.losses import MulticlassHKR

model = Sequential(
    [
        tf.keras.Input((28, 28, 3)),
        SpectralConv2D(filters=16, kernel_size=3),
        GroupSort2(),
        ScaledL2NormPooling2D(),
        tf.keras.layers.Flatten(),
        SpectralDense(units=10),
    ]
)

model.compile(
    loss=MulticlassHKR(alpha=50, min_margin=0.1),
    optimizer="adam",
    metrics=["accuracy"],
)
model.fit(x_train, y_train)
```

Renault Welding use case

Training

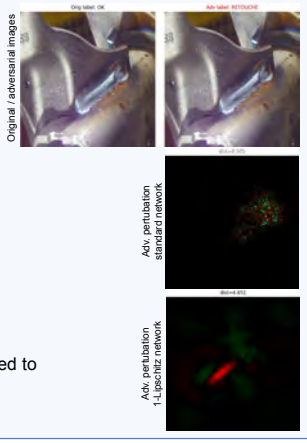
- Classification of welding images in two categories: OK or Retouche.
- Small 1-Lipschitz VGG-like network
- Trainings with different losses and hyper-parameters for accuracy/robustness trade-off:
  - **hinge loss** with different margins
  - **HKR loss** [4] with different margins and regularization factors

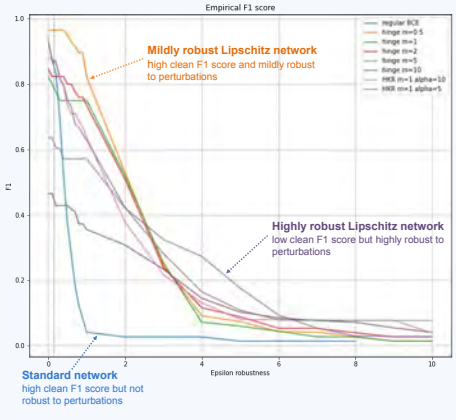
Metrics

- **Clean F1 score**: F1 score of a network on original test images
- **Robust F1 score**: F1 score of a network using adversarial test images instead of original test images

Results

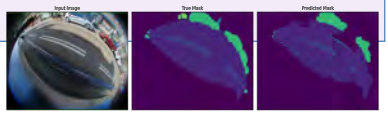
- 1-Lipschitz networks are more robust than standard networks
- Parameters of losses enable a simple accuracy/robustness trade-off
- The adversarial perturbation with 1-Lipschitz network is highly structured compared to more random perturbation with standard network





Perspectives

- New losses to handle accuracy/robustness for multiclass problems
- Upgrades of deel-lip library with recent works + deel-torchlip library transfer
- Image segmentation on Valeo Woodscape use case
- Robust object detection by design



[1] Figure from **Explaining and Harnessing Adversarial Examples**, I. Goodfellow, et. al., https://arxiv.org/abs/1412.6572
[2][3] Figures from **Pay attention to your loss: understanding misconceptions about 1-Lipschitz neural networks,** L.Béthune, T. Boissin et. al., https://arxiv.org/abs/2104.05097v5
[4] **Achieving robustness in classification using optimal transport with hinge regularization**, M. Serrurier et. al., https://arxiv.org/abs/2006.06520

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Confiance ai

FRANCE 2030

Robustification of NN by Diffusion Purification

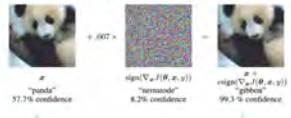
Martin GONZALEZ[1], Nelson FERNANDEZ-PINTO[2]

[1]IRT SystemX, [2]Air Liquide

Adversarial attacks (huge ML community)

Are *formalized* to
- Be imperceptible
  - To *human perception*
  - To *detection* mechanisms
- Exploit the NN's architecture
  - By accessing the model's infrastructure
  - By mimicking the model's behavior
- Maximize the ML damage
  - Attack the model's accuracy
  - Attack a specific target

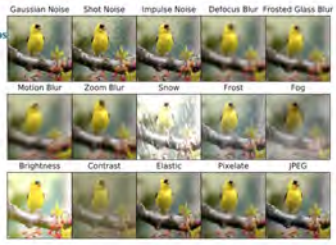
Serve as *worst-case* **In-Distibution analysis for network robustness**



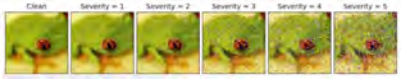
Corruption attacks (small & growing ML community)

Are NOT *formalized*
- Occur in *real-world* scenarios
  - Meteorological changes
  - Digital equipment changes
  - Bad quality data
- Are *human-centered*
  - Human detection
  - Can only be simulated
- Incur into realistic damage
  - Non-foreseeable scenarios

Serve as *average-case* **Out-of-Distribution analysis for network robustness**

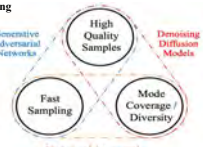


Example of Impulse Noise:
- *caused* by the discrete nature of electric charge
- *occurs* in photon counting in optical devices
- *simulated* by Poisson processes

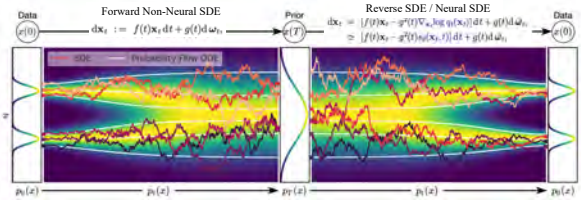


Defenses

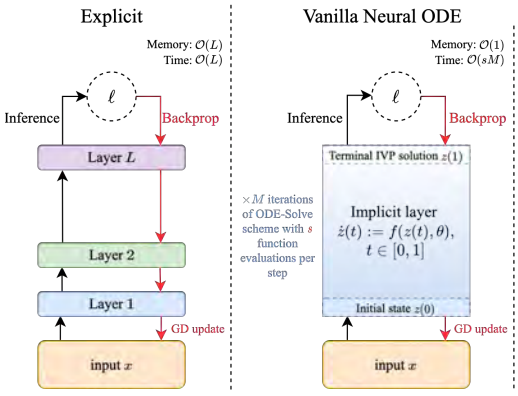
	Adversarial Training: Train on adversaries	Input Purification: Purify perturbations with **denoising diffusion generative models**
Against seen threats	☺	☹
Against unseen threats	☹	☺
Training complexity	☹	☹

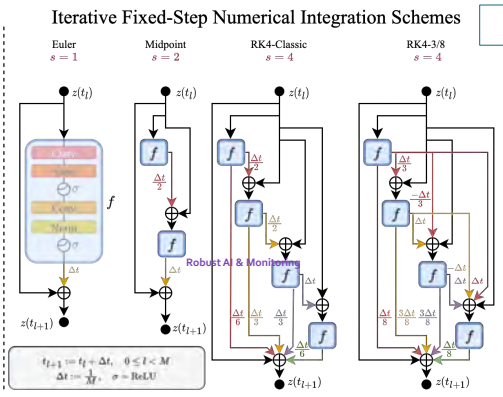


Denoising Diffusion Probabilistic Models



Neural Differential Equations (NDEs) [1] as Implicit Layers





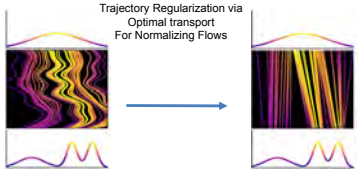
Analytic Formulation

$$\begin{cases} \dot{z}(t) = f(t, z(t), \theta(t), x) \\ z(0) = h_x(x) \\ \hat{y}(T_x) = h_y(z(T_x)) \end{cases} \qquad t \in \mathcal{T}_x = [0, T_x]$$

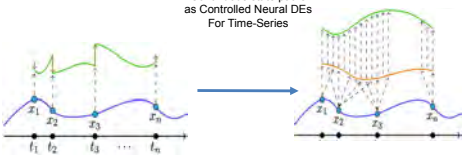
- h_x, h_y: feature extractor, classifier NNs
- t: instant layer of the *depth continuum* $\mathcal{T}_x$
- dependence on t for f (resp. θ): depth–dependence (resp. depth–variance)
- dep. on h_x for $z(0)$: input layer augment.
- dep. on x for f (resp. T_x): data–control (resp. depth–adaptation)
- T_x may be tunned by a hyper–network.

Neural DEs meet Dynamical Systems : Taxonomy & Methods [2]

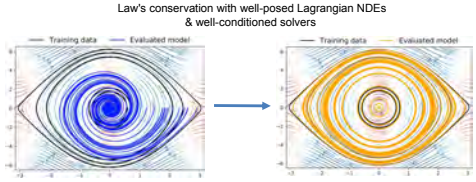
DS-Inspired Neural DEs:
DS idea implemented to a *pre-specified* NDE



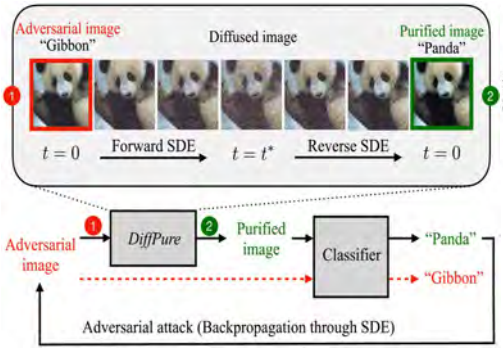
DS-Based Neural DEs:
DS idea implemented as *specifying* a NDE

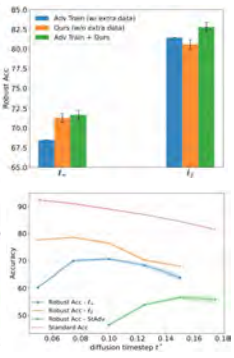


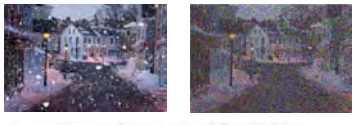
DS-Destined Neural DEs:
NDE induced from a *pre-specified* DS model



The Denoising Diffusion Purification Method [3,4]







	Snow100K-S [ ]		Snow100K-L [ ]	
	PSNR ↑	SSIM ↑	PSNR ↑	SSIM ↑
SPANet [ ]	29.92	0.8260	23.70	0.7930
JSTASR [ ]	31.40	0.9012	25.32	0.8076
RESCAN [ ]	31.51	0.9032	26.08	0.8108
DesnowNet [ ]	32.33	0.9500	27.17	0.8983
DDMSNet [ ]	34.34	0.9445	28.85	0.8772
SnowDiff$_{64}$	**36.59**	**0.9626**	**30.43**	**0.9145**
SnowDiff$_{128}$	36.09	0.9545	30.28	0.9000
All-in-One [ ]	–	–	28.33	0.8820
TransWeather [ ]	32.51	0.9341	29.31	0.8879
WeatherDiff$_{64}$	**35.12**	**0.9539**	**29.55**	**0.8988**
WeatherDiff$_{128}$	34.72	0.9509	29.21	0.8911

(a) Image Desnowing

Promising Directions
- Powerful SDE Solvers = Faster & Accurate Image Restoration
- Guided Diffusions through multi-output classification
- Combination with probably certifiable defenses
- Monitoring strategies based on the solver parameters, diffusion time-stamp, well-posedness of the model's inductive bias …
- **Ongoing applications on Confiance.ai Use Cases**

Bibliography:

1. Chen et al, **Neural Ordinary Differential Equations,** NeurIPS, 2018,
2. Gonzalez et al, **Noisy learning for Neural ODEs acts as a robustness locus widening**, ICML, 2022,
3. Nie et al, **Adversarial purification with diffusion models,** ICML 2022,
4. Ozdenizci, Legenstein, **Restoring vision in adverse weather conditions with patch-based denoising diffusion models**, preprint, 2022.

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Robust AI & Monitoring

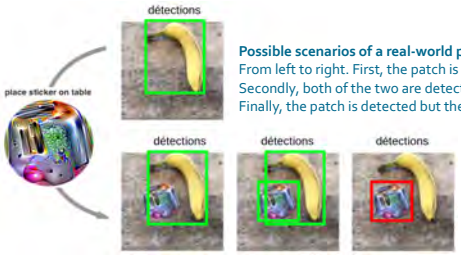
Benchmarking and deeper analysis of adversarial patch attack on object detectors

Pol LABARBARIE [1,2], Stéphane HERBIN [2], Adrien CHAN-HON-TONG [2], Milad LEYLI-ABADI [1]

[1] IRT SystemX, Palaiseau, France; [2] ONERA/DTIS, University Paris-Saclay, Palaiseau, France

1. Context

A realistic attack for physical applications named adversarial patch-based attack (APA) has been introduced recently. It relies on adding a heavily textured patch to the scene causing false alarms and non-detections as depicted by the figure below.



Possible scenarios of a real-world patch attack on object detector. From left to right. First, the patch is not detected but the object is. Secondly, both of the two are detected conducting a false positive. Finally, the patch is detected but the object is not.

From a trustworthy AI point of view, the last scenario (red box), *i.e* deletion of objects of interest detections is not acceptable. Placing a patch on a stop sign or on the roadway may result in misclassification of a stop sign (Song *et al.*, 2018) or in the missed detection of a pedestrian crossing the road (Saha *et al.*, 2020).

2. Objectives

- Identification of the research community requirements which are
  - Understanding the impact of putting a patch in the scene
  - Design defense strategies
- Through measuring the patch resilience wrt. different variations as
  - Geometric transformations
  - Radiometric transformations
- Main contributions:
  - Define various categories of evaluation criteria
  - Proposing a pipeline to rank adversarial patch attacks

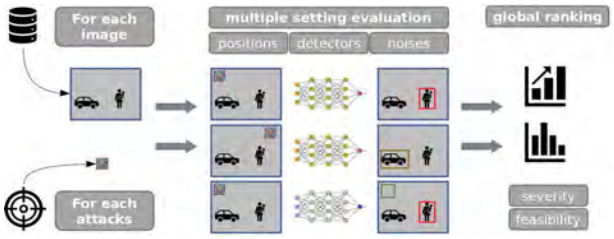
3. Proposition

- Focusing on realistic physical attacks instead of digital attacks (the attacker has access to components).

Category	Setting	Description
Radiometric	Varying weather conditions Filters	Brightness, snow, rain, … JPEG transformations
Geometric	Rescaling Crop Affine transformations Distance w.r.t learning position	*** *** Rotations Shift from learning position
Transferability	Detector sensitivity Detector generalization	Sensitivity of a detector parameters to APAs Generalisation of an APA through multiple detectors

Table of evaluation settings by category and their brief description.

- The resulting average precision (AP) is used to rank attacks for each setting. The overall rating measures the real impact in physical conditions of each APA.



Structure of the proposed pipeline to evaluate APAs.

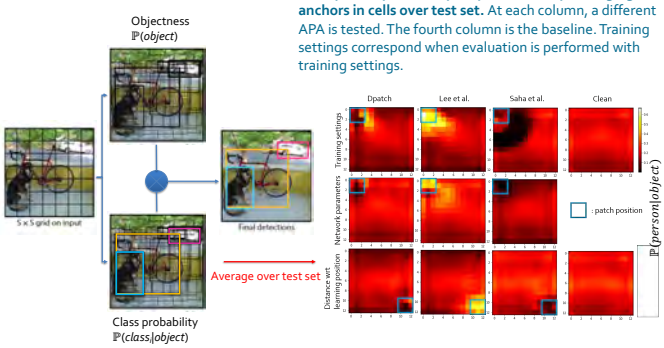
4. Application and results

Experimental setup :

- Using the PASCAL VOC test dataset
- Evaluating patch contextual effects *i.e* the patch does not overlap with the object of interest (creation of subdataset without overlapping) and detections on the patch are removed
- Attacking the *person* class
- Patch learned at top-left location and applied at the learned position by default

Evaluating three state-of-the-art attacks; Dpatch (Liu *et al.*, 2018), Lee *et al.* (Lee *et al.*, 2019) and Saha *et al.* (Saha *et al.*, 2020) on YOLOv2.

Class person probability map obtained by averaging anchors in cells over test set. At each column, a different APA is tested. The fourth column is the baseline. Training settings correspond when evaluation is performed with training settings.



YOLOv2 detection pipeline. Divides the image into a *S* x *S* grid. For each cell predicts *B* modifications of anchors, objectness score for those boxes and class probabilities.

Setting	Attack	Attacked AP (%) w/ f.p	Attacked AP (%) w/o f.p	Cleaned AP (%)
Same as training	Dpatch	71.42	75.01	76.13
	Lee *et al.*	10.56	74.36	
	Saha *et al.*	**59.36**	**59.47**	
Other initialization	Dpatch	73.34	75.25	
	Lee *et al.*	60.35	75.42	
	Saha *et al.*	75.55	75.55	
Shift from learning position	Dpatch	70.61	77.87	80.01
	Lee *et al.*	53.02	78.73	
	Saha *et al.*	74.28	75.87	

Table of the evolution of the AP score for different setting evaluation and for different APA.

Conclusions:

- Our framework allows to evaluate real impact of APAs
- Dpatch and Lee *et al.* have low contextual effects limiting their criticality
- Current attacks are sensitive to setting change lowering the practical risk of current APA's

5. Future work

- Plan to improve this framework in future works to add more transferability experiments (in particular with transformer model)
- Improve current attacks to make them resilient to setting change.

6. References

[Song *et al.*, 2018] Dawn Song *et al.*, Physical adversarial examples for object detectors. In 12th USENIX workshop on offensive Technologies (WOOT 18), 2018.
[Saha *et al.*, 2020] Aniruddha Saha *et al.*, Role of spatial context in adversarial robustness for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 784–785, 2020.
[Lee and Kolter, 2019] Mark Lee and Zico Kolter. On physical adversarial patches for object detection. Preprint arXiv:1906.11897, 2019.
[Liu *et al.*, 2018] Xin Liu *et al.*, Dpatch: An adversarial patch attack on object detectors. SafeAI 2019 (AAAI Workshop on Artificial Intelligence Safety), 2018.

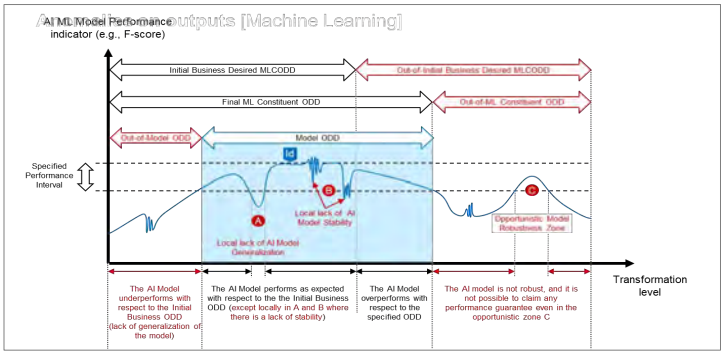
Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

MultiTimescale Monitoring of AI Models

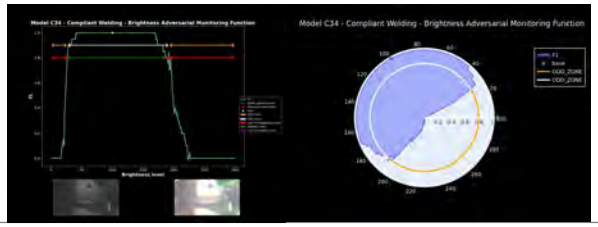
Fateh KAAKAI (1), Paul-Marie RAFFI (2), Guillaume BERNARD (3)

(1) Thales, (2) IRT SystemX, (3) LNE

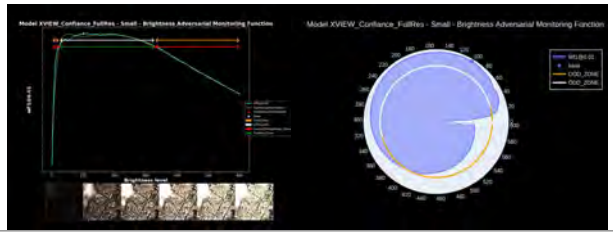
1 When the system does not behave or does not continue to behave as specified or in the specified context … we observe the advent of **anomalies** such as weak generalization capability, lack of stability, lack of robustness, concept/distribution drift, intrinsic limitation of the AI technology, lack of explainability, unsafe unintended behavior, etc. These anomalies increase the **safety risk** related to the failure conditions of the product and this situation could be unacceptable.



Welding Use Case (Renault)



Aerial Photograph Interpretation (Thales)

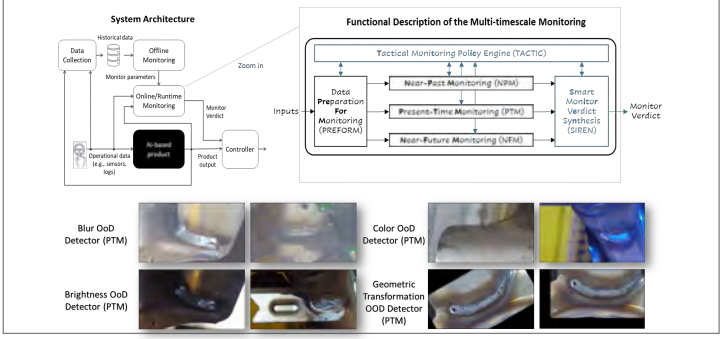


Some useful definitions from Confiance.AI taxonomy (v2):

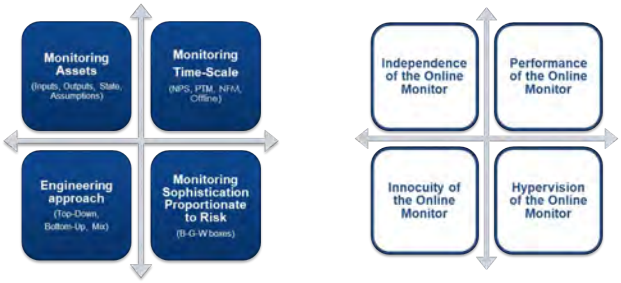
- **Robustness**: The capacity of a model to preserve its expected / intended performance under well-characterized abnormalities or deviations to its inputs and operating conditions outside its operational design domain (ODD)
- **Stability:** The capacity of an ML model to preserve its expected / intended performance under well-characterized and bounded perturbations to its inputs and operating conditions within its operational design domain (ODD)

2 The deployment of a monitoring device operating in parallel with the IA model (online monitoring) is a concrete way of managing this type of residual risk induced by a model for which it is not possible or feasible to formally demonstrate the achievement of the performance/safety objectives resulting from the system analyses. Online monitoring is a safety architectural pattern that is well known to safety engineers, but it had to be adapted to AI technologies. We propose to combine several monitoring time scales: a monitoring of the product at the present time, a monitoring on a configurable time window of the near past, a monitoring on a configurable time window of the near future.

Functional Architecture of the Multi-Timescale Monitoring

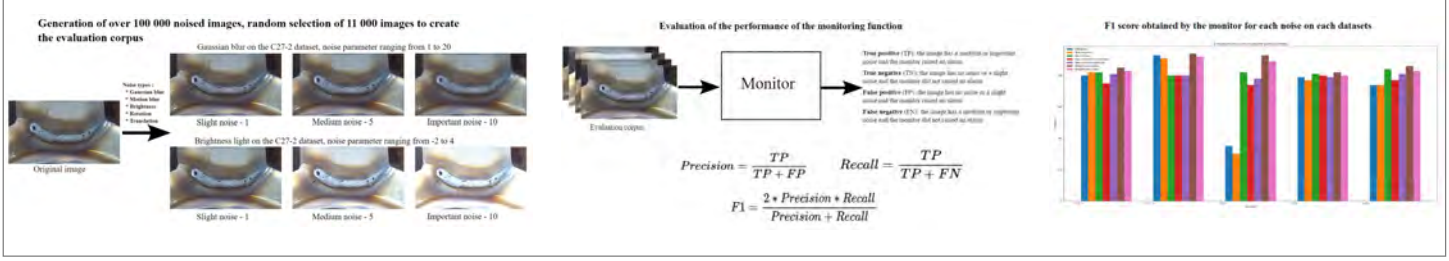


Key Engineering Principles to Design Online Monitor for AI Models



3 The monitoring libraries developed within the frame of Confiance.AI program are independently verified by LNE engineers

Monitor Verification Methodology and Results



Confidence indicators based on time series uncertainty decomposition for system monitoring

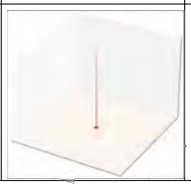
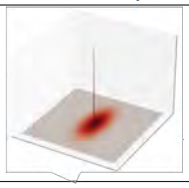
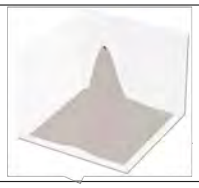
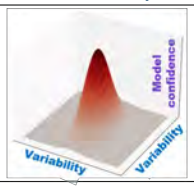
Kevin PASINI

IRT- SystemX

1. Uncertainty decomposition formalism for times series monitoring

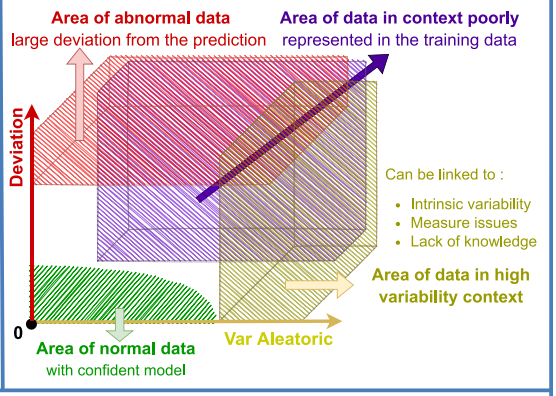
Estimation of 3 time-dependant indicators :

- $|y_t - \hat{y}_t|$: Deviation - model error
- σ_t^A : Aleatoric variance - data variability
- σ_t^E : Epistemic variance - model uncertainty

Scope	Standard ML Determinist	Data variability Aleatoric	Model uncertainty Epistemic	Uncertainty decomposition Aleatoric & Epistemic
Hypothesis	μ-estimator reliable Insignificant variance	μ-estimator reliable Significant variance	μ-estimator uncertain Insignificant variance	μ-estimator uncertain Significant variance
Formalism	$Y_t = y_t \approx F_{\theta^*}(x_t)$ $\hat{Y}_t \sim \mathcal{L}(\mu_t, 0)$	$Y_t = \overline{y_t} \pm \varepsilon_t \approx \mathcal{P}(Y\|x_t, \theta^*)$ $\hat{Y}_t \sim \mathcal{L}(\mu_t, \sigma_t^A)$	$Y_t = \overline{y_t} \pm \varepsilon_t \approx \mathbb{E}_\theta[F_\theta(x_t)]$ $\hat{Y}_t \sim \mathcal{L}(\mathcal{L}'(\mu_t, \sigma_t^E), 0)$	$Y_t = \overline{y_t} \pm \varepsilon_t \approx \mathbb{E}_\theta[\mathcal{P}(Y\|x_t, \theta)]$ $\hat{Y}_t \sim \mathcal{L}(\mathcal{L}'(\mu_t, \sigma_t^E), \sigma_t^A)$
Mean Variance	Determinist : μ_t Nulle : 0	Determinist : μ_t Determinist : σ_t^A	Probabilist : $\mathcal{L}'(\mu_t, \sigma_t^E)$ Nulle : 0	Probabilist : $\mathcal{L}'(\mu_t, \sigma_t^E)$ Determinist : σ_t^A
Illustration				

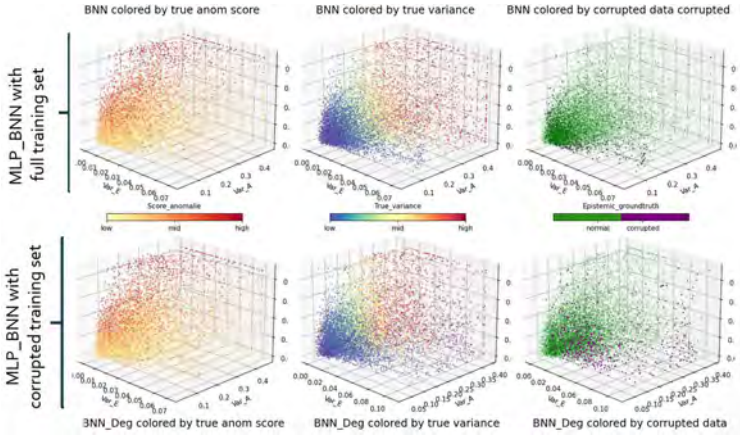
2. Theorical abstraction :

3 orthogonal indicators
to design **confidence indicators :**
Scores of Anomaly, Confidence, variability



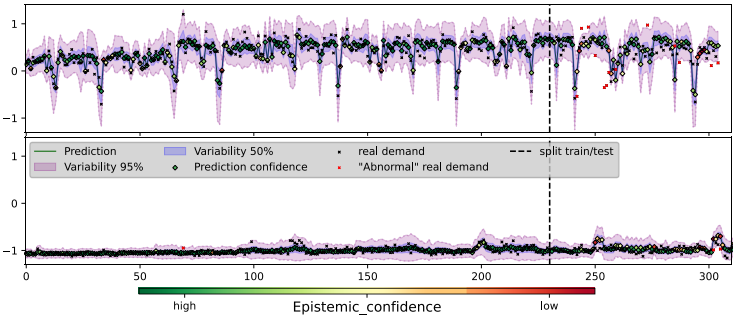
3. Application on synthetic series

3D indicator space representation of data for a control & degraded model



4. Application on real data (Gas demand) :

MLP-BNN indicators for train & test set on 2 different "data context"



Benchmark 3 approach of UQ decomposition :

- **RF-UQ** : Random forest with UQ - Ensemble of model
- **MLP-BNN** : Bayesian NN - Specific loss + non-deterministic NN
- **MLP-EDL** : Evidential Deep Learning - Specific loss + deterministic NN

5. Proto-pipeline of UQ components

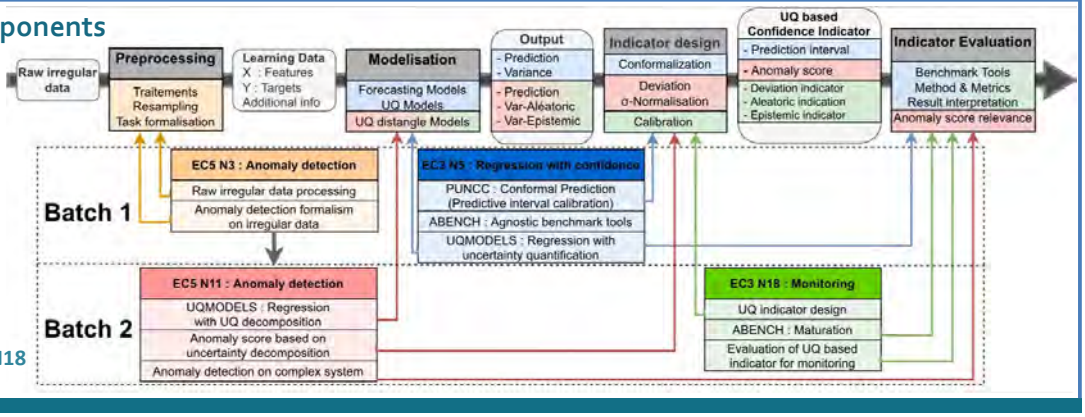
Interaction within Confiance.AI :

Works of the 1st year :

- Anomaly detection formalism - **EC5N3**
- Regression with confidence - **EC3N5**

Works of the 2nd year :

- Anomaly on complex system - **EC5N11**
- System Monitoring on Time series - **EC3N18**



Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Robustness of Neural Networks Based on MIP Optimization

Ramzi BEN MHENNI, Mohamed IBN KHEDHER, Stéphane CANU

IRT SystemX, INSA Rouen

1-Introduction

Inputs

- *Neural network*
- Data *(Images)*

MIP Formulation → MIP solver *Gurobi / Cplex* → Improved version of the MIP solver → Evaluation on images

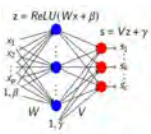
EC3.9

Outputs

1. Check if an adversarial example exists!
2. Find the optimal adversarial example

➢ **Develop a dedicated Mixed-Integer Programming (MIP) solver**

2-MIP Formulation [1,2,3]



The Neural Network function f with c output nodes

$z = ReLU(Wx+\beta)$, $f(x) = Vz+\gamma$; $h = Wx+\beta$, $z = \max(h,0)$, $f(x) = s = Vz+\gamma$

The associated classification (or decision function)

$C_f(x) = \underset{k=1,\dots,c}{\operatorname{argmax}}\, s_k$

Formalizing the search for adversarial exemples :

$\min_{x \in [0,1]^p} \|x - a\|_p^p$ subject to $h = Wx+\beta$, $z = \max(h,0)$, $s = Vz+\gamma$, $\max s_i > s_{i^*}$

Minimizing the Adversarial Distortion

Reformulation

MIP

$\min \|x - a\|_p^p$ subject to $h = Wx+\beta$, $z \geq 0$, $z \leq M_+ h$, $z \leq h + M_-(1-h)$, $z \geq h$, $s = Vz+\gamma$, $\max s_i > s_{i^*}$

3-MIP Optimization

Branch-and-Bound principle

alternating between two steps :

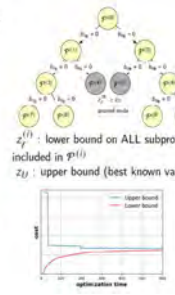
branching : decision $b_{ij} \neq 0$ or $b_{ij} = 0$? → binary search tree

bounding : can the node i contain an optimal solution ? → compute a lower bound $z_i^{(i)}$

If $z_i^{(i)} \geq z_U$, the node i is pruned

Efficient Branch-and-Bound :

- **good lower bound + fast algorithm (relaxation)**
- good upper bound (heuristics)



$z_i^{(i)}$: lower bound on ALL subproblems included in $\mathcal{P}^{(i)}$

z_U : upper bound (best known value)

dedicated Branch-and-Bound

Lower Bound and convex relaxation

Idea : build a convex problem without binary variables

Continuous relaxation

$\min_{x \in [0,1]^p} d(a-x)$ s.t. $h = Wa+\alpha$, $z \geq h$, $z \geq 0$, $z \leq M_+ h$, $z \leq h - M_-(1-h)$, $s = Vz+\alpha$, $s \leq s_y$

Convex relaxation

$\mathcal{P}^R$: $\min_{x \in [0,1]^p} d(a-x)$ s.t. $h = Wa+\alpha$, $z \geq h$, $z \geq 0$, $z \leq u_i h$, $s = Vz+\alpha$, $s_i \leq s_y$

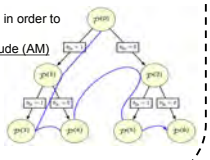
equivalent problem

Branching rules and exploration strategy

Idea : exploit the sparsity of the solution vector associated to the ReLUs activation functions

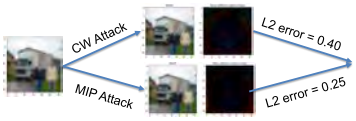
❖ *most ReLU functions will not be activated*

- Which node will be explored first?
  - ➢ *Depth-Up First Search (DUFS)*
- Which variable used in order to subdivide problem?
  - ➢ Maximum amplitude (AM)



4-Results

Solution quality (Clfar10)

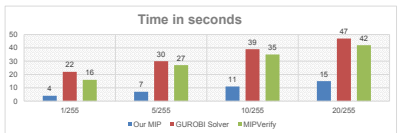


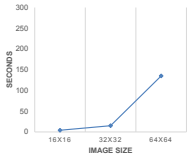
Attacks	PGD [Madry et al., ICLR 2018]	Evasion [Biggio, et al., ECML 2013]	FSGM [I. Goodfellow et al, ICLR 2015]	CW [Carlini & Wagner (CW), 2017]
Improvement	56.7 %	51.4 %	64.8 %	21.2 %

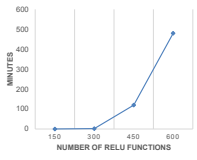
Quality improvement of the solution (norme L2)

Improvement $= \frac{\|Image - Attque_{MIP}\|_2}{\|Image - Attque\|_2} \times 100$

Computing time







5-Conclusions

Evaluation of the solution quality

- ➢ improvement of the solution quality (attack)

Evaluation of the computing time

- ➢ our method is more efficient than the generic MIP solver Gurobi and the state-of-the-art methods for MIPs
- ➢ complexity is linear as function of the image size
- ➢ complexity is exponential according to the number of ReLUs.
  - ➢ not adapted to the model with a high number of ReLUs

1. Fischetti, M. and Jo, J. Deep neural networks and mixed integer linear optimization. Constraints (2018)
2. Bunel, R. et al. "Branch and Bound for Piecewise Linear Neural Network Verification." J. Mach. Learn. Res. (2020)
3. V. Tjeng, K. Xiao and R. Tedrake: Evaluating Robustness of Neural Networks with Mixed Integer Programming (2017)

Robust AI & Monitoring

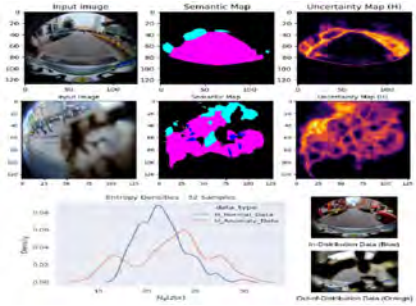
OOD Detection using DNN Latent Space Uncertainty

Fabio Arnez, Ansgar Radermacher

Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

Semantic Segmentation Uncertainty Estimation

Semantic segmentation uncertainty estimation comparison for in-distribution and out-of-distribution data

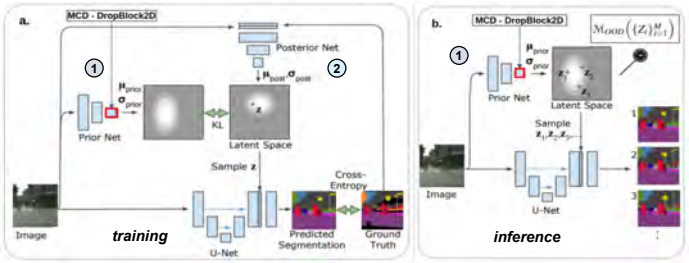


Contribution

Use the DNN uncertainty from intermediate latent features for Out-of-Distribution Detection

Probabilistic U-Net for Semantic Segmentation

Probabilistic U-Net with Bayesian Prior Net for Semantic Segmentation



The Probabilistic U-Net finds useful embedding of segmentation variants in the **latent space** – a central component of this architecture – by introducing a Posterior Net ②

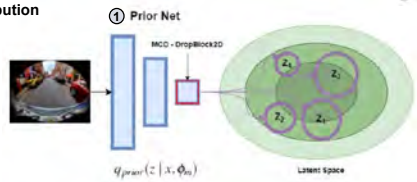
The difference between the distributions at the output of the Prior Net ① and the Posterior Net ② is penalized using the KL divergence.

During inference, the Prior Net ① encodes each input image x and estimates the probability of segmentation variants.

Capturing Uncertainty from Latent Features

$$① \underbrace{q_\Phi(\mathbf{z} \mid \mathbf{x}, \mathcal{D}_p)}_{\text{Prior Net Posterior Predictive Distribution}} = \int q(\mathbf{z} \mid \mathbf{x}, \phi) \underbrace{p(\phi \mid \mathcal{D}_p)}_{\text{Monte-Carlo DropBlock2D}} d\phi \qquad \phi_m \sim p(\phi \mid \mathcal{D}_p)$$

$$\Phi = \{\phi_m\}_m^M$$



We apply Monte-Carlo DropBlock2D to capture ***epistemic*** uncertainty from the Prior Net ① in the Probabilistic U-Net Architecture

Bayesian Generative Classifier for OoD Detection

$$\mathbb{H}(z \mid x) = \int_z p(z \mid x) \log \frac{1}{p(z \mid x)} dz$$

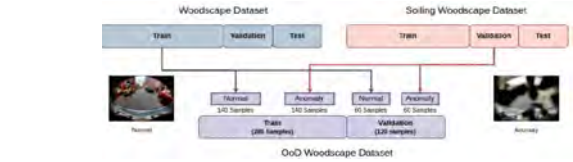
Average surprise (entropy) of observing latent feature z at output of Prior Net, for an input image x.

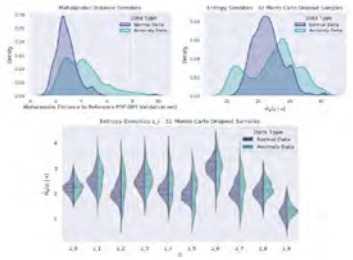
For an unknown latent feature z, we can compute the poster probability of each class (normal, anomaly) using Bayes' Rule:

$$p(y \mid z) = \frac{p(z \mid y)P(y)}{\int_{y \in Y} p(z \mid y)P(y)} \qquad \frac{p(z \mid y)p(y)}{\sum_{y \in Y} p(z \mid y)p(y)}$$

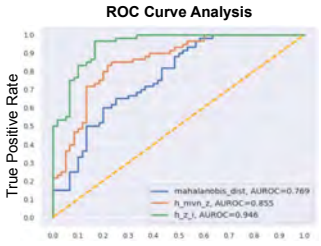
Regression Classification

First results





Densities for Normal and Anomaly Datasets



	Accuracy	Matthews correlation coefficient (MCC)	F1-score	Area under the receiver Operating Characteristic (AUROC)	False-Positive Rate at 90% True Positive rate (FPR90(TPR))
d_M	0.7	0.473	0.763	0.769	0.5
$\hat{H}_\phi(z \mid x)$	0.783	0.572	0.797	0.855	0.4
$\hat{H}_\phi(z_i \mid x)$	**0.825**	**0.685**	**0.849**	**0.946**	**0.16**

Latent Feature Evaluation for OOD Detection
(Metrics, as suggested by Ferreira et al. and Blum et al., both 2021)

Conclusions

- We use the uncertainty from intermediate latent features for Out-of-distribution detection in a semantic segmentation task.
- Early results show that the entropy from latent features can be useful to build classifiers that act as data-driven monitoring functions.

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Design method for improving the detection of out of distribution data of type anomaly by multi-epoch ensemble method

Héléna Vorobieva*°

*Safran Tech, Digital Sciences & Technologies Department °IRT SystemX

Context

- Inspection of industrial parts with **images divided into patches**
  - ➢ Classification or semantic segmentation
- To gain robustness: use several networks via **ensemble methods**
  - ➢ One **neural network at different convergence points** (epochs)

Problem statement

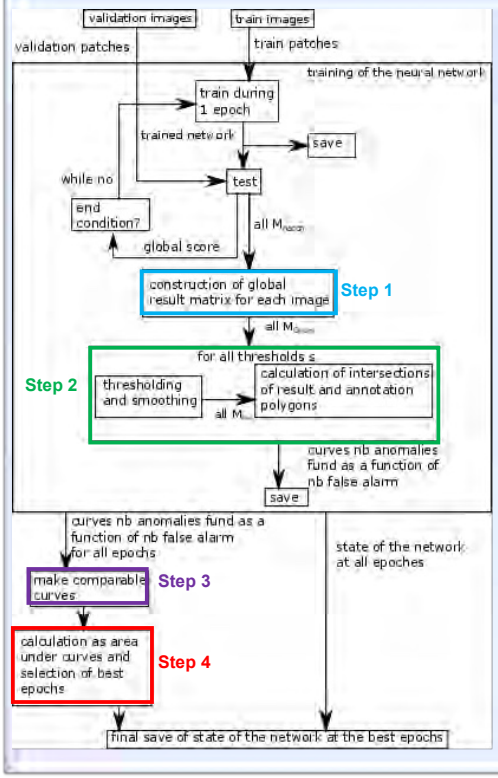
- Industrial problem: **detect anomalies** with few false alarms **in whole image**, even if use of patches
- SotA for best epochs: cost function/scores **inside patches**
- ➢ Problem: **find a score for the whole image** (not local to patches)

Solution

1) Reconstruct global results of whole images by merging results of patches

2) Threshold densely the global results and compare with ground truth

3) Score: **AUC of nb anomalies found function of nb false alarms**



Calibration process during training

Step 1: Construction of a global result matrix for each image of the validation set

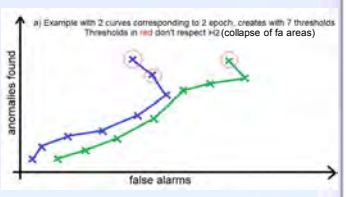
- Merge results of patches M_{patch} using their position → matrix M_{Gnorm} (size of initial img)

Step 2: Thresholding and classification of detected anomalies polygons for each epoch

- Fix dense thresholds S_i in [0,1]
- For each epoch, for each S_i:
  - Threshold each M_{Gnorm} to produce M_{bin} and list all result polygons P_{Ri}
  - For each ground truth polygon P_{GT}, if $\exists\ P_{Ri} : P_{GT} \cap P_{Ri} \neq \emptyset$ then anomaly P_{GT} found
  - For each result polygon P_{Ri}, if $\nexists\ P_{GT} : P_{GT} \cap P_{Ri} \neq \emptyset$ then polygon P_{Ri} false alarm
- ➢ Points $S_{i\_epoch}(nb_{anom}, nb_{fa})$ → $curve_{epoch}(nb_{anom}, nb_{fa})$

Step 3: Make all $curve_{epoch}$ comparables

- Add points $S_\infty(0,0)$
- Given $S_1<S_2$:
  H1: $nb_{anom}(S_1) \geq nb_{anom}(S_2)$, H2: $nb_{fa}(S_1) \geq nb_{fa}(S_2)$
- On each curve, for each S_h not respecting H2 :
  - ➢ new $S_h=(F, nb_{anomN})$
    where $S_N(nb_{faN}, nb_{anomN})$ 1st threshold from which H2 OK
    $F=max_{all\_epoch}(nb_{fa})$ among S_i respecting H2



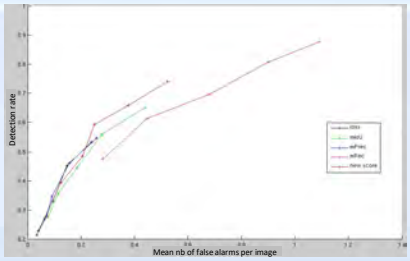
Step 4: Final score

- $N_{admitted}$ = max nb_{fa} accepted in whole val set in a sub-optimal operating regime, ex: $N_{admitted}=nb_{img}$ in val set
- ➢ Final score: AUC in [0, $N_{admitted}$] (higher=better)

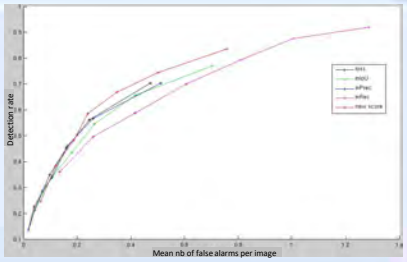
Experiment

Safran UC

- Img 2432x2050x5 divided into 256x256x5 patches
- Classification with resnet-18
- Training during 182 epochs
- $N_{admitted}$=273=nb img in val set
- Test on 1148 img



Test on 5 best epochs (curve with 5 thresholds)



Test on 10 best epochs (curve with 7 thresholds)

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec - Saclay, France

Posters Village: Trustworthy Embedded AI

Introduction to the themes of the village

Jacques Yelloz, Thomas Wouters

How to guide and handle the deployment of trustworthy AI components on target hardware in the frame of industrial applications?

The objective of the "Trustworthy Embedded AI" village is to show you the challenges and issues related to the deployment of trustworthy AI components on target hardware in the framework of industrial applications.

The corresponding activity can be seen as an extension of the design algorithms phase: it is a question of providing the tools necessary to carry out the implementation of an IA component on an embedded hardware target that has limited resources. Generally, the implementation takes as input data or specification a model of the AI component (usually a neural network) that is to be implemented on a specific target. The code corresponding to this implementation is most often specific to the target. During this implementation phase, it is also a question of preserving the properties linked to trust such as, for example, explainability or robustness. This implementation step is most often optional for AI components deployed at the Cloud level. Indeed, frameworks such as pytorch, which allow designing and generating models of AI components, provide all the components necessary for the deployment of these models (see the Model Serving functions at the level of pytorch or MLFLOW). Thus from a practical point of view, the challenge aims to move from a model operating in a Cloud environment to a model operating in an embedded environment with therefore additional constraints related to embedded systems.

This transition from the world of the Cloud to that of the embedded results in developments or adaptations that revolve around the following topics:

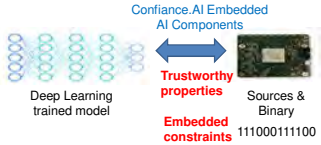
- Methodological: to define the workflows and methodologies taking into account the dimension of embedded systems
- Real time : The challenge is to adapt the compilers in order to give better guarantees related to the predictability of the algorithm execution time. Methods such as WCET on GPU target and Reactive programming are illustrated.
- Benchmarking environment : an environment composed of a set of toolchains, AI models and target in order to support benchmarking, algorithm resource estimation and implementation
- MLMD (Machine Learning Model Description) : A review of a suitable description format to ensure uniformity of the semantic description of the model, regardless of its toolchain provenance, in order to improve the transition from design to implementation in the frame of critical systems.

Trustworthy Embedded AI : scope and challenges

Thomas Wouters [1] and Jacques Yelloz [2]

[1] IRT SystemX - [2] Safran

SCOPE



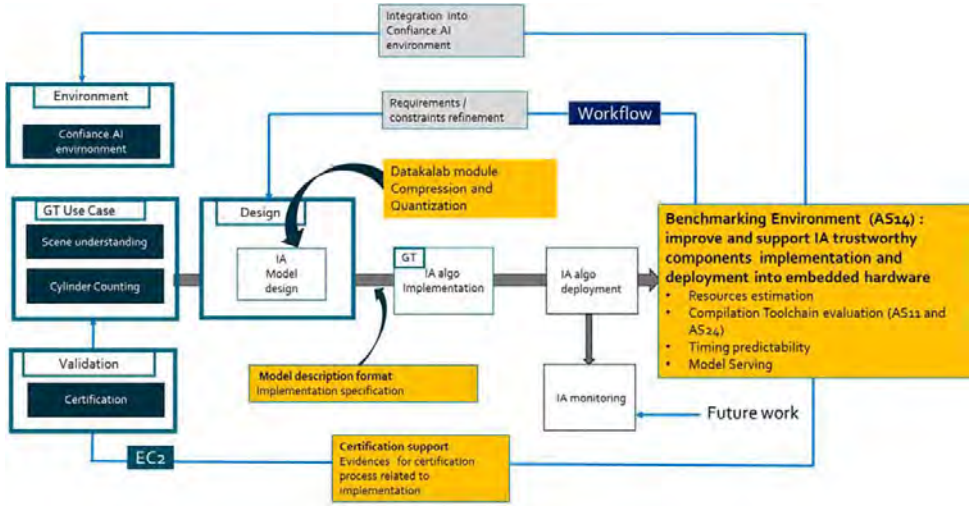
CHALLENGES

- Resources estimation
  - ✓ refine the embedded constraints
  - ✓ support hardware sizing
- Interface/ Interoperability
  - ✓ MLMD : machine learning model description specification to per form implementation indepe ndence from the model design framework
- Compilation toolchain and benchmarking environment :
  - ✓ Ensure conservation of trust properties after the implementation phase : for instance Timing predictabili ty and repetability of ML inference Model
  - ✓ Guidelines and consistency between Algorithm , Framework and Target
- Model optimization
  - ✓ Compression and quantization to fit the constraints of embedded hardware (limited resources) and with trust guarantees (ie : error control of accuracy)
- Certification
  - ✓ methodology for the implementation phase
  - ✓ model survivability in spite of Hardware faults
- Articulation between Confiance.AI and other initiatives on embedded AI

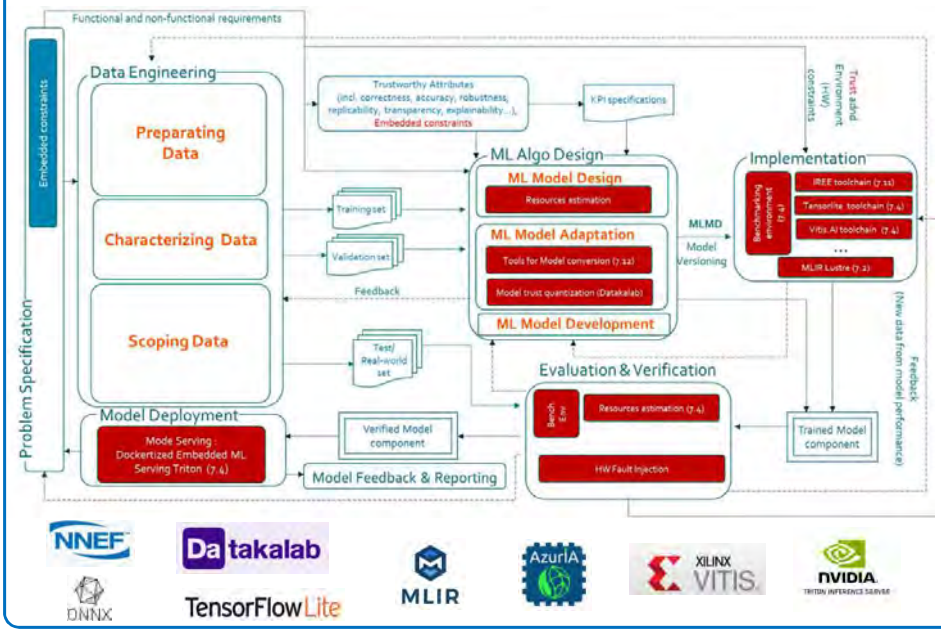
Link with others Confiance.AI projects

Goal is to support others Confiance.AI projects : How to manage the transition from Trustworthy Models running (serving) in the Cloud to models running in embedded constrained systems ?

- ➢Confiance.AI methodology and workflows update to address the challenges of embedded systems
- ➢Trustworthy IA components adaptation to Edge environment (design and implementation phase)
- ➢New IA / ML Pipeline to deal with Models implementation and serving (Trustworthy Environment)



Contributions to Confiance.AI Environment



Benchmarking of AI on Embedded Platforms

Nassim ABDERRAHMANE [1] – Theo ALLOUCHE [2] – Lionel DANIEL [3] – Frédéric FERESIN [3] – Omar HLIMI [1] – Eric JENN [1]– Christophe MARABOTTO [1] – Floris THIANT[4]

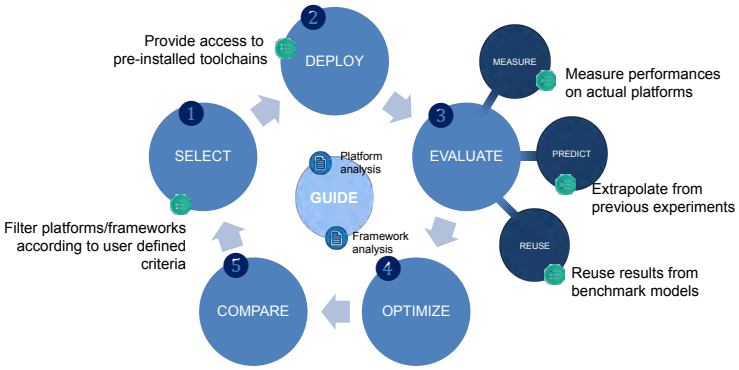
[1] IRT Saint Exupéry - [2] ATOS – [3] AzurIA – [4] IRT SystemX

The operational needs

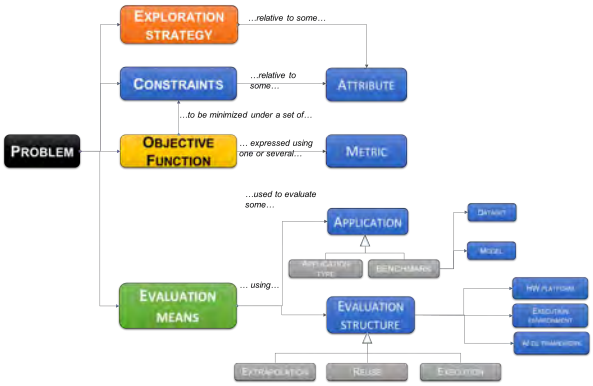
Determine the optimal implementation of a ML model on an embedded platform considering **confidence** and **industrial performance** criteria such as **precision, performance, determinism, maturity,** etc.

This optimization process involves:

- Multiple and complex deployment chains
- Multiple and complex hardware platforms

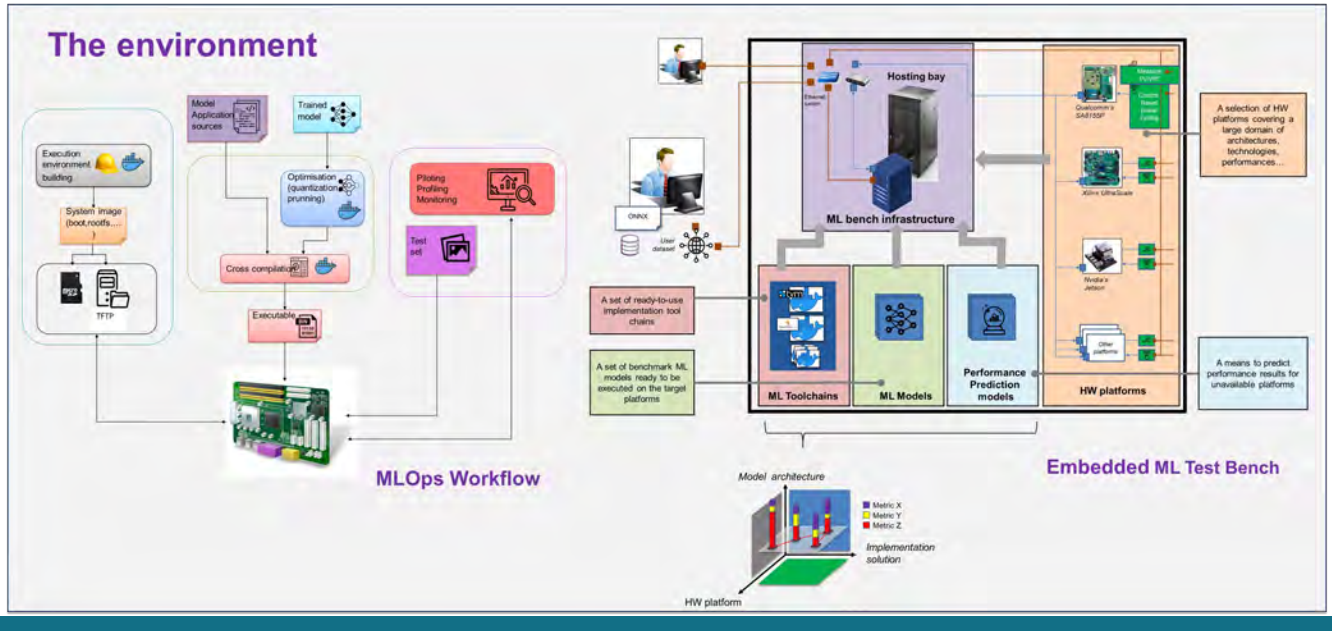


The Optimization Problem



Our objectives

- Select a first batch of HW platforms, models and toolchains
- Define evaluation criteria and cost functions
- Characterize ML implementation toolchains
- Characterize HW platforms
- Elaborate a set of reference benchmarks
- Elaborate deterministic / statistical performance models
- Provide guidance to choose the toolchain / platform couple
- Simplify access to the ML implementation and deployment tools
- Address an ever changing technical landscape (framework, HW)



Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Embedded/Reactive Machine Learning Programming

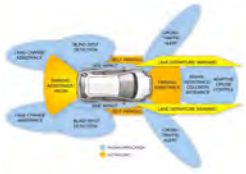
Hugo Pompougnac [1]- Dumitru Potop Butucaru [1] - Albert Cohen [2] - Floris Thiant [3]

[1] INRIA - [2] Google – [3] IRT SystemX

Embedded ML system design: Reactive and Transformational, HPC and RTE

Reactive

- Cyclic execution
- Interaction with environment
- Stateful (for various reasons)
- Real-Time Embedded (**RTE**) main focus



Transformational

- One input, one output
- Functions (stateless computations)
- **HPC** main focus

- **ML specification** (Keras, PyTorch…)
  - **Reactive** intuition **(dataflow)**
  - **Transformational** semantics
    - e.g. LSTM assumes all input data arrives at once
- **ML/HPC compilers, ML runtimes**
  - **Transformational**
  - Performance-focused
- **Embedded implementation, execution platform**
  - **Reactive**
  - Predictability- and safety-focused

The paradox: Reactive/RTE intuition and implementation, Transformational/HPC formalization and compilers
Difficulties to design and to implement: significant manual code interfacing

Contribution: Reactive ML Programming

ACM TACO 2022/HiPEAC 2022

LLVM/MLIR (SSA form) **+** **Lustre/SCADE** (dataflow synchronous) **=** **MLIR-lus**

- Intermediate Representation (IR) and compilation toolbox
  - ML/HPC-native (TF ecosystem)
- Optimization
  - Average-case
  - Incremental « lowering »
- Mainly data parallelism
- Globally sequential, locally concurrent
- Static Single Assignment
- Transformational systems
  - Focus on efficiency
- Core SSA = no absence
  - LLVM adds undef, poison
  - Focus on semantics preservation

- High-level specification formalism
  - RTE-native
- Correction guarantees
  - Functional and non-functional
  - Worst-case analysis
- Mainly task parallelism
- Globally sequential, locally concurrent
- Static Single Assignment
- Reactive systems
  - Embedded real-time applications
- Absence is key part of semantics
  - Checking correction

```
lus.node @lstm(%data: tensor<3x1xf32>) ->
(tensor<3x1xf32>) {
  // Build a clock that is true every 5 cycles
  %lstm_clk = lus.inst @modulocounter %five
  %tmp0 = lus.fby %zero %state0out
  %tmp1 = lus.fby %zero %lstm_out
  %24a = lus.when %lstm_clk %zero
  %24b = lus.when not %lstm_clk %tmp0
  %24 = lus.merge %lstm_clk %24a %24b
  %25a = lus.when %lstm_clk %zero
  %25b = lus.when not %lstm_clk %tmp1
  %25 = lus.merge %lstm_clk %25a %25b
  // LSTM core
  %v26 = tf.MatMul(%v24, %o76)
  %v28 = tf.MatMul(%data, %o22)
  %v29 = tf.AddV2(%v28, %v26)
  %v30 = tf.BiasAdd(%v29, %o78)
  %v31_0, %v31_1, %v31_2, %v31_3 =
      tf.Split(%split_dim, %v30)
  %v32 = tf.Relu(%v31_2)
  %v33 = tf.Sigmoid(%v31_0)
  %v34 = tf.Mul(%v33, %v32)
  %v35 = tf.Sigmoid(%v31_1)
  %v36 = tf.Mul(%v35, %v25)
  %lstm_out = tf.AddV2(%v36, %v34)
  %v40 = tf.Relu(%lstm_out)
  %v41 = tf.Sigmoid(%v31_3)
  %state0out = tf.Mul(%v41, %v40)
  // Output subsampling
  %subsampled = lus.when %lstm_clk %lstm_out
  lus.yield (%subsampled: tensor<3x1xf32>)
}
```

Full-fledged compilation

- Multi-threaded code or IREE-based (GPU/CPU)
- **No performance loss due to reactive execution**

Future: predictable ML runtimes (TF/IREE-focused)

- Memory allocation, scheduling
- Compatibility between frameworks, hardware, and runtimes

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

Trustworthy Embedded AI

Definition of a format for a safe embeddability of a ML Model

Marie-Charlotte TEULIERES[1]

[1] AIRBUS

Why specify a format for the ML Model implementation?

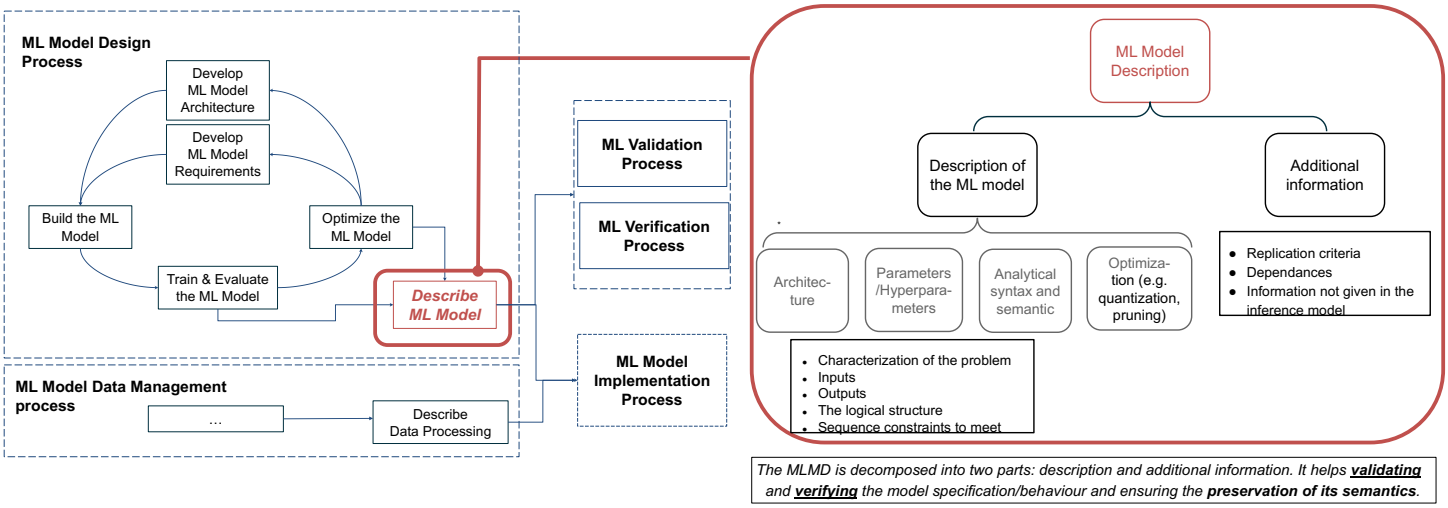
The description of the ML Model is defined as the **interface** between the **design** and the **implementation** processes. For safety-related component, one challenge of the embeddability is to demonstrate that the **implementation** process does not alter the safety/functional/operational properties of the ML model obtained by the **design** process.

To this purpose, the ML model has to be:
- Explicitly and fully described, with no possible interpretation,
- Exactly replicable on a software/hardware target, with no possible approximation,

In order to:
- Ensure the preservation of the semantics
- Be fully verifiable (for conformity to regulation requirements)

The definition of a format is a NEED for embeddability purposes

The concept of ML Model Description (MLMD) for safety-related ML component has been introduced by the standardization joint WGs EUROCAE WG-114 and SAE G-34 in the draft of the future ML standard (**ED-xx/AS6983**) for the development/certification of aeronautical safety-related systems (manned/unmanned aircraft).



*The MLMD is decomposed into two parts: description and additional information. It helps **validating** and **verifying** the model specification/behaviour and ensuring the **preservation of its semantics**.*

Gap Analysis : Industry & Standards requirements VS existing formats?

Standards Objectives and Industries requirements generate criteria for the description format for ML Component. Here are presented an analysis of the gaps between existing formats regarding these criteria.

CRITERIA \ FORMAT	PyTorch	TENSORFLOW	NNEF	ONNX
Interoperability	✗	✗	✓ *Converter from ONNX / TensorFlow / Caffe(2)	✓ *Converter from ~ 25 frameworks
Auditable (/Human Readable)	✗	✗ *Indirectly with a converter	✓	— *Indirectly with a converter
Explicit Description	—	✓	✓	✗
Data Processing Operations	✗	✗	✗	✗
Optimization / Quantization			— *theoretically but converters are not robust to quantization Up to int8 quantization	✓ Up to int8 quantization, can be custom
Non Adherence to Learning Environment			✓	✓
Execution order			✓	✓
Platform agnostic			Parser for C++ / Python Description is platform independent	ONNX Runtime has a list of compatible execution providers for CPU,GPU, IoT/Edge/Mobile etc. implementation
Structure	• one single file	• A Graph file • A folder with variables	• A Graph file • A folder with parameters • A quantization	• one single file • possibility to have parameters stored apart
Additional Informations			proposes a folder structure to be able to collect ML related file, additional file can be added manually	proposes a "doc_string" for documentation of the model

There is no "ideal" existing format.
NNEF & ONNX are the two most complete existing format.
Main issues are on the description of data processing operations, optimization operations and explicit description.

The particular case of N2D2

N2D2 is a platform for ML Design/Training/Deployment, developed by CEA.
It is based on known format such as **ONNX**, but also on a native format INI.

INI files advantages:
CEA is connected to Confiance.AI, INI files are human readable
INI files Disadvantages:
No interoperability of INI Native Format
No explicit description of operations

Confiance.ai Days
4-6 Oct. 2022
CentraleSupélec
Paris-Saclay, France

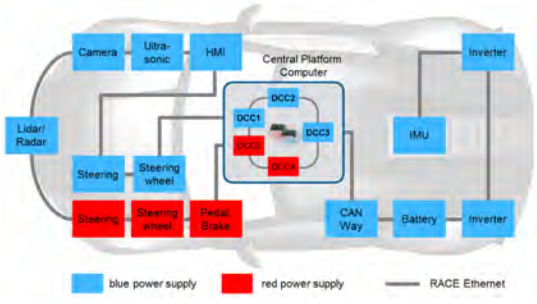
Worst-Case Execution Time Analysis of Neural Networks on GPU accelerators

Michaël ADALBERT[1,3], Christine ROCHANGE[1], Thomas CARLE[1], Serge TEMBO MOUAFO[2], Eric JENN[2], Makhlouf HADJI[3]

IRIT - IRT Saint Exupery - IRT SystemX

1. Context & Objectives

Autonomous Vehicles are critical and have to meet specific requirements to be considered as safe, in particular temporal requirements.

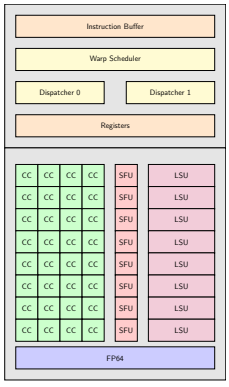


https://www.researchgate.net/figure/E-E-Architecture-with-Central-Platform-Computer_fig3_307804303

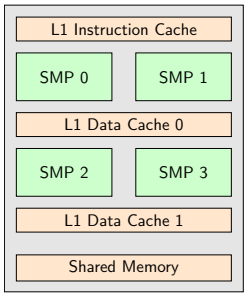
Out objective is to estimate the worst-case execution time of a program running on GPU architectures through static analysis. We have to adapt techniques that already exist for CPUs to GPU and to model the architecture.

2. GPU & SIMT execution model

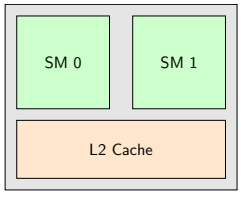
GPUs have hundreds of execution units to run the same program on thousands of threads at the same time.



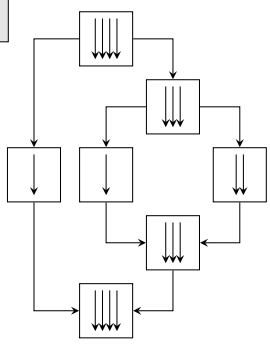
SMP



SM



GPU

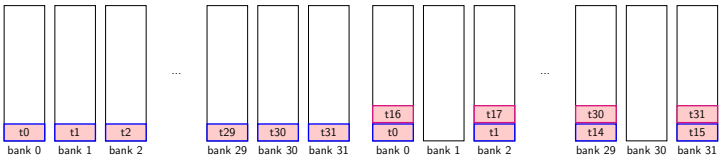


Threads on a GPU are grouped into warps that contains 32 threads. They execute instructions in lockstep. This mechanism leads to a program when the GPU has to execute a divergent program.

3. Research method & example

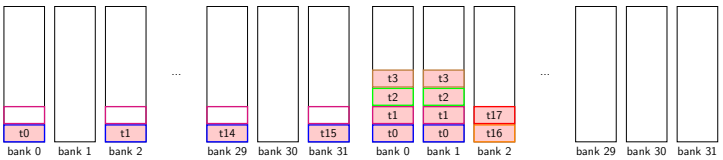
To understand the GPU components, we follow this method:
- Make an hypothesis on a component
- Perform experiments with micro-benchmarks
- Analyse the results to validate or not the hypothesis
- Implement into a simulator

The shared memory is a multi-banked on-chip memory. On accesses threads of the same warp are grouped into transactions. When multiple threads need different words located in the same bank, there is a conflict. So, multiple transactions must be issued.



1 transaction — 2 transactions (1 conflict)

However, some cases are problematic:



2 transactions (64-bit data) — 6 transactions

After multiple experimentations, we conclude that threads are grouped into pools according to the access size:

Access size	32-bit	64-bit	128-bit
Number of pools	1	2	3
Threads per pools	32	16	8
Base (cycles)	1	8	16

$$nTrans = \sum_{i=1}^{nPools} (1 + \max_{j\in[|0,31|]} (conflict(pool_i, bank_j))$$

$$time = 22 + base + 2 \times \sum_{i=1}^{nPools} (\max_{j\in[|0,31|]} (conflict(pool_i, bank_j)))$$

4. Results on matrix multiplication

	GPU		Simulator	
Matrix size	4*4	128*128	4*4	128*128
Reads	8	262144	8	262144
Writes	2	65536	2	65536
Cycles	1875	21797462	1878	22451385

5. Publication

[1] Michaël Adalbert, Thomas Carle, Christine Rochange. PasTiS: building an NVIDIA Pascal GPU simulator for embedded AI applications

Bayesian optimization with deep ensembles for AutoDL

Houssem Ouertatani[1], Cristian Maxim[1], El-Ghazali Talbi[2], Smail Niar[3]

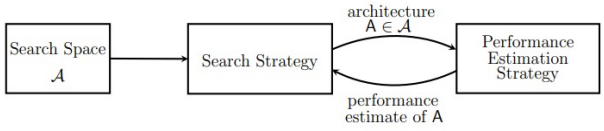
IRT SystemX[1], CRIStAL Lille - INRIA Lille[2], LAMIH[3]

Context and research questions

Automatic design of neural network architectures (AutoDL)

Potential optimization objectives:

- Precision: accuracy (classification), IoU (semantic segmentation)
- Embedded AI: e.g. latency, memory, power consumption
- Trustworthiness: e.g. NetTrustScore, Expected Calibration Error (ECE)



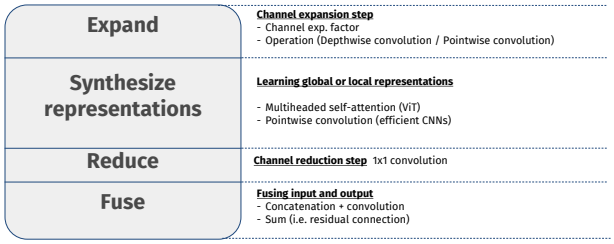
Search space

Unified search space, encompassing efficient CNNs and efficient hybrid CNN-ViTs

CNNs	ViTs
Static weights Local receptive field Inductive biases for images Can train on moderate sized datasets	Data-driven weights Global receptive field Flexibility Data-hungry

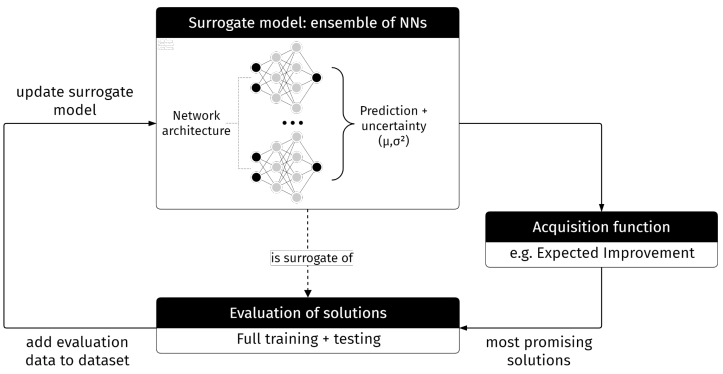
Hybrid CNN-ViTs: Can we achieve the best of both worlds ?

Structure of a block in the proposed search space:



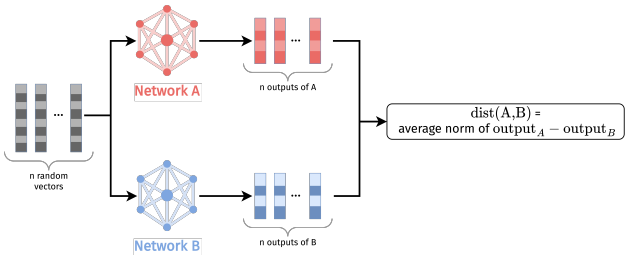
Among the literature vision architectures included in this search space: MobileNet, MobileViT.

Bayesian optimisation with deep ensembles

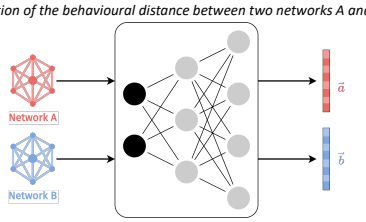


Pre-training procedure

1. Using behavioural distance to enforce close representations for networks which behave similarly



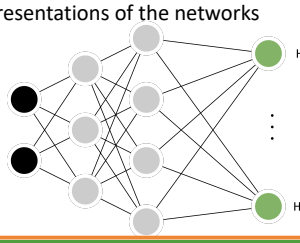
Calculation of the behavioural distance between two networks A and B



$$\mathcal{L}_{\text{bdist}} = \left| \|\vec{a} - \vec{b}\| - \text{dist}(\text{A}, \text{B}) \right|$$

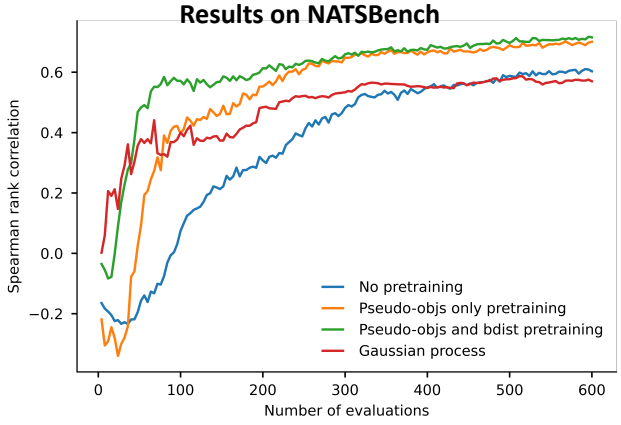
Pre-training one of the ensemble networks using behavioural distance loss

2. Pre-training the ensemble networks on pseudo-objectives to construct good representations of the networks



$$\mathcal{L}_{\text{pretraining}} = \sum_{i}^{m} \mathcal{L}_{\text{pseudo-obj}(i)}$$

Results on NATSBench



[1] El-Ghazali Talbi, "Automated Design of Deep Neural Networks: A Survey and Unified Taxonomy," ACM Computing Surveys 54, no. 2 (March 5, 2021): 34:1–34:37.
[2] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter, "Neural Architecture Search: A Survey"
[3] Hadjer Benmeziane et al., "Hardware-Aware Neural Architecture Search: Survey and Taxonomy," vol. 5, 2021, 4322–4329, accessed March 28, 2022.
[4] Kai Han et al., "A Survey on Vision Transformer," IEEE Transactions on Pattern Analysis and Machine Intelligence (2022): 1–1, accessed March 28, 2022.
[5] Sachin Mehta and Mohammad Rastegari, "MobileViT: Light-Weight, General-Purpose, and Mobile-Friendly Vision Transformer," arXiv:2110.02178 [cs] (March 4, 2022)

Acknowledgements

The program and organization committee would like to thank all the contributors to the Confiance.ai Days 2022 call for papers

This work has benefited from state aid under the “Investing for the Future” (PIA) program within the framework of the Research and Technology Organisation IRT SystemX.