Confiance.ai Days 2021 Poster Booklet

Aurélie Bourrat, Bertrand Braunschweig, Loic Cantat, Lionel Cordesses, Georges Hebrail, Frédéric Jurie, Morgane Toumazet, Patrice Aknin

OCTOBER 4-6 2021 – TOULOUSE, FRANCE

Confiance.ai Days 2021
Poster booklet

program and organization committee

Patrice Aknin, Aurélie Bourrat, Bertrand Braunschweig, Loïc Cantat, Lionel Cordesses, Georges Hebrail, Frédéric Jurie, Morgane Toumazet

Acknowledgements
The program and organization committee would like to thank all the contributors to the Confiance.ai Days 2021 call for posters

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.

List of contributions

- **Vianney Taquet, Grégoire Martinon, Abdou Akim Goumbala, Nicolas Brunel**. MAPIE: Model Agnostic Prediction Interval Estimator. By Quantmetry and LMME – ENSIIE Evry

- **Salim Amoukou, Nicolas Brunel**. Consistent Probabilistic Sufficient Explanations via random Forests. By Stellantis and LMME – ENSIIE Evry

- **Hatem Ibn Khedher**. Mathematical Programming Approaches for AI models Verification and Certification. By IRT SystemX

- **Stefan Suwelack**. The Data Curation Canvas – a Data-centric Approach to Trustful AI. By Renumics GmbH

- **Maxime Savary-Leblanc, Xavier Le-Pallec, Sébastien Gérard**. Fostering Trust in Software Assistants for Software Engineering. By CRISTAL Univ. Lille and CEA List

- **Alexandre Autret, Armin Dietz, Lorena Gayarre Peña, Gaël Richard Alexandre Carlot, Clément Le Couedic, Mehdi Benchoufi, Elsa D. Angelini**. Domain adaptation for COVID-19 detection on lung ultrasound imaging. By Capgemini, CRESS Univ. de Paris, LTCI – Télécom Paris, TechOpen Factory

- **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 ? By CEA List and Valeo.ai

- **Thomas Delorme, Jeremy Trione, Laurence Guillon, Anthony Rossi**. Testing machine learning algorithm in predictive maintenance. By Naval Group

- **Marc Nabhan, Kevin Pasini, Luca Mossina**. Predictive Uncertainty Quantification for Time Series. By Air Liquide R&D, IRT SystemX and IRT Saint Exupéry

- **Guillaume Ollier, Morayo Adedjouma, Simos Gerasimou, Chokri Mraidha**. A cross-domain framework for Operational Design Domain specification. By CEA List and Univ. of York

- **O. Matz, M. El Ouazzani, J. Gantet, N. Diarra, M. Guillaumont, B. Deguilhem**. Speech recognition under constraint. By Capgemini

- **Aurélien Benoit-Lévy**. Randomized smoothing for time-series. Application to the Air Liquide demand forecasting dataset. By CEA List

- **Hugo Pompougnac, Dumitru Potop-Butucaru, Albert Cohen**. A formalism for embedded machine learning applications. By INRIA Kairos and Google France

- **Cyrielle Chappuis**, Trust in AI: increasing acceptance of robotics and artificial intelligence agents through impression design. By Capgemini

MAPIE: Model Agnostic Prediction Interval Estimator

Vianney Taquet[1], **Grégoire Martinon**[1], **Abdou Akim Goumbala**[1], **Nicolas Brunel**[1,2]
1: Quantmetry, 52, rue d'Anjou, 75008, Paris, France
2: Laboratoire de Mathématiques et de Modélisation d'Evry, ENSIIE, University Paris Saclay
vtaquet, gmartinon, agoumbala, nbrunel @ quantmetry.com

Abstract

Estimating uncertainties associated with the predictions of machine learning models is of crucial importance to assess their robustness and predictive power. In this contribution, we present MAPIE [1], an open-source python scikit-learn-contrib package which implements recent resampling and conformal methods, backed by strong mathematical guarantees, to easily estimate uncertainties for regression and classification tasks.

1 Introduction

Decision makers are increasingly relying on machine learning (ML) algorithms. It becomes therefore important to combine the predictive performance of such complex models with practical guarantees on the reliability and uncertainty of their results. However, robust packages that allow data scientists to readily estimate uncertainties associated with the predictions of ML models were still lacking in the data science community.

In this contribution, we present MAPIE, an open-source package, developed at Quantmetry as a Quantlab R&D project, re-introducing the notion of uncertainty in ML. We developed MAPIE with a two-fold objective. First, MAPIE implements state-of-the-art uncertainty quantification methods associated with strong theoretical guarantees on the marginal coverage. Second, MAPIE is model-agnostic and can estimate uncertainties associated with any scikit-learn-compatible estimator and can therefore be integrated in advanced and industrialised ML pipelines. We started the development of MAPIE by implementing the jackknife+ method introduced in [2] and its variations for single-output regression problems. We are now extending MAPIE to other settings with critical applications in industry such as classification, time series, or image segmentation.

2 MAPIE for regression

MAPIE uses various resampling methods based on the jackknife+ strategy recently introduced in [2] allowing the user to estimate robust prediction intervals with any kind of machine learning model for regression purposes on single-output data. The jackknife+ method is based on the construction of a set of leave-one-out models. Each perturbed model is trained on the entire training data with one point removed. Interval predictions are then estimated from the distribution of the leave-one-out residuals estimated by these perturbed models. The novelty of this elegant method is that predictions on a new test sample are no longer centered on the predictions estimated by the base model as with the standard jackknife method but on the predictions from each perturbed model. This small and seemingly minor change allows estimated prediction intervals to be always stable and theoretically guaranteed.

However, the standard jackknife+ method is computationally heavy as it requires to compute as many models as the number of training samples. It is therefore possible to adopt a lighter cross-validation approach, called the CV+. The CV+ method acts as a standard cross-validation: K perturbed models are trained, with K ranging typically from 5 to 10, on the entire training set with each fold removed, and the corresponding residuals are computed. As for the jackknife+, prediction intervals are centered on the predictions performed by each out-of-fold model. The same stability is therefore guaranteed by the theory although the prediction intervals are usually slightly wider since each perturbed model is trained on a lower number of samples. For even lighter computations, it is possible to adopt a split-conformal approach in which residuals are computed on a single calibration set.

Figure 1 compares the prediction intervals estimated by MAPIE on a one-dimensional toy dataset using the CV+ method for three base regressors: (i) a polynomial function of degree 10; (ii) a XGBoost model using the `scikit-learn` API; (iii) a simple 3-layer MLP neural network using a `KerasRegressor` wrapper TensorFlow. It can be seen that the prediction intervals are very similar among the base models with identical coverages of 0.97 and interval widths all close to 2.4.

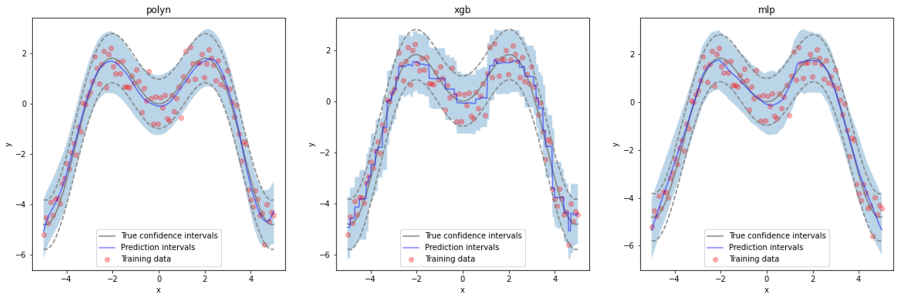


Figure 1: Prediction intervals estimated by MAPIE for a one-dimensional toy regression dataset using a polynomial function (left), a XGBoost regressor (middle), and a multilayer perceptron (right) as base regressor. The toy dataset is generated from a $x \sin(x)$ function with constant noise. The dashed gray lines and blue areas depict the 95% theoretical confidence intervals and prediction intervals estimated by MAPIE.

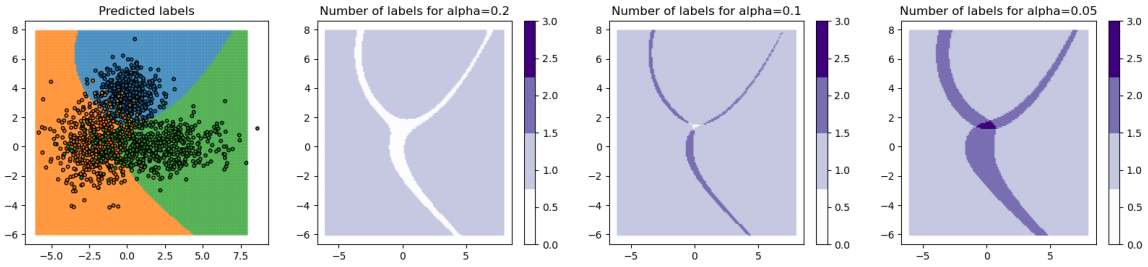


Figure 2: Left panel: Predictions estimated by the Gaussian Naive Bayes classifier on a two-dimensional toy classification dataset. Other panels: Number of labels included in the prediction sets estimated by MAPIE for different significance level α values. The toy dataset is a two-dimensional synthetic dataset with three labels. The distribution of the data is a bivariate normal with diagonal covariance matrices for each label.

3 MAPIE for classification

MAPIE uses conformal methods to estimate prediction sets associated with any scikit-learn-compatible classifier for multi-class problems. Instead of returning the class with the highest probability, MAPIE estimates a prediction set of several classes such that the probability that the true label of a new test point is included in the prediction set is always higher than the target confidence level $1-\alpha$. Conformal methods are particularly useful for usecases requiring to classify a high number of classes, such as e-mail forwarding or image classification. In practice, after training a base model on a training set, MAPIE computes the distribution of conformity scores on a calibration set to estimate a quantile associated with the desired α value. Conformity scores can for instance be the softmax score of the true class output by the model [3] or the cumulated score of all classes until the true class is reached [4]. Finally, MAPIE creates a prediction set for a new test point that includes all classes whose conformity score is higher than the estimated quantile. Split-conformal and cross-conformal variations are both included.

Figure 2 compares the number of labels included in the prediction sets estimated by MAPIE using the "score" strategy for different significance levels α on a toy classification dataset. The apparition of empty prediction sets is a corner case when the confidence level is too small but can be managed with post-processing such as completion algorithm [3]. For smaller α values, MAPIE emphasizes these ambiguous regions with prediction sets containing several labels.

4 References

[1] https://github.com/scikit-learn-contrib/MAPIE/

[2] Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. "Predictive inference with the jackknife+." Ann. Statist., 49(1):486–507, February 2021.

[3] Mauricio Sadinle, Jing Lei, and Larry Wasserman. "Least Ambiguous Set-Valued Classifiers With Bounded Error Levels." Journal of the American Statistical Association, 114:525, 223-234, 2019.

[4] Yaniv Romano, Matteo Sesia and Emmanuel J. Candès. "Classification with Valid and Adaptive Coverage." NeurIPS 202 (spotlight).

Consistent Probabilistic Sufficient Explanations via random Forests

Salim I. Amoukou [1] *and Nicolas J.B Brunel* [2]

Abstract

To explain the decision of any model, we extend the notion of *probabilistic sufficient explanation* (P-SE). For each instance, this approach selects the minimal subset of features that is sufficient to keep the same prediction with high probability, while removing the other features. Our P-SE can deal with regression, non-binary features, without learning the law of X. We also prove the consistency of our method.

1 Introduction

Many methods have been proposed to explain the prediction of a black-box model from different perspectives, including model agnostics approaches (LIME, SHAP, Anchors), or logic-based [1, 2].

In this paper, we generalize the concept of *probabilistic sufficient explanations* (P-SE) introduced in [3] which is a relaxation of logic-based explanation (e.g. [2]). It explains the classification of an instance by choosing the minimal subset of features such that only observing those features is sufficient to give us strong probabilistic guarantees that the model will behave similarly, no matter what is observed for the remaining features. This subset is called sufficient explanation (also known as sufficient reason or prime implicant [1, 2]). Note that it may not be unique.

Our contributions are as follows: we extend the P-SE to the regression case and make it support non-discrete features. Second, our method allows us to explain any data generating process not only a specific model and we no longer need to learn the law of X. Third, to deal with the non uniqueness of the sufficient explanation, we introduce probabilistic local explanatory importance, which indicates how frequent each feature is in the set of all sufficient explanations. Last, we prove the consistency of our method.

2 Probabilistic Sufficient Explanations for Regression

Let assume we have a sample $\mathcal{D}_n = (X_i, Y_i)_{i=1,\dots,n}$ i.i.d distributed as $(X, Y) \sim P_{(X,Y)}$ where $X \in \mathbb{R}^p$, $Y \in \mathbb{R}$. Without loss of generality, we assume that Y is the output of a measurable function f i.e $Y = f(X)$.

In our framework, the explanations of an instance $\boldsymbol{x}$ is the minimal subset $\boldsymbol{x}_S, S \subset [p]$ such that given only those features, with high probability under the data distribution $p(\boldsymbol{X})$, the model makes "almost" the same prediction as on the full example. The main probabilistic reasoning tool that we use for our explanations are the Same-Decision-Probability (SDP) [4]. Below, we propose a definition of the SDP in the regression setting:

Definition 2.1. ***(Same Decision Probability of a regressor).*** *Let $f : \mathcal{X} \longrightarrow \mathcal{Z}$ a regressor, the Same Decision Probability at level t of coalition $S \subset [\![1, p]\!]$, w.r.t $\boldsymbol{x} = (\boldsymbol{x}_S, \boldsymbol{x}_{\bar{S}})$ is*

$$SDP_S\left(f; \boldsymbol{x}, t\right) = P\left(d\left(f(\boldsymbol{x}_S, \boldsymbol{X}_{\bar{S}}), f(\boldsymbol{x})\right) \leq t \,|\boldsymbol{X}_S = \boldsymbol{x}_S\right)$$

In a regression setting, the SDP gives the probability to stay close to the same prediction $f(\boldsymbol{x})$ at level t, when we fixed $\boldsymbol{X}_S = \boldsymbol{x}_S$ or when $\boldsymbol{X}_{\bar{S}}$ are missing. The higher is the probability, the better is the explanation based on S. Therefore, we focus on the minimal subset of features such that the classifier makes the same decision with a given (high) probability π, given only them. Formally:

Definition 2.2. ***(Minimal Sufficient Explanations).*** *Given a model f, an instance $\boldsymbol{x} = (\boldsymbol{x}_S, \boldsymbol{x}_{\bar{S}})$, $S \triangleq S^\star_\pi(\boldsymbol{x})$ is a Sufficient Explanation for probability π if $SDP_{S^\star_\pi(\boldsymbol{x})}\left(f; \boldsymbol{x}, t\right) \geq \pi$ and no subset Z of $S^\star_\pi(\boldsymbol{x})$ satisfies $SDP_Z\left(f; \boldsymbol{x}, t\right) \geq \pi$. Hence, a Minimal Sufficient Explanation is a Sufficient Explanation with minimal size.*

For a given instance, Sufficient Explanation or Minimal Sufficient Explanation may not be unique. We denote C-SE as the set of all Sufficient Explanations, and M-SE as the set of Minimal Sufficient Explanations. Therefore, we propose to compute the following local importance summary for each variable:

Definition 2.3. ***(Local Explanatory Importance).*** *Given a model f, an instance $\boldsymbol{x} = (\boldsymbol{x}_S, \boldsymbol{x}_{\bar{S}})$, $S \triangleq S^\star_\pi(\boldsymbol{x})$ and its C-SE or M-SE. The local explanatory importance of X_i is how frequent X_i is choose in the C-SE or M-SE.*

2.1 SDP and Sufficient Explanations via Random Forest

In order to find the coalitions $S^\star_\pi(\boldsymbol{x})$, we need to compute the SDP for any subset S. However, computing the SDP is known to be computationally hard, even for simple Naive Bayes model, the computation of SDP is known NP-hard.

To consistently estimate the SDP, we propose a variant of the Random Forest algorithm. The algorithm is based on two ideas: Projected Forest [5, 6] and Quantile Regression Forest [7]. The projected Forest is an adaptation of random Forest algorithm that consistently estimates $E[Y|X_S = x_S]$ instead of $E[Y|X = x]$, and the Quantile Regression Forest uses the Random Forest algorithm to go beyond condition mean estimation and to estimate conditional distribution function $P(Y \leq y|X = x)$.

[1] LaMME, University Paris Saclay, Stellantis Paris, salim.ibrahim-amoukou@universite-paris-saclay.fr
[2] LaMME, ENSIIE, University Paris Saclay, Quantmetry Paris, nicolas.brunel@ensiie.fr

The algorithm that estimates the $SDP_S(f;\boldsymbol{x},t)$ can be described as follows: we drop observations down the initial trees of a trained Random Forest, ignoring splits which use a variable outside of S i.e when a split involving a variable outside of S is met, data points are sent both to the left and right children nodes. Consequently, each observation falls in multiple terminal leaves of the tree. We drop the new query point $\boldsymbol{x}_S$ down the tree, following the same procedure, and retrieve the set of terminal leaves where $\boldsymbol{x}_S$ falls. Next, we collect the training observations which belong to every terminal leaf of this collection, in other words, we intersect the collection of leaves where $\boldsymbol{x}_S$ falls. Finally, we average the outputs $\mathbb{1}_{d(Y_i,f(\boldsymbol{x}))\leq t}$ of the selected training points to generate the estimation of $SDP_S(f;\boldsymbol{x},t) = P\left(d\left(f(\boldsymbol{x}_S,\boldsymbol{X}_{\bar{S}}),f(\boldsymbol{x})\right)\leq t\,|\boldsymbol{X}_S=\boldsymbol{x}_S\right)$. The estimator is defined as

$$\widehat{SDP}_S(f;\boldsymbol{x},t) = \sum_{i=1}^{n} w_{n,i}^{(\boldsymbol{x}_S)}(\boldsymbol{x}_S;\Theta_1,\ldots,\Theta_k,\mathcal{D}_n)\mathbb{1}_{d(Y_i,f(\boldsymbol{x}))\leq t}, \tag{2.1}$$

$w_{n,i}^{(\boldsymbol{x}_S)}(\boldsymbol{x}_S;\Theta_1,\ldots,\Theta_k,\mathcal{D}_n) = \frac{1}{k}\sum_{l=1}^{k}\frac{\mathbb{1}_{X_i\in A_n^{(\boldsymbol{x}_S)}(\boldsymbol{x}_S;\Theta_l,\mathcal{D}_n)}}{N_n^{(\boldsymbol{x}_S)}(\boldsymbol{x};\Theta_l,\mathcal{D}_n)}$ is the classic weight of an observation i in a Random Forest [8] but applied to the Projected forest, where k is the number of trees, Θ_l the random parameter vector that determines how the l-th tree is grown (e.g. which variables or observations are considered for split at each node), $A_n^{(\boldsymbol{x}_S)}(\boldsymbol{x}_S;\Theta_l,\mathcal{D}_n)$ is the leaf of the associated Projected l-th tree where $\boldsymbol{x}_S$ falls and $N_n^{(\boldsymbol{x}_S)}(\boldsymbol{x};\Theta_l,\mathcal{D}_n)$ is the number of observation that falls in $A_n^{(\boldsymbol{x}_S)}(\boldsymbol{x}_S;\Theta_l,\mathcal{D}_n)$. Below, we show the uniform a.s. consistency of our estimator.

Theorem 1. *Consider a model f and a random forest which satisfies mild Assumptions (4.1-4.3 in [9]) then,*

$$\forall\boldsymbol{x},\in\mathbb{R}^d,\forall S\subseteq[p],\ \sup_{t\in\mathbb{R}}\left|\widehat{SDP}_S(f;\boldsymbol{x},t) - SDP_S(f;\boldsymbol{x},t)\right| \xrightarrow[n\to+\infty]{a.s} 0 \tag{2.2}$$

With 2.1 we can estimate directly the SDP from a sample $\mathcal{D}_n$. However, finding the C-SE/M-SE using a greedy algorithm is computationally hard, since the number of subsets is exponential. Therefore, we propose to reduce the number of variables by focusing only on the most influential variables. We search the sufficient explanations in the subspace of the 10-variables frequently selected in the forest used to estimate the SDP, reducing the complexity from 2^p to 2^{10}. This selection procedure is already used in other works [5, 10], and it is mainly based on Proposition 1 in [11], which highlighted the fact that forest naturally splits the most on influential variables.

3 Conclusion

An important finding of this work is the generalization of the P-SE in a more realistic context, and the ability to compute efficiently and consistently the SDP for any distribution $P_{(X,Y)}$ under mild assumptions. An open question in SDP is the choice of t and π that might depend on the context and the corresponding admissible error. A natural choice is to relate t to the prediction variance, that can be estimated by resampling techniques.

References

[1] Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining bayesian network classifiers. *arXiv preprint arXiv:1805.03364*, 2018.

[2] Adnan Darwiche and Auguste Hirth. On the reasons behind decisions. *arXiv preprint arXiv:2002.09284*, 2020.

[3] Eric Wang, Pasha Khosravi, and Guy Van den Broeck. Towards probabilistic sufficient explanations. In *Extending Explainable AI Beyond Deep Models and Classifiers Workshop at ICML (XXAI)*, 2020.

[4] S. Chen, Arthur Choi, and Adnan Darwiche. The same-decision probability: A new tool for decision making. 2012.

[5] Clément Bénard, Gérard Biau, Sébastien Da Veiga, and Erwan Scornet. Shaff: Fast and consistent shapley effect estimates via random forests. *arXiv preprint arXiv:2105.11724*, 2021.

[6] Clément Bénard, Sébastien Da Veiga, and Erwan Scornet. Mda for random forests: inconsistency, and a practical solution via the sobol-mda. *arXiv preprint arXiv:2102.13347*, 2021.

[7] Nicolai Meinshausen and Greg Ridgeway. Quantile regression forests. *Journal of Machine Learning Research*, 7(6), 2006.

[8] Yi Lin and Yongho Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474):578–590, 2006.

[9] Kevin Elie-Dit-Cosaque and Véronique Maume-Deschamps. Random forest estimation of conditional distribution functions and conditional quantiles. *arXiv preprint arXiv:2006.06998*, 2020.

[10] Clément Bénard, Gérard Biau, Sébastien Veiga, and Erwan Scornet. Interpretable random forests via rule extraction. In *International Conference on Artificial Intelligence and Statistics*, pages 937–945. PMLR, 2021.

[11] Erwan Scornet, Gérard Biau, and Jean-Philippe Vert. Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741, 2015.

Mathematical Programming Approaches for AI models Verification and Certification

Hatem Ibn Khedher [1]

[1] Capgemini engineering R&I France, 2 rue Paul Dautier, CS 90599, 78457 Vélizy Villacoublay

Abstract

Breaking deep neural networks with adversarial attack requires an intelligent approach that decides about the maximum allowed margin in which the neural network decision is invariant. We propose a new formulation based on linear programming approach modelling adversarial attacks. Then, propose an efficient technique for verifying deep neural networks properties and certifying the artificial intelligence model.

CONTEXT

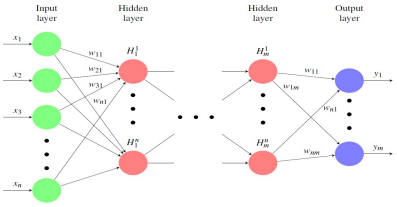
Neural Networks
Uncertainty in AI

OBJECTIVE

The goal of our project was to deploy, evaluate, model a robust AI models. This models could be used for future networks and applications. We defined our success according to the following objectives:

- **New formulation based on linear programming approach modelling adversarial attacks.**
- **Efficient technique for verifying neural networks properties**

Background



A Deep Neural Network is a extension of neural network with several hidden layers. It consists of three typical types of layers: an input layer, some hidden layers of neuron computations and an output layer. Each neuron is a simple processing element that responds to the weighted inputs it received from other neurons.

Trustworthy AI models: an adversarial attack

$$
\begin{aligned}
&\min Z = \left(\sum_{j\epsilon n} y_j - \sum_{j\epsilon n} \epsilon_j\right)\\
&\textit{Subject to :}\\
&\forall j \in L_1 : a_{in}(j) = x_j + \epsilon_j\\
&\forall j \in L_1 : a_{out}(j) = x_j + \epsilon_j\\
&\forall j \in L_k (2 \leqslant k \leqslant m-1) : a_{in}(j) = \sum_{i\epsilon\Gamma^-(j)} w_{ij} a_{out}(i)\\
&\forall j \in L_k (2 \leqslant k \leqslant m-1) :\\
&\quad a_{out}(j) \geqslant (\theta - 1)M + a_{in}(j)\\
&\quad a_{out}(j) \leqslant (1-\theta)M + a_{in}(j)\\
&\quad a_{out}(j) \leqslant \theta \times M\\
&\quad a_{in}(j) \geqslant (\theta - 1) \times M\\
&\quad a_{in}(j) \leqslant \theta \times M\\
&\quad \theta \in \{0,1\}\\
&\forall j \in L_m : a_{in}(j) = \sum_{i\epsilon\Gamma^-(j)} w_{ij} a_{out}(i)\\
&\forall j \in L_m : a_{in}(j) = \sum_{i\epsilon\Gamma^-(j)} w_{ij} a_{out}(i) \leqslant \beta\\
&a_{0j} \leqslant x_j \leqslant b_{0j}\\
&\epsilon_j \leqslant y_j \times (b_{0j} - x_j)\\
&\epsilon_j \geqslant y_j \times (a_{0j} - x_j)\\
&a_{0j} - x_j \leqslant \epsilon_j \leqslant b_{0j} - x_j\\
&y_j \in \{0,1\}
\end{aligned}
$$

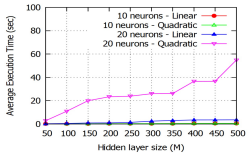
optimal adversarial attack approach that tells you about the minimal perturbation that brakes the AI model.

Trustworthy AI models: certification

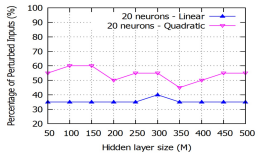
$$
\begin{aligned}
&\max Z = Constant\\
&S.T. :\\
&\forall j \in L_1 :\\
&a_{in}(j) = x_j\\
&a_{out}(j) = x_j\\
&\forall j \in L_k (2 \leqslant k \leqslant m-1) :\\
&a_{in}(j) = \sum_{i\epsilon\Gamma^-(j)} w_{ij} a_{out}(i)\\
&a_{out}(j) = \max\{a_{in}(j); 0\}\\
&\forall j \in L_m :\\
&a_{in}(j) = \sum_{i\epsilon\Gamma^-(j)} w_{ij} a_{out}(i)\\
&a_{in}(j) = \sum_{i\epsilon\Gamma^-(j)} w_{ij} a_{out}(i) \leqslant \beta\\
&a_0 \leqslant x_j \leqslant b_0
\end{aligned}
$$

Then, verify and certify you AI model.

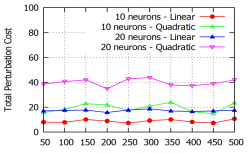
Testing and Evaluation: adversarial attack



Execution time (sec)



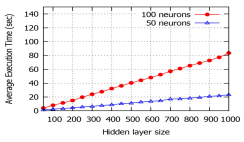
Perturbed inputs (PPI)



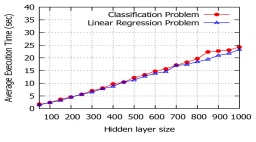
Perturbation cost TPC)

- The linear approach is slightly efficient
- in terms of perturbation cost.
- It outperforms the quadratic one in terms of PPI
- The efficiency and feasibility of our algorithm is justified in terms of average execution time

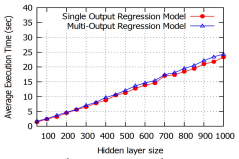
Testing and Evaluation: Certification



Convergence time



Neural network models



Single versus multi outputs

- The rapid convergence to optimal solutions even for large problem instances.
- The scalability of our proposed Big-M-based approach to the optimal solution under different scenarios and NN models

CONCLUSION

We proposed a new optimization technique for adversarial attack process. We considered the integration of new constraints such as the number of perturbed inputs
Then, we examined the safety of neural network models against input perturbations i.e. in an uncertain environment.

PERSPECTIVE

- Extend our modelling to other deep learning architectures such as Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM)
- Validate our proposed approaches on real use cases such as self-driving cars etc.
- Extend our study to other activation functions and DNN parameters

Acknowledgement :This research work has been carried out in the framework of IRT SystemX, Paris-Saclay, France, and Capgemini Engineering

Contact : hatem.ibnkhedher@altran.com

Capgemini engineering

The Data Curation Canvas – a Data-centric Approach to Trustful AI

Dr. Stefan Suwelack, Renumics GmbH

Abstract

We present the Data Curation Canvas as a template for collaboratively curating optimal training data sets. This hands-on method empowers engineers to develop more trustful ML models for applications in engineering and manufacturing in a data-centric way.

Introduction

Data-driven tools methods have become valuable tools in engineering and manufacturing. They allow to speed-up development cycles and to optimize complex production systems. Applications include anomaly detection and root cause analysis for test and simulation data, efficient surrogate modeling for complex simulations, better quality control as well as automation of manual modeling tasks. A key success factor for these tools is the trustworthiness of the ML-methods. In this context, the reliability and the explainability of ML models are especially important. There are three reasons for this:

1. ML-based solutions typically contain interactive interfaces: Users can override or correct ML-based suggestions. In these scenarios it is very helpful to know how reliable the results of the ML model are for a specific datapoint.
2. The introduction of data-driven methods into engineering workflows is a huge change management task. Experienced engineers need to develop an understanding of the capabilities and limitations of ML-based tools. Methods that illustrate and explain the results of data-driven algorithms can be very helpful for this.
3. The use of data-driven methods in highly regulated or safety-critical environments is still very limited. The lack of reliability and explainability are the most important hurdles for adoption.

Methods

Significant research has been carried out in the context of explainability and uncertainty quantification of ML models [1]. The developed methods can be categorized into model-based approaches (white box machine learning models) and post-hoc methods that are used in combination with traditional black box models [2].

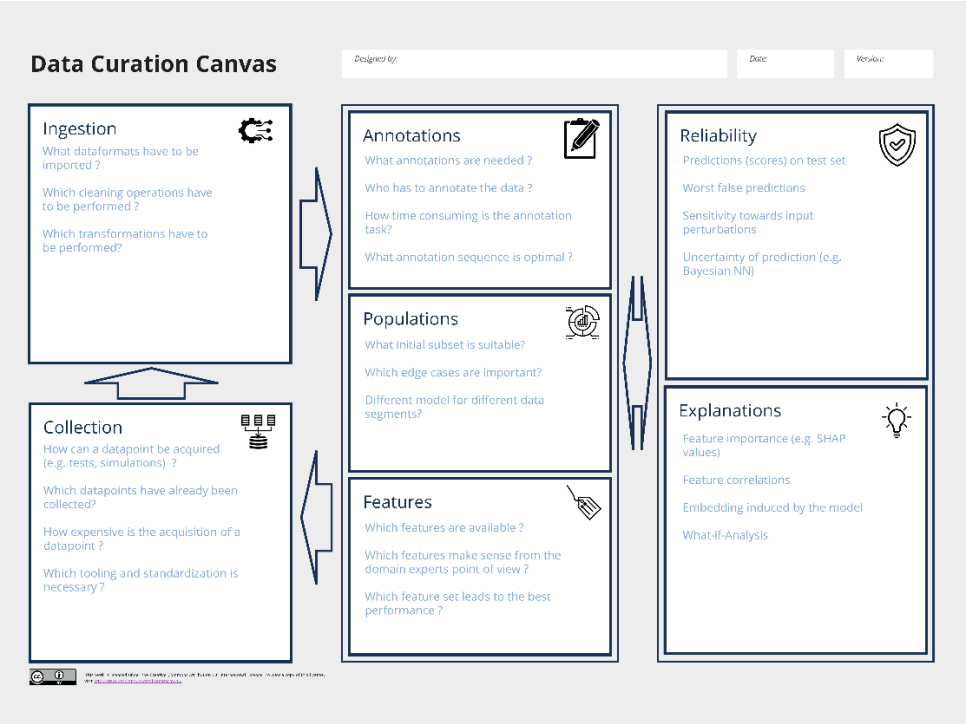


Fig.1: Domain experts and data scientists create a trustful ML model by iterating through the cycle outline by the Data Curation Canvas

We build on top of established post-hoc methods to formulate a pragmatic data-driven workflow for developing trustful ML-models. A key idea is to empower domain experts and data scientist to collaboratively curate optimal training data sets by monitoring the performance of the model. In this context, the monitoring is based not only on model predictions, but on additional information such as similarities, uncertainties or feature importances. Similar workflows have been successfully implemented in the development of data-driven driver assistance systems [3]. We formalize the approach with the “Data Curation Canvas” (Fig. 1). This collaborative template draws inspiration from canvas-based methods that are very popular for agile development of products and business models [4].

The goal is to establish a cycle between data Collection, Ingestion and model validation in a way that the dataset (and in turn the ML model) is improved until a desirable level of reliability is reached. The main task within the framework is for the data scientist and the domain experts to identify suitable training datasets: They must identify populations (i.e. segments) in the data and decide which range of populations the ML-tool has to cover and which training data has to be collected. Furthermore, they must identify robust features that can be used by the ML method and (in the case of supervised learning) they need to establish robust annotations of the datapoints. To do that, they use information from one or several ML models. For that purpose, they not only use raw predictions, but other types of information with regards to the Reliability and Explainability of the model. In terms of Reliability this includes information about the sensitivity of the output towards input perturbations as well as information about the aleatoric and epistemic uncertainty (e.g. through Bayesian neural networks). Furthermore, post-hoc Explanations such as feature importances (e.g. through Shapley-values) or similarity measures (e.g through embeddings) are used. This analysis benefits from the possibility to bring all this information into a single interactive application (Fig. 2).

Fig.2: Prediction scores, ground truth values and embeddings are used to qualitatively understand and improve a training dataset for 3D object classification.

Bibliography

[1] Arrieta, Alejandro Barredo, et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." Information Fusion 58 (2020): 82-115.
[2] Kenny, Eoin M., et al. "Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies." Artificial Intelligence 294 (2021): 103459.
[3] Karpathy, Andrei, “Keynote“, Workshop on Autonomous Driving, CVPR 2021
[4] Osterwalder, Alexander, and Yves Pigneur. Business model generation: a handbook for visionaries, game changers, and challengers. Vol. 1. John Wiley & Sons, 2010.

Fostering Trust in Software Assistants for Software Engineering

Maxime Savary-Leblanc
UMR 9189 CRIStAL
Univ. Lille, CNRS, Inria, Centrale Lille
Lille, France
maxime.savary-leblanc@univ-lille.fr

Xavier Le-Pallec
UMR 9189 CRIStAL
Univ. Lille, CNRS, Inria, Centrale Lille
Lille, France
xavier.le-pallec@univ-lille.fr

Sébastien Gérard
Université Paris-Saclay, CEA, List
F-91120, Palaiseau, France
sebastien.gerard@cea.fr

***Abstract*—Software engineers must cope with a broad range of development tasks, in addition to mastering a business domain that is sometimes unknown at first. A solution to alleviate their workload is to integrate Software Assistants in their development environment, offering new knowledge and features. However, to achieve successful collaboration, software engineers must trust these systems and their suggestions. This poster identify different research lines on fostering trust between software engineers and their software assistant, and introduces our solution to build trust-centered assistants applied for software modeling.**

***Index Terms*—Trust, Modeling Assistant, Software Engineering**

I. Introduction

Software Engineering is a broad term covering a wide variety of tasks, described in the ISO12207 standard [1]. Although this software lifecycle is often distributed today between several teams, the work of software engineers consists in the mastering of various tasks, framed with company-specific productivity criteria. At the same time, software engineers must learn to master complex business domains to achieve the expected software systems. These tasks require knowledge that software engineers must acquire during their career and apply by working with software tools.

While tools to support software engineering have not stopped evolving since their appearance around the 1960s, they still struggle to integrate knowledge in a form that could help software engineers. For example, debugging tools have become common in most development tools nowadays, and facilitate a specific task in the development work. In particular, they allow to execute a program step by step, and to locate an error in the code. However, in order to understand an error, or the behavior of a piece of code, software engineers must regularly quit their development environment to find their answer online, as for example on StackOverflow [2]. These context switches are responsible for a decrease in productivity, concentration, and satisfaction. They occur on all tasks of the software life cycle as soon as it involves a software tool.

Software assistants are a solution to bring knowledge into the working environment of software engineers. However, for them to be useful, they must be accepted by software engineers [3], what depends on the notion of *trust*. Software assistants must display trust indicators and provide specific behavior to enable trust to grow in the human-assistant collaboration . The criteria for building this relationship of trust between the system and the user apply to all levels of the system, including the knowledge base, the algorithm, and the user interface. They must be discussed and considered at the system design stage in order to be implemented.

II. Background

A. Supporting Software Engineering: Software Assistants

Tools for programmers naturally exist since the beginning of Software Engineering around 1960. At that time, they were single tools focused on some specific tasks of the SE life cycle, cumbersome to use, and acting in isolation of each other. A new wave of systems then gradually replaces tools with more comprehensive functionalities, and fall under the emerging field of Computer-Aided Software Engineering (CASE) tools, which lay the foundation for modern-day IDEs. As environments improve, other issues emerge such as the need for collaboration to produce ever more complex systems, which paves the way for the Computer-Supported Cooperative Work (CSCW) community and more specifically the Collaborative Software Engineering (CSE) community. The CSE community then seeks to enhance environments to cope with different forms of collaboration.

During the 90's, the *agent* research fields explodes and brings to light a new opportunity for collaboration: that with the machine acting as an autonomous system with which users (or other agents) could interact and work.Some agents are refined into *intelligent agents* that are reactive, proactive, and social agents tailored for human-agent collaboration. However, due the lack of computing resources and/or data to exploit, such agent-based systems never became mainstream in Software Engineering [4].

Since, the broad Software Agent community has remained active, and has branched into several sub-categories. Particularly, the notion of *conversational agent* (a.k.a. bot or chatbot) is gaining importance in the last years, and has appeared in the sectors of customer support or video game [4], [5]. In 2016, Storey and Zagalsky laid the foundation for research on bots in software engineering and described how bots are increasingly used to support tasks that traditionally required human intelligence [6]. It has particularly been applied to Software Engineering to create *BOTse* [7] or *DevBots* [4] (bots for Software Engineering) [8]. A consensual definition established

during the BOTse Dagstuhl seminar in 2020 [7] defines bots as systems featuring at least one of the following characteristics: (i) automates one or more feature(s), (ii) performs one or more function(s) that a human may do, (iii) interacts with a human or other agents.

At ICSE'06, Boehm predicted a new kind of developer-helping systems for 2020 as "that provide feedback to developers based on domain knowledge, programming knowledge, systems engineering knowledge, or management knowledge" [9]. The description of previous bot systems is almost inline with these expectations but still lacks one essential characteristic that Boehm described as "the use of knowledge". Storey et al. [6] identify bots embedding knowledge as one specific type of bots. Thus, knowledge appear as an inflexion point, which opens the way for the study of a specific type of systems–knowledge-empowered DevBots–that we will call software assistants for Software Engineering.

B. Trust in Software Assistants

Software assistants aim to embed knowledge into the work environment of software engineers. Thus, they are capable of providing suggestions and recommendations, and can be perceived as specific recommender systems. As choices made by designers during modeling do not reflect their personal tastes but rather the project constraints to respect, software assistants might rely on a hybrid of content-based and knowledge-based approaches.

To the best of our knowledge, nothing in available research literature addresses the notion of trust for such hybrid recommender systems. We analyzed multiple prior works to help us frame our approach. Because recommender systems are information systems, we gathered articles linking both information and trust [3], [10] and cross-referenced them against literature pertaining to trust and evaluation of recommender systems [11], [12]. To analyse the contents of all these papers, we constructed a model (shown in Figure 1), which represents a consolidated conceptual model of trustworthiness in recommender systems. (Note: The greyed-out elements in this diagram are not relevant to the discussion in this poster.)

In this initial work, we focused on the impact of Perceived Usefulness, Perceived Transparency, and Perceived Control towards increasing the trustworthiness potential of our system. In particular, we addressed the following characteristics: *Information Trustworthiness*, *Information Transparency*, *System Transparency* and *System Control*.

III. Poster content

This poster presents a discussion of the application of information trustworthiness to the design of Software Assistants, based on the literature related in Section II-B. It also presents statistics collected during our systematic literature review of software assistants for software engineering, and our modeling experts interviews. Finally, it presents our approach to the design of software assistants applied to software modeling [13]. We propose a link[1] to our prototype of software modeling assistant for the Papyrus[2] modeling tool, based on Eclipse and developed by the CEA.

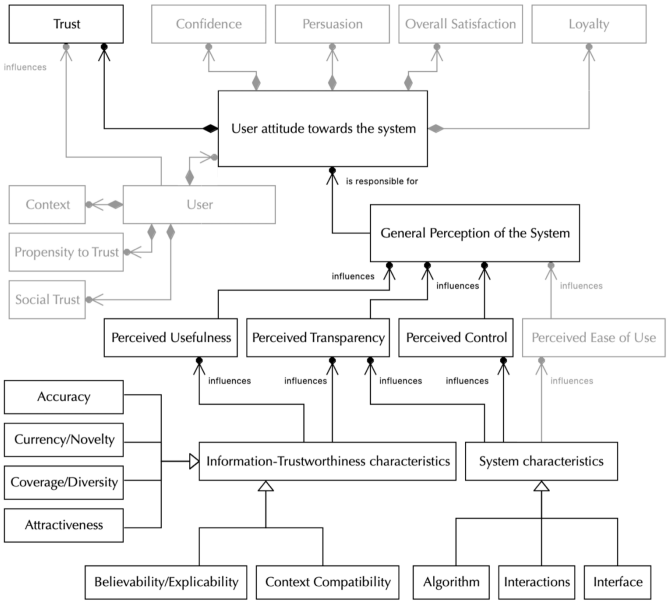


Fig. 1. A conceptual model of trustworthiness in recommender systems.

References

[1] "ISO/IEC 12207:1995/Amd 2:2002 Information technology – Software life cycle processes, year = 2002," Tech. Rep.

[2] Y. Wu, S. Wang, C.-P. Bezemer, and K. Inoue, "How do developers utilize source code from stack overflow?" *Empirical Software Engineering*, vol. 24, no. 2, pp. 637–673, Apr. 2019.

[3] M. Hertzum, "The importance of trust in software engineers' assessment and choice of information sources," *Information and Organization*, vol. 12, no. 1, pp. 1–18, Jan. 2002.

[4] L. Erlenhov, F. Gomes de Oliveira Neto, R. Scandariato, and P. Leitner, "Current and Future Bots in Software Development," vol. BotSE 2019, May 2019, pp. 7–11.

[5] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, "The rise of social bots," *Commun. ACM*, vol. 59, no. 7, p. 96–104, Jun. 2016. [Online]. Available: https://doi.org/10.1145/2818717

[6] M.-A. Storey and A. Zagalsky, "Disrupting developer productivity one bot at a time," ser. FSE 2016, vol. 2016 Symposium on Foundations of Software Engineering, 2016, p. 928–931.

[7] M.-A. Storey, A. Serebrenik, C. P. Rosé, T. Zimmermann, and J. D. Herbsleb, "BOTse: Bots in Software Engineering (Dagstuhl Seminar 19471)," *Dagstuhl Reports*, vol. 9, no. 11, pp. 84–96, 2020.

[8] S. Pérez-Soler, E. Guerra, and J. de Lara, "Collaborative modeling and group decision making using chatbots in social networks," *IEEE Software*, vol. 35, no. 6, pp. 48–54, 2018.

[9] B. Boehm, "A view of 20th and 21st century software engineering," ser. ICSE '06, vol. ICSE 2006. New York, NY, USA: Association for Computing Machinery, May 2006, pp. 12–29.

[10] K. Chopra and W. Wallace, "Trust in electronic environments," in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, Jan. 2003, pp. 10 pp.–.

[11] P. Pu, L. Chen, and R. Hu, "A user-centric evaluation framework for recommender systems," in *RecSys 2011*, ser. RecSys '11, Oct. 2011, pp. 157–164.

[12] N. Tintarev and J. Masthoff, "A survey of explanations in recommender systems," in *2007 IEEE 23rd International Conference on Data Engineering Workshop*, 2007, pp. 801–810.

[13] M. Savary-Leblanc, X. Pallec, and S. Gérard, "A recommender system to assist conceptual modeling with uml," in *SEKE 2021*, 06 2021.

[1] https://youtu.be/6mxoJOhwgbk

[2] https://www.eclipse.org/papyrus/

Domain adaptation for COVID-19 detection on lung ultrasound imaging

Alexandre Autret[a], **Armin Dietz**[a], **Lorena Gayarre Peña**[a], **Gaël Richard**[a], **Alexandre Carlot**[b], **Clément Le Couedic**[b], **Mehdi Benchoufi**[b,c] **and Elsa D. Angelini** [d]

[a] Capgemini Engineering, France. Authors contact: alexandre.autret@edu.devinci.fr and armin.dietz@capgemini.com
[b] echOpen factory, 67 rue Saint-Jacques, 75005 Paris, France
[c] Université de Paris, Centre of Research in Epidemiology and Statistics (CRESS), French Institute of Health and Medical Research (INSERM), National Institute of Agricultural Research (INRA), Paris, France
[d] LTCI, Telecom Paris, Institut Polytechnique de Paris, France

Abstract

Medical imaging is often susceptible to significant experimental bias, for example by the use of different hardware and protocols. Combined with the fact that medical datasets are often small, due to the difficulty of collecting sensitive data, applying deep learning models raises several challenges. We report preliminary results on forcing a CNN classifier to learn domain-invariant representations of COVID-19 features on lung ultrasound images by using domain adaptation.

Introduction

One of the most common difficulties with training a classifier on medical images is the large domain shift introduced by data being collected in various hospitals: different hardware, protocols, and setups can lead to very different data, and it can be difficult to ensure that a given classifier learns features robust enough to handle a new source of data. A traditional approach for solving this problem has been domain adaptation. Schoenauer-Sebag et al. designed an adversarial domain adaptation architecture, MuLANN[1], based on DANN[2], for microscopy data from varying experimental setups. In this study we evaluate the benefits gained from MuLANN to classify lung ultrasound images for COVID-19 detection based on a training cohort from two probes used in two different hospitals.

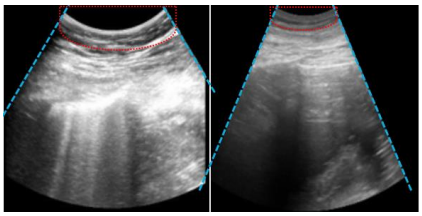


Fig. 1: Images from two different ultrasound probes, recorded at two different hospitals.

Methods

Our dataset was comprised of 1,288 lung ultrasound videos from 161 patients, coming from 2 different hospitals and probes (see figure 1). Eight videos were acquired per patient, at 8 distinct locations in the chest. Binary labels were provided per-video by clinical experts, and defined as 0=healthy and 1=pathological. A single patient could then have videos labelled as healthy and others as pathological. A maximum intensity projection was applied to encode the videos into static images, to save computational resources and to highlight artifacts characteristics of COVID19 lesions.

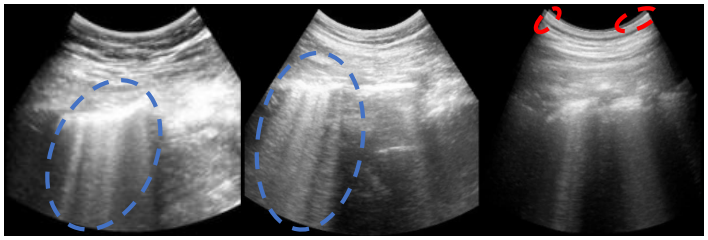


Fig. 2: Real sample (left) compared to two synthetic samples. Middle one show how ADASYN preserves important artifacts, while the one on the right shows the only issue certain synthetic samples would have: minor interpolation issues on the edges.

Regular oversampling of the minority class was also tested, but the model would still consistently predict most datapoints as belonging to class 0. ADASYN was therefore necessary, and was applied to each probe type (domain) separately, as they have specific image resolution, and combining them would lead to interpolation issues (see figure 2).

We compared two CNN classifier models: A traditional VGG-16[4]-based CNN with pretrained convolution layers on the ImageNet dataset, and the same architecture extended with an adversarial module for domain adaptation as described in MuLANN[1]. Each model was trained for 25 epochs using stochastic gradient descent, with a data split of 80-20% for training and validation sets, stratified by classes and domains.

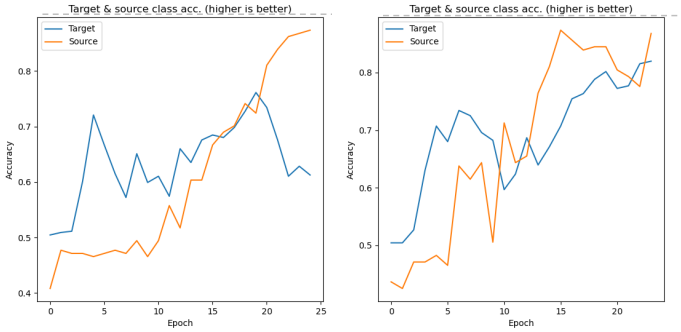


Fig. 3: Validation accuracy per epoch per domain without domain adaptation (left) vs. MuLANN (right).

Results

Domain adaptation significantly improved overall validation accuracy, from 68.6% to 78.6%. As shown on the accuracy curves in figure 3, our model without domain adaptation would very effectively learn from data from one probe, but wouldn't learn from the other at all. With MuLANN, our model effectively learned on data from both probes simultaneously.
To further visualize the effectiveness of domain adaptation, we performed a t-SNE[5] projection on the model's latent space and colored each point in the whole-cohort based on its class (figure 4) and domain (figure 5). With MuLANN, we can see a distinct boundary between classes while there is some large overlap without domain adaptation.

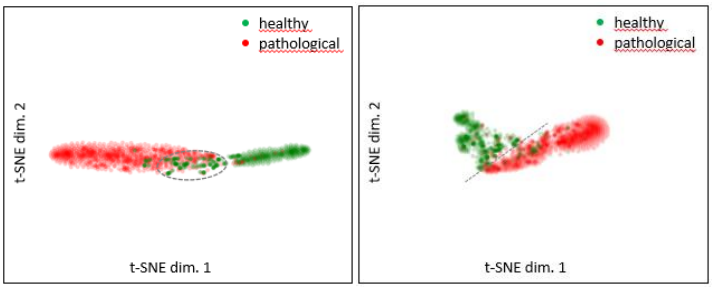


Fig. 4: t-SNE of model's latent space at last epoch, with each point colored w.r.t its class (green=healthy, red=pathological). On the left, without domain adaptation, On the right, with MuLANN.

When we analyze the features by corresponding domains, we can see that without domain adaptation, the model seems to have learned to discriminate the domains, in separate areas within class-specific clusters. With MuLANN, the domains appear more evenly distributed in both clusters.

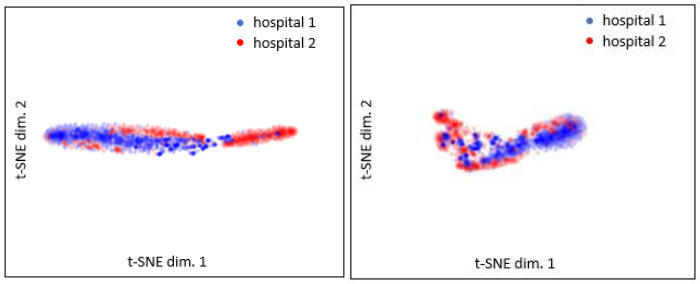


Fig. 5: t-SNE of model's latent space at last epoch, with each point colored w.r.t its domain (blue=hospital 1, red=hospital 2). On the left, without domain adaptation, on the right, with MuLANN.

Discussion

As shown by the accuracy curves, without domain adaptation, the model learned very effectively on source domain data (orange), at the expense of the target domain (blue).
MuLANN significantly improved target domain accuracy, at the cost of slightly reducing source domain accuracy. One possible interpretation is that a domain-invariant representation of data made the model perform similarly across domains, rather than perform well on one domain on which it may rely on domain-specific features, at the cost of reduced performance on other domains, as was the case without domain adaptation.
The fact that the learning curves for each domain evolve similarly seems consistent with this idea.
As shown by the t-SNE plots, without domain adaptation, the model seemed to discriminate domains as much classes, which wasn't the case with MuLANN.

Conclusion

Architectures such as DANN and MuLANN can be effective on medical imaging, and domain adaptation should be considered whenever a model is expected to be used on medical data from multiple sources. However, domain adaptation alone may not always be sufficient, especially on small datasets, so data augmentation techniques can be used to supplement it, such as ADASYN to counter class imbalance. It could be interesting to study the effect of other data augmentation techniques like SamplePairing[6] and MixUp[7], which can be used to generate additional samples and ensure that more robust features are learned.

References

[1] Schoenauer-Sebag et al., "Multi-Domain Adversarial Learning", 2019. URL https://arxiv.org/pdf/1903.09239.pdf
[2] Ganin, Yaroslav, et al. "Domain-adversarial training of neural networks." The journal of machine learning research 17.1 (2016): 2096-2030. URL https://arxiv.org/pdf/1505.07818.pdf
[3] He, Haibo, et al. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning." 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). IEEE, 2008.
[4] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition.", 2014. URL https://arxiv.org/pdf/1409.1556.pdf
[5] Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.11 (2008).
[6] Inoue, Hiroshi. "Data augmentation by pairing samples for images classification." (2018). URL https://arxiv.org/pdf/1801.02929.pdf
[7] Zhang, Hongyi, et al. "mixup: Beyond empirical risk minimization.", 2018. URL https://arxiv.org/pdf/1710.09412.pdf

Is active learning better than random selection for real-word tasks ?

Fritz Poka Toukam, Thomas Dalgaty, Hedi Ben-Younes
Nicolas Granger, Spyros Gidaris, Camille Dupont, Oriane Simeoni
EC5-FA6

Abstract

In real-world problems, datasets typically do not come with a complete set of annotations. Annotation is a very expensive process for industry. This is the case in particular for object detection tasks; an annotator must locate all objects in an image with a bounding box and give each a label. In order to reduce costs, different schemes can be put in practice, e.g. self-supervision and semi-supervision methods. Deep Active Learning is a another relevant strategy whereby an acquisition function is used to select a subset of the most *useful* images within a large unlabelled pool to be labelled by an expert. Typically, active learning has been applied to datasets that do not necessarily reflect well the characteristics of those found in industrial problems. We propose here to apply active learning in the object detection setting to more realistic datasets of driving scenes, and to study its combination with unsupervised learning paradigms.

Introduction

The characteristics of real-world machine learning problems often do not correspond to the tasks used for benchmarking within the artificial intelligence community. Notably, real-world datasets are often composed of a large amount of unlabelled data. Labelling data is expensive and time consuming, and therefore it is not a practical route towards deploying an industrial system. Some unsupervised machine learning paradigms, namely self-supervised [3, 1] and semi-supervised [10, 6] learning, offer a means to make use of this unlabelled data.

Another idea is active learning [2] (Fig.1 (left)), which is an approach that seeks to train a model using a small labelled subset of the available data. This occurs over several active learning *cycles*. A model is trained on an initial *seed* of annotated images that are, most often, randomly selected.

All of the unlabelled data-points are then assessed using an *acquisition function* that aims to score each unlabelled data-point based on some measure of uncertainty [4] or diversity [9]. The images with the highest score, ideally corresponding to the most relevant images to the model, are then labelled by an *oracle* and added to the labelled pooled used in the next cycle of training. During successive active learning cycles the model will gradually improve as it integrates more of the data. Ideally, after a certain number of cycles (or after a labelling budget has been exhausted) the performance of the model should approach that of a model trained through a supervised approach - normally requiring only a fraction of the labels.

While active learning has been applied successfully to image classification, many questions remain: how can diversity be ensured in the labelled set, how to account for class imbalance, how will these methods extend to real world problems? Furthermore, more complex, and potentially more industrially relevant tasks and models, for example instance segmentation and object detection, and their combination with unsupervised learning approaches have not been well explored.

We aim to apply active learning to object detection in real-world industrial datasets (i.e., Woodscape [11] - Fig.1(right)). Additionally, we propose to investigate how unlabelled data can be best leveraged via the combination of active learning with unsupervised learning approaches.

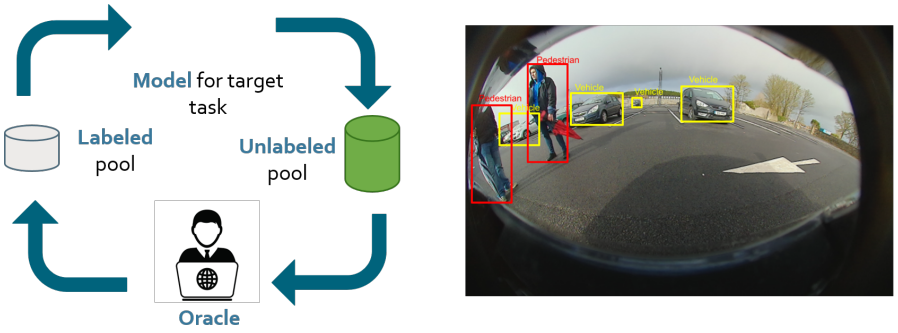


Figure 1: (left) A cartoon of the active learning cycle. (right) An example real-world data point (from Woodscape [11])

Some first results

As a starting point for our experiments, we apply a fast object detection model widely used in industry (YOLOv5 [5]) to object detection. We report results from two different scenarios: (i) we use a YOLOv5 model with random initialisation, (ii) we pre-train the model on the COCO dataset [8]. The COCO dataset shares common classes to our datasets of interest, namely the classes *person* and *car*, however the type of data are significantly different. We investigate several active learning baselines on the BDD100K dataset [12] - an autonomous driving dataset with 10 classes and 70 000 images in the training set. We consider a budget of 700 images per cycle and perform 10 active learning cycles, therefore using at most 7000 images. The figure 2 shows the mean Average Precision at 50% of IoU (mAP50) obtained with the model trained on the labelled set at the end of each cycle. We observe that, in both scenarios, the best results are achieved with the maximum uncertainty selection acquisition function which scores each data-point based on the highest entropy calculated amongst detected objects. Furthermore, When using the COCO pre-training, we achieve 52.5 mAP with 10% of the dataset vs 59.6 mAP with the full dataset. Without pre-training we achieve 49.5 mAP with 10% vs 58.3 with the full dataset. Results with active learning are also significantly better than when random selection is performed.

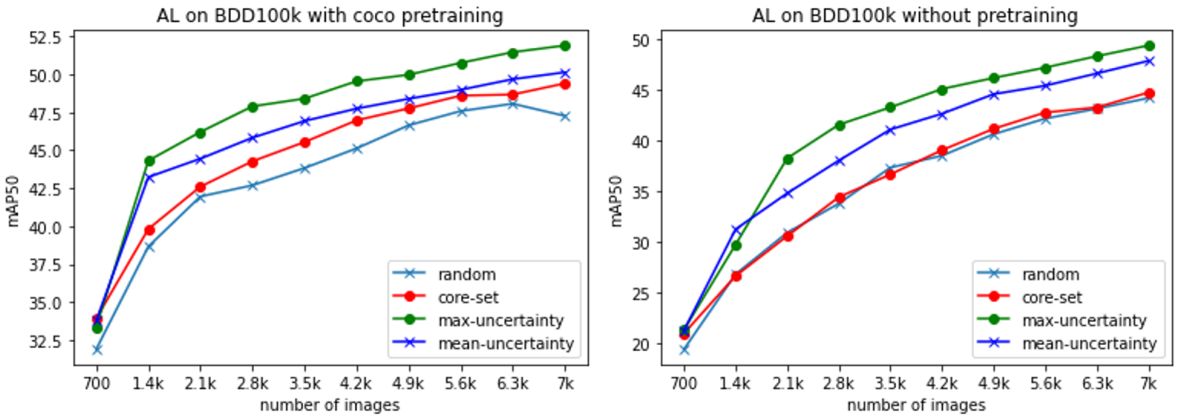


Figure 2: Plots of the mean average precision (mAP) at 50% intersection-over-union, with the number of labelled images for (left) a model pre-trained on the COCO dataset [7] and (right) a model with no pre-training.

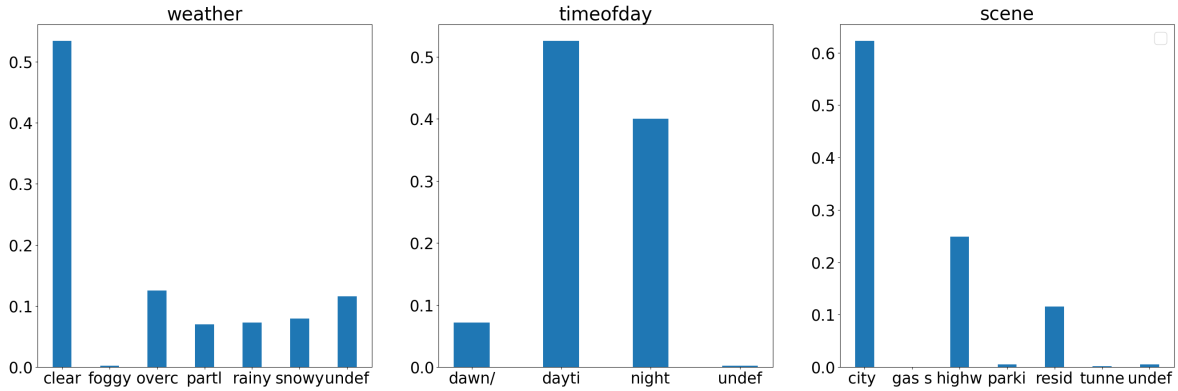


Figure 3: Distribution of the three types of meta-data over the training set of the BDD100k dataset.

Perspectives

We achieve promising results on the large dataset BDD100k with classic active learning methods. We are currently investigating how those results can be improved, and in particular how unlabelled data can be best integrated to the active learning setup. Self-supervised and semi-supervised methods are known to achieve great results by themselves, but it is less clear how active learning can best benefit from such paradigms. We also consider using the meta-data stored during the image acquisitions. In particular, the Figure 3 shows the distribution of values over the training set for the three different type of meta-data proposed in the dataset BDD100k. We are currently testing methods to best incorporate this information.

References

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[2] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

[3] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[4] Y. Gal, R. Islam, and Z. Ghahramani. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR, 2017.

[5] G. Jocher. ultralytics /yolov5.

[6] D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and L. Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

[9] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[10] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[11] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricár, S. Milz, M. Simon, K. Amende, et al. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9308–9318, 2019.

[12] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning, 2020.

TESTING MACHINE LEARNING ALGORITHMS IN PREDICTIVE MAINTENANCE

Version	1.0	
Auteurs	**Thomas DELORME**	**NAVAL GROUP/SER/KTM/FASTLAB**
	Jérémy TRIONE	**NAVAL GROUP/SER/KTM/FASTLAB**
	Laurence GUILLON	**NAVAL GROUP/SER/KTM/FASTLAB**
	Anthony ROSSI	**NAVAL GROUP/SER/KTM/FASTLAB**
Date	**24/09/2021**	

Division Services
Cosin / Data Sciences
N° / Reference : Naval Group Toulon - BP 517 - 83041 Toulon Cedex 9

NAVAL GROUP

NAVAL GROUP

ABSTRACT

The first experiments in predictive maintenance done for aircraft engines almost 10 years ago have attracted all industries. The average annual growth rate of predictive maintenance was 53.65% between 2015 and 2020 in France.

With this emergence of predictive maintenance, machine learning and artificial intelligence approaches have been extensively applied in industry for handling the health status of equipments.

Time series anomaly detection has been an important topic in data science, with papers dating back to the 1950s.
In recent years, there has been an explosion of interest in this topic, much of it driven by the success of deep learning in other domains and for other time series tasks.

Massive amounts of operational and processes conditions data are generated from several equipments, but most data is not complete, has not good quality and are poorly labelled.
Most of time series available in industry doesn't have any seasonality, cyclicity or evident pattern.

The objective of the poster is to test several Machine Learning technics for anomaly detection on problematic univariate time-series with minimal pre-processing on data.

Algorithms tested are:
- Clustering based methodologies:
  - DBSCAN
- Classification based:
  - Isolation forest;
  - OneClassSVM;
- Prediction based:
  - Auto-ARIMA;
  - TBATS;
  - Quantile Gradient Boositng.
- Deep Learning based:
  - LSTM Autoencoders.

Classical methods usually used for anomaly detection on time series have a much more questionable behavior when the data series are coming from heterogeneous operating situations.
Most of machine learning approaches show their limits due to the nature of the data, and get easily outscaled by more basic methods like defining a threshold.

Keywords: predictive maintenance; machine learning; anomaly detection; time series

Predictive Uncertainty Quantification for Time Series

Marc Nabhan, Air Liquide R&D
Kevin Pasini, IRT SystemX
Luca Mossina, IRT Saint Exupéry

September 2021

Abstract

Many machine learning models output a single prediction without any measure of uncertainty, or with hardly interpretable ones. In the context of the project EC3 in Confiance.ai, the fifth action sheet, called "Predictive Uncertainty Quantification", aims at associating meaningful and rigorous measures of uncertainties, such as prediction intervals, to regression tasks. They are applied to a time series use case provided by Air Liquide.

1 Air Liquide Use Case

Among the first industrial use cases accepted in Confiance.ai, Air Liquide proposed a time series use case focused on forecasting customers' demand. In fact, the production units at Air Liquide should guarantee that all customers are always in supply. Therefore, predicting the future trends of customer demands based on historical data, as precisely as possible, is highly useful for the production sites and dispatchers.

The use case includes a dataset on historical consumption, customers and orders data, geographical distribution, seasonality and contextual data, as well as an XGBoost regression model predicting future values of the target variable. However, this model provides no measure of uncertainty, which can have a great impact on production and operations, leading to energy loss or customer dryout. Since this is a regression problem, we aim to construct Prediction Intervals (PI) to quantify uncertainty in the forecasts. These intervals can be interpreted as error margins on the predictions, yielding upper and lower bounds containing the true labels with high probability. If they are guaranteed, they can increase the trust in the predictions of the single-value forecast model.

2 Predicting Uncertainty Intervals

For a time series Y (Observations) associated with a series of explanatory variables X (Features) and for an $(1-\alpha)\%$ confidence requirement: the aim is to build PIs, by estimating the uncertainty associated with an observation y from these attributes x, and a miscoverage probability α. As uncertainty can vary according to features and is not directly observed, estimation will often build upon the variability in the forecasting residuals as an intermediate to reach uncertainty.

Our first experiment consists in evaluating a forecasting model (Gradient Boosting Regressor) several ways, to perform Uncertainty Quantification (UQ) in order to build PIs. A Gradient Boosting Regressor is an ensemble learning model that combines several weak predictors (Decision tree) to perform regression. The Boosting mechanism aims to successively add weak learners (Tree) to correct errors of the current model that combines the weak models learned so far.

Gradient Boosting can also be used to quantify uncertainty by performing quantile regression using the quantile loss, which penalizes positive and negative errors in a antisymmetric way in order to approximate quantile $q_\beta(x) = \inf\{y : F(y \mid X = x) \geq \beta\}$. By fixing lower and upper thresholds α_{lo} and $(1-\alpha_{hi})$ so that $\alpha_{lo} + \alpha_{hi} = \alpha$, we can estimate the PI as $C_{GB}(x) = [\widehat{q}_{\alpha_{lo}}(x), \widehat{q}_{1-\alpha_{hi}}(x)]$.

Uncertainty Quantification can also be performed by Conformal Prediction (see Section 3). Our experiments aim to perform robust evaluation on real data of PIs quality obtained by the different UQ techniques through two metrics: the **Average Coverage Error** (ACE), the difference between the actual and expected global coverage, and the **Sharpness**, the average size of the prediction interval.

3 Conformal Prediction and Prediction Intervals

Conformal Prediction (CP, Vovk et al. 2005) is a set of distribution-free, non-asymptotic, model-agnostic methods to do UQ by constructing PIs whose probability coverages are backed by theoretical guarantees.

Specifically, given exchangeable training and test data[1] drawn from $P_{X,Y}$ and a fixed miscoverage probability $\alpha \in (0,1)$, CP yields a prediction interval $\widehat{C}^{\alpha}(\cdot)$ such that

$$\mathbb{P}\left(Y \in \widehat{C}_{\alpha}(X)\right) \geq (1-\alpha) \tag{1}$$

holds, on average. In particular, over many test predictions, $\widehat{C}_{\alpha}(X)$ will contain the true values Y with a frequency of at least $(1-\alpha)\%$. For example, the PI can be constructed from a regression function $\widehat{f}(\cdot)$ or a quantile regressor $\widehat{q}(\cdot)$ as:

$$\widehat{C}_{\alpha}(X) = \begin{cases} \widehat{f}(X) \pm \delta^{f}_{\alpha} & \delta^{f}_{\alpha} \geq 0, \quad \widehat{f}(\cdot) : \text{regression function} \\ [\widehat{q}_{\alpha_{lo}}(x) - \delta^{q}_{\alpha},\, \widehat{q}_{1-\alpha_{hi}}(x) + \delta^{q}_{\alpha}] & \delta^{q}_{\alpha} \in \mathbb{R}, \quad \widehat{q}(\cdot) : \text{quantile regression function} \end{cases}$$

The quantities δ^{f}_{α} and δ^{q}_{α} are derived from the $(1-\alpha)$-th empirical quantile of specifically designed regression residuals, known as nonconformity scores, proper to each CP algorithm, computed by evaluating the predictor on held-out calibration data. The width of the interval is tied to the quality of the predictor and the uncertainty of the phenomenon, and can be used to quantify uncertainty. Since Equation 1 is guaranteed to hold for any prediction model, CP can be extended to calibrate pre-trained models, given some new calibration data $D_{calibration} \sim P_{X,Y}$.

3.1 Conformalized Quantile Regression for Uncertainty Quantification

The state-of-the-art approach in CP is based on quantile regressors (Gupta et al., 2019). Here, we present the first results we obtained with **Conformalized Quantile Regression** (CQR) by Romano et al. (2019): after splitting disjointly the data as $D_{train} = \{D_{fit}, D_{calibration}\}$, we fit a Gradient Boosting quantile regressor on D_{fit}, and compute the nonconformity scores $R_i = \max\{\widehat{q}_{\alpha_{lo}}(X_i) - Y_i, Y_i - \widehat{q}_{1-\alpha_{hi}}(X_i)\}$ for every $(X_i, Y_i) \in D_{calibration}$, yielding $\bar{R} = \{R_i\}$.

For a new test point X_{new}, the CQR Conformal PI then boils down to:

$$\widehat{C}^{CQR}_{\alpha}(X_{new}) = \left[\widehat{q}_{\alpha_{lo}}(X_{new}) - Q_{1-\alpha}(\bar{R}),\, \widehat{q}_{1-\alpha_{hi}}(X_{new}) + Q_{1-\alpha}(\bar{R})\right], \tag{2}$$

where $Q_{1-\alpha}(\bar{R})$ is the $(1-\alpha)(1+\frac{1}{|D_{calib}|})$-th empirical quantile of $\bar{R}$.

Following the metrics specified in Section 2, we obtain a promising empirical coverage of 0.874 for a nominal $(1-\alpha) = 0.9$ and $ACE = |0.9 - 0.874| = 2.6\%$, despite the fact that our time series does not seem to comply with the hypothesis of data exchangeability, a topic currently under experimentation.

4 Conclusion and Perspectives

For clarity, we restricted the presentation to few UQ approaches (Quantile regression & CQR), although we are carrying out extensive benchmarks on others approaches. Stemming from these promising results, we are currently working towards:

- Further non-conformal UQ approaches (Bayesian Modeling, Variance Regression, ML sub-sampling estimation).
- Specific CP methods for time-dependent, non-exchangeable data (Xu and Xie, 2021)

References

Gupta, C., Kuchibhotla, A. K., and Ramdas, A. K. (2019). Nested conformal prediction and quantile out-of-bag ensemble methods. *arXiv preprint arXiv:1910.10562*.

[1] CP also applies to independent, identically distributed data.

Romano, Y., Patterson, E., and Candes, E. (2019). Conformalized quantile regression. *Advances in Neural Information Processing Systems*, 32:3543–3553.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic learning in a random world*. Springer.

Xu, C. and Xie, Y. (2021). Conformal prediction interval for dynamic time-series. In *Proceedings of ICML 2021*, volume 139, pages 11559–11569.

A Example of Conformal Prediction Intervals on Air Liquide use case

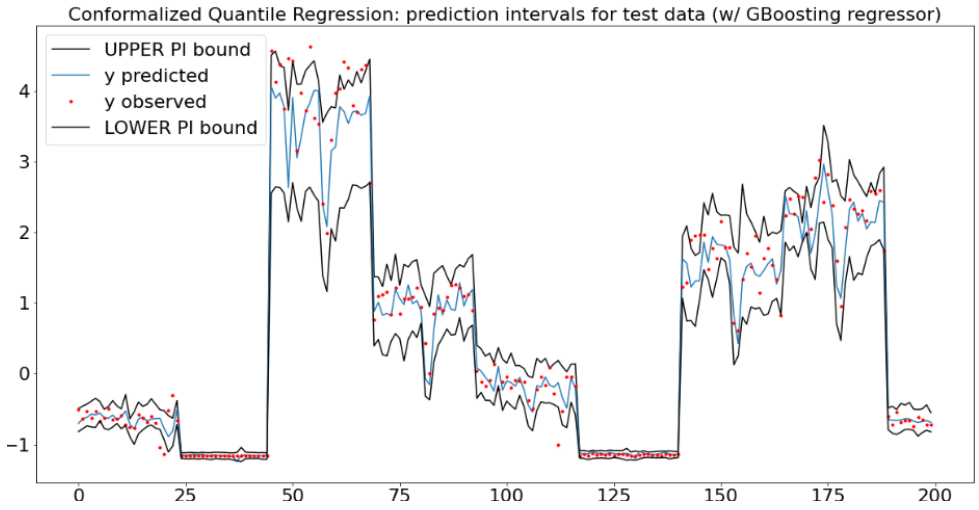


Figure 1: CQR prediction intervals, first 200 test points. Empirical coverage is 0.874 for $1 - \alpha = 0.9$, empirical average width of the interval is 0.671.

A cross-domain framework for Operational Design Domain specification

Guillaume Ollier[1], Morayo Adedjouma[1], Simos Gerasimou[2], Chokri Mraidha[1]
[1]*Université Paris-Saclay, CEA LIST, Dept. Ingenierie Logiciels et Systemes, P.C. 174, Gif-sur-Yvette, 91191 Cedex, France*
[2]*University of York, Department of Computer Science, United Kingdom*
guillaume.ollier@cea.fr

***Abstract*—This paper presents a method to generalize the concept of "Operational Design Domain" (ODD) used in the automotive domain to any cyber-physical system. The approach proposes to use domain-level and meta-theories taxonomies to develop a cross domain ontology for the definition of the ODD.**

***Index Terms*—ODD, Ontology, Autonomous system**

I. Introduction

To solve the challenge of the specification of the intended capabilities and limitations of Autonomous Systems (ASs) based on AI models, a solution is to capture the scenario-space covering all possible Usage Scenarios (USs) of the system. Such scenario-space is defined in the automotive domain with the concept of Operational Design Domain (ODD) [1]. Within ODDs, USs are decomposed into Operating Conditions (OCs) which might include environmental conditions (e.g, illuminance, weather, traffic), conditions on the ego system (e.g, speed limitations, maneuvers), etc. The OC terms and their relations can be formalized through an ontology, i.e, *"a representation, formal naming, and definition of the categories, properties, and relation between the concepts, data and entities that substantiate one, many, or all domains of discourse"* [2]. In other words, an ontology can represent the body of knowledge in a given field. The ODDs could provide a huge benefit for scenario-based AS safety processes.

However, while current approaches supporting the ODD specification are only adapted for one specific domain, it may be interesting to define a commonly controlled vocabulary that may embody the knowledge related to ASs from different application domains in a harmonized way. In this paper, we then present a cross-domain approach for ODD specification. Our goal is to define a method to formalize the overall scenario-space relevant for ASs irrespective of the application domain. We collect taxonomies related to ASs from different domains: automotive, avionics, and manufacturing. We compile and structure the concepts from the taxonomies with generic concepts (e.g, time, space, weather), into an ontology from which we can extract OCs for the ODD specification of these different domains. Our multi-domain formalization aims to facilitate the definition of the scenario-space for a new application domain using the captured knowledge from existing ones. We can validate this approach by testing it on ASs from various UCs concerning different domains including the domains used to extract the generic concepts but also new domains which combine concepts from other domains, e.g, taxi-drone which combine drones and passenger vehicles [3].

II. Background

To address the problem of the harmonization of scenario representation through different domains, we use upper ontologies [4], i.e, general concepts which can be reused to express knowledge for several different domains, e.g, space, time, weather, infrastructure. The knowledge transfer from a well-defined domain to new domains is resolved by these upper ontologies that ensure completeness of semantic representation. These upper ontologies can be cross-cutting to all domains, e.g, ATIC for time representation [5], RCC-8 for space representation [6], or they can concern only a domain set, e.g. CORA [7] for the autonomous robotic domain ontology. We also integrate domain standards (e.g, the PAS1883 [8] which defines a taxonomy for safe automated driving systems, the taxonomy of unmanned aircraft, and their operations [9]). We further take into consideration the specification of additional concepts from current work that we found worth including in the ODD specification. For example, we reuse the concepts of uncertainty and exposure probability as specified by OpenODD project from ASAM [10].

III. Approach

Our approach follows several steps as presented below. The steps concerning the compilation of the concepts used

to describe OCs (SA1 and SA2) are independent of the steps concerning the formalization language used to structure these concepts (SB1 and SB2).

SA1 Theories & Standards: We list all standards and meta theories which present concepts to describe usage scenario for ASs from various domains.

SA2. Taxonomies: By using all the acquired standards, we build a taxonomy that lists all OCs extracted from theories and standards.

SB1. Ontology Language Definition: We define a domain-specific language to capture our knowledge representation as ontologies.

SB2. Ontology Formalization: We specify the supplementary rules and interfaces to adapt our ontology representation on US description.

S3. Meta-domains Knowledge: We formalize our taxonomies (SA2) with our ontology language (SB2).

S4. Generics domains Knowledge: Any new AS domains can be represented using upper ontologies, e.g, a drone ontology can be built using a weather ontology and an aerial environment ontology. The upper ontology representation is completed with domain-specific concepts, e.g, the drone maneuvers. The classic domains (i.e, automotive, aviation, robotic manufacture) are defined in a similar way to make them compatible with the ODD formalization.

S5. ODD Specification: For the ODD definition of a specific UC, we extract from the corresponding domain ontology (defined at S4) all the required concepts to characterize the scenario space of the UC. Furthermore, the ODD specification includes the OCs together with their properties and applicable range or limit values, e.g, the OC "moderate rain" may be included in the ODD with 2.5 mm/h as minimum values and 7.6 mm/h as maximum values. It is also completed by information to help the safety analysis, e.g. the exposure probability and acceptable uncertainty thresholds.

S6. ODD Usage: To refine the studied AS limits, we define restrictions as a list of rejected OC combinations. We then obtain a representation of the system scenario-space which can be used as an input for safety oriented AS development.

IV. Conclusion & Future Work

We presented an approach to formalize the ODD of autonomous systems for any domain. We detailed all the needs to achieve this formalization from the domain representation to the system-specific constraints representation. We further need to implement the tool for domain and ODD specification. For the choice of the ontology language, we have to compare the existing ones (Unified Foundational Ontologies [11], Ontology Web Language [12]) to select the most appropriate give our requirements and extend it if needed. To make our approach usable even for non-experts, a user interface could guide stakeholders through the domain description and ODD boundaries specification. The OC selection could be achieved with predefined questions on the system. Finally, our validation process includes evaluating our approach on UCs from various domains.

V. Acknowledgements

This work was partially supported by the European H2020-ECSEL CPS4EU project under grant no 692474 and the Confiance.ai project of the Grand Défi "Securing, certifying and enhancing the reliability of systems based on artificial intelligence" launched by the French Innovation Council.

References

[1] SAE Mobilus. SAE J3016 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles. Technical report, Society of Automotive Engineers International, 2018.

[2] George H Mealy. Another look at data. In *Proceedings of the November 14-16, 1967, fall joint computer conference*, pages 525–534, 1967.

[3] Muhammad Asghar Khan, Bilal Ahmed Alvi, Alamgir Safi, and Inam Ullah Khan. Drones for good in smart cities: a review. In *Proc. Int. Conf. Elect., Electron., Comput., Commun., Mech. Comput.(EECCMC)*, pages 1–6, 2018.

[4] Viviana Mascardi, Valentina Cordì, and Paolo Rosso. A comparison of upper ontologies. In *Woa*, volume 2007, pages 55–64. Citeseer, 2007.

[5] James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[6] Anthony G Cohn, Brandon Bennett, John Gooday, and Nicholas Mark Gotts. Qualitative spatial representation and reasoning with the region connection calculus. *GeoInformatica*, 1(3):275–316, 1997.

[7] Joanna Isabelle Olszewska, Marcos Barreto, Julita Bermejo-Alonso, Joel Carbonera, Abdelghani Chibani, Sandro Fiorini, Paulo Goncalves, Maki Habib, Alaa Khamis, Alberto Olivares, et al. Ontology for autonomous robotics. In *2017 26th IEEE international symposium on robot and human interactive communication (RO-MAN)*, pages 189–194. IEEE, 2017.

[8] British Standard Institution. PAS 1883 Operational Design Domain (ODD) taxonomy for an automated driving system – Specification. Standard, British Standard Institution, 2020.

[9] Anna Masutti and Filippo Tomasello. *International regulation of non-military drones*. Edward Elgar Publishing, 2018.

[10] ASAM OpenODD project details. https://www.asam.net/project-detail/asam-openodd/. Accessed: 2021-09-27.

[11] Giancarlo Guizzardi. Ontological foundations for structural conceptual models. 2005.

[12] Holger Knublauch, Daniel Oberle, Phil Tetlow, Evan Wallace, JZ Pan, and M Uschold. A semantic web primer for object-oriented software developers. *W3c working group note, W3C*, 2006.

Speech recognition under constraint

O. MATZ*, M. EL OUAZZANI, J. GANTET, N. DIARRA, M. GUILLAUMONT, B. DEGUILHEM
R&I Department – DEMS – Capgemini Engineering

ABSTRACT: Recent development in the field of speech recognition has made it possible to achieve human parity at the cost of an enormous amount of transcribed speech. In this work, we tackle the issues of data scarcity and robustness of voice recognition algorithms by developing cutting-edge approaches based on state-of-the-art methods such as self-supervised learning and data-centric AI.

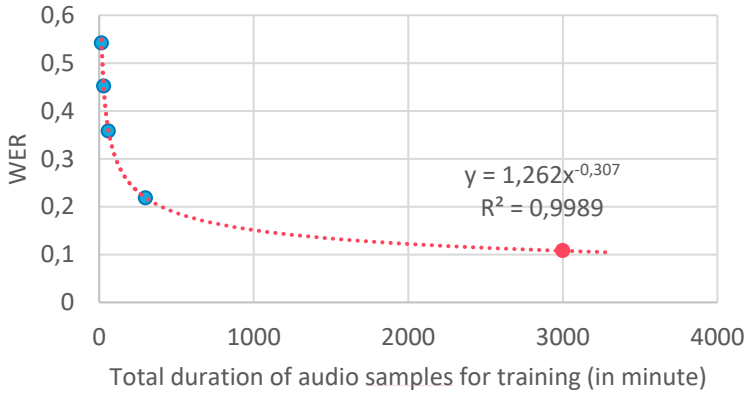
Today, speech recognition technologies are mature enough to be integrated into marketable solutions. However, these are owned by large groups with access to significant resources, both from a computational and data point of view. In particular, the development of a voice recognition algorithm requires several tens of thousands of hours of transcribed recording to achieve the performance of solutions currently on the market. In addition, if voice recognition solutions are widely used in our daily lives, they require significant computing resources and are not deployed on device but via cloud computing services. In most cases, speech recognition offers do not meet the needs of industry because (i) they do not guarantee data confidentiality i.e., manufacturers refuse to send their private and sensitive data to the cloud, (ii) the cost of use is important, (iii) they are not adapted to the recording condition in industry and (iv) they are not adapted to the specific business vocabulary of the manufacturer. In addition, the state-of-the-art algorithms proposed in the literature are evaluated on datasets that are not representative of the real world (audiobook, recording in a silent environment, professional audio recording equipment, …). For example, our preliminary work on the subject showed that the precision of these state-of-the-art algorithms was not acceptable on real world data nor in industrial conditions: we observe a word error rate greater than 40% and up to 90%. In this context, the development of voice recognition solutions for French language deployed locally, requiring little data labeled for training and robust to noisy environments represents a real challenge in the field of industry.

To solve these challenges, our work focus on the development of voice recognition algorithm focuses on 3 axes:

- Develop a Speech to Text approach adapted to a small labeled dataset by focusing work mainly on recent approaches of self-supervised learning
- Robust the Speech to Text algorithms under real conditions (noise, reverberation, disturbance, …)
- Reduce the complexity of algorithm for on-device deployment to overcome constraints related to cloud computing (cost, GDPR and data privacy compliant, …)

To tackle the problem of data scarcity, we have developed a self-supervised algorithm based on contrastive learning and the wav2vec2 architecture. More precisely, our approach is made up of two steps: first we use a pretraining block which is based on contrastive predictive coding to learn a high-level representation of the data in an unsupervised way, then in a second phase we add a connectionist temporal overlay to the pretraining block to adapt the model to the transcription task. For the pretraining phase, we have collected more than 10 000 hours audio recording in real life condition and use them for training phase in order to learn a high-level representation of an audio sample. Then, the model is finetuned on few hours of labelled speech. We do some experiments by varying the quantities of labelled audio use in the finetuning step and show that our algorithm allows to reach 20% raw WER, i.e., without any language model post-processing, and thus with only 5 hours of labelled speech while

traditional speech recognition algorithm requires thousands of hours to reach similar accuracy. Although our work deserves further study, our first results are very encouraging and highlight how self-supervised approaches allow to reach similar accuracy as solutions from big cloud provider with only a few ten of hours of labelled audio. Thus, in line with the work of Baevski et al. our work breaks the barrier of labelled audio data scarcity and paves the way for a new paradigm in speech recognition.



Our work also addresses the problem of the robustness of the model. In particular, most of the state-of-the-art Speech to Text model are build and benchmarked on datasets that are not representative of real conditions such audio books or recording in clean conditions. Indeed, conditions in industry are very different from these idealized datasets with the presence of noise, reverberation, or with different sample rate which causes a significant decrease in the performance. To face these issues, we have put a lot of effort to build real world dataset of more of 300 hours labelled with speech from different source (interview, documentary, conference, debate, …), sound environment (indoor, outdoor, open space, factory, station, …) and from various speaker. Moreover, we have also developed a denoising algorithm to deals with very noisy conditions. In our work, we have developed an autoencoder to denoise the raw audio data in real time based on the recent development of audio source separation. Our work highlights the efficiency of deep learning-based approaches for end-to-end audio denoising, and our results outperform state-of-the-art algorithm for French noisy speech with $0 \leq SNR \leq 15$. However, for very noisy speech, i.e., with a SNR < 0 dB, the noise removal is effective but goes along with a very low intelligibility making transcription very hard, even for humans.

This work was granted access to the HPC resources of IDRIS under the allocation 2021-AD011012570 made by GENCI.

References:

[1] D. Amodei et al., *'Deep Speech 2: End-to-End Speech Recognition in English and Mandarin'*, 2015.
[2] Q. Xu et al., *'Self-Training and Pre-Training Are Complementary for Speech Recognition'*, 2020.
[3] A. Baevski et al. *'wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations'*, 2020.
[4] A. Défossez et al. *'Demucs: Deep Extractor for Music Sources with Extra Unlabeled Data Remixed'*, 2019.
[5] A. Défossez, et al. *'Real Time Speech Enhancement in the Waveform Domain'*, 2020.

Corresponding author: olivier.matz@capgemini.com

Poster submission for EC4.6 Randomized smoothing for time-series. Application to the Air Liquide demand forecasting dataset

Aurélien Benoit-Lévy
Institut LIST
CEA, Université Paris-Saclay
aurelien.benoit-levy@cea.fr

Abstract

In this poster, we investigate the robustness of neural network models to adversarial attacks in the context of time series forecasting. We present percentile smoothing [1], a technique similar to randomized smoothing [2, 3], which is more suited for regression tasks (as opposed to classification). We use the Air Liquid demand forecast dataset, and provide robustness intervals for the prediction of a simple neural network to ℓ_2-bounded attacks.

1. Introduction

Randomized smoothing, a technique used to provide robustness to adversarial attacks, is mainly studied in the literature for classification tasks. For classification, one would expect a model to be stable to small perturbations around a test image. However for regression, the situation is different, as by construction, one would expect the output of a model to vary if the inputs are perturbed. It appeared that certification to adversariel attacks for regression models is largely under-studied in the literature. One of the only paper tackling this question [1] introduces *percentile smoothing* to bring some level of certification to detection algorithms such as YOLO, Faster-RCNN, ... In the detection problem, one is task to predict the position of the corners of the boxes of the object to be detected, and this can be seen as a regression problem.

We therefore adapt the formalism developed in [1] to the case of time series forecast and provide robustness interval for ℓ_2-bounded adversarial attacks. In the following, we first describe the Air Liquide dataset and its preprocessing. Then we present the formalism of percentile smoothing. Finally we show some preliminary results on the Air Liquide data.

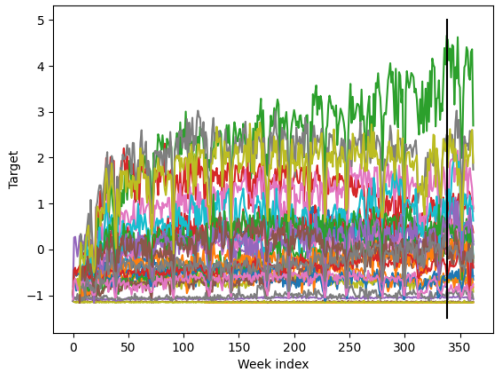


Figure 1. Target variable for the 29 times series of the Air Liquide dataset. The vertical black line indicates the separation between the train and the test sets.

2. Presentation of the Air Liquide dataset

The Air Liquide demand forecast dataset consists in tabular data with continuous and categorical variables. We studied this dataset and realised that it is actually the concatenation of 29 times series that are largely independent. Since the data is anonymised, we can only assume that each time series correspond to a given client or production line[1]. There are four variables that are real-values, one of which, `Numerical_0`, constitues the target to predict. Figure 1 shows the target variable for the 29 times series.

3. Percentile smoothing

Following [1], we introduce the percentile smoothing version of a function $f : \mathbb{R}^d \to \mathbb{R}$

$$\underline{h}_p = \sup \{y \in \mathbb{R} | \mathbb{P}[f(x+G) \leq y] \leq p\}, \quad (1)$$

$$\overline{h}_p = \inf \{y \in \mathbb{R} | \mathbb{P}[f(x+G) \leq y] \geq p\}, \quad (2)$$

[1] This is just for context, the real meaning of the dataset does not matter in this use case.

where $G \sim \mathcal{N}(0, \sigma^2 I)$.Those two quantities coincide when f is continuous (which is our case here), but they might differ if f is not real-valued. If $p = 0.5$, those percentiles reduce to the median of $f(x + G)$.

The main result of [1] is the following lemma:

Lemma 1. *A percentile-smoothed function h_p with adversarial perturbation δ can be bounded as*

$$\underline{h}_{\underline{p}}(x) \leq h_p(x + \delta) \leq \overline{h}_{\overline{p}}(x), \forall ||\delta||_2 < \epsilon, \qquad (3)$$

where $\underline{p} := \Phi\left(\Phi^{-1}(p) - \frac{\epsilon}{\sigma}\right)$ and $\overline{p} := \Phi\left(\Phi^{-1}(p) + \frac{\epsilon}{\sigma}\right)$, with Φ being the standard Gaussian CDF. In practice $p = 0.5$ as we consider the median. As can be seen, the interval between the bounds is controlled by the values of $\frac{\epsilon}{\sigma}$. While ϵ is the size of the attack against which we want certification, there is some freedom in the choice of the amount of noise (characterised by σ) we can used to perform the smoothing. If one uses too less noise ($\sigma << \epsilon$), the noised inferences would be clustered around a central value, but the percentiles to interrogate would be very large, and it would require a large number of random realisations of the data to get accurate estimated of $\underline{h}_{\underline{p}}(x)$ and $\overline{h}_{\overline{p}}(x)$. On the contrary, if the noise is too large, the smoothed function will lose its predictive power and the resulting results will present poor performances.

3.1. Estimation of h_p

Given a function f, the percentile smoothing of f is obtained the following way:

- draw n samples $G \sim \mathcal{N}(0, \sigma^2 I)$,
- run the regressor f on $x + G$,
- return the 50^{th} percentile of the resulting array.

Similarly, we get a numerical approximation of the bounds l, u, such that

$$P(\overline{h}_{\overline{p}} \leq u) \geq \alpha \,\text{and}\, P(\underline{h}_{\underline{p}} \geq l) \geq \alpha, \qquad (4)$$

for some confidence threshold α (typically $\alpha = 0.9999$).

4. Preliminary results

For these preliminary results, we focused on the implementation and the demonstration of the percentile smoothing method rather than the sheer performances of the prediction model. We thus considered a simple multi-layer perceptron (MLP), and we keep the same pre-processing of the data as the Air Liquide XGBoost code. Since the time series are independent, we trained a model for each time series. Results are summarised in Fig. 2, where we show the test part of the target (in black). The prediction of the MLP is shown in orange. The smoothed prediction (corresponding

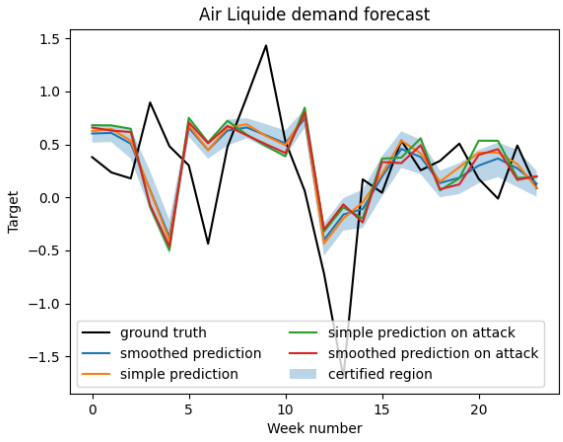


Figure 2. Illustration of the percentile smoothing technique on one Air Liquide time series.

to $h_p(x)$ in the formalism developed in Sect. 3). The shaded region is delimiting the area between $\underline{h}_{\underline{p}}(x)$ and $\overline{h}_{\overline{p}}(x)$. It corresponds to the robustness interval of the model when attacked by perturbations δ, such as $||\delta||_2 < \epsilon$. An illustration of such an attack is given by the green line, which is the prediction of a Fast Gradient Sign Method perturbation. As can be seem, around week 20, the prediction of the MLP model exceeds the upper bound of the robustness interval. However, the smoothed prediction on this attack (red line) is well within the theoretical bounds. It is important to note that this robustness interval is not a confidence interval in the sense where the true answer would lie between the bounds of this interval. Rather it gives the interval in which the predictions can vary when the inputs are modified by a ℓ_2 bounded perturbation.

5. Conclusion and perspectives

We have implemented the percentile (or median) smoothing technique to the Air Liquide demand forecasting dataset to provide robustness. Several extensions of this work can be envisaged, such as certification against different types of perturbations.

References

[1] Ping-yeh Chiang, Michael J. Curry, Ahmed Abdelkader, Aounon Kumar, John Dickerson, and Tom Goldstein. Detection as Regression: Certified Object Detection by Median Smoothing. *arXiv:2007.03730*, July 2020. 1, 2

[2] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. abs/1902.02918, 2019. 1

[3] Greg Yang, Tony Duan, J. Edward Hu, Hadi Salman, Ilya P. Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. abs/2002.08118, 2020. 1

A formalism for embedded machine learning applications

Hugo Pompougnac[1], Dumitru Potop-Butucaru[1] and Albert Cohen[2]

[1]INRIA, team Kairos
[2]Google France

September 2021

Abstract

In the quest for (critical) embedded systems featuring machine learning (ML) components, we propose a method and a language allowing the conjugate the expressive power of Lustre for embedded control aspects and TensorFlow/MLIR for ML/HPC aspects. We explore the theoretical and engineering aspects of this language.

The Static Single Assignment (SSA) form has proven an extremely useful tool in the hands of compiler builders. First introduced as a representation to facilitate optimizations, it became a staple of optimizing compilers. More recently, its semantic properties—e.g. functional *determinism* while still allowing for limited *concurrency*—established it as a sound basis for High-Performance-Computing (HPC) compilation frameworks such as MLIR, where different abstraction levels of the same application[1] share the structural and semantic principles of SSA, allowing them to co-exist while being subject to common analysis and optimization passes (in addition to specialized ones).

But while compilation frameworks such as MLIR concentrate the existing know-how in HPC compilation for virtually every execution platform, they lack a key ingredient needed in the *high-performance embedded systems* of the future—the ability to represent reactive control and real-time aspects of a system. They do not provide first-class representation and reasoning for systems with a cyclic execution model, synchronization with external time references (logical or physical), synchronization with other systems, tasks and I/O with multiple periods and execution modes.

And yet, while the standard SSA form does not cover these aspects, it shares strong structural and semantic ties with one of the main programming models for reactive real-time systems: *dataflow synchrony*, and its large and structured corpus of theory and practice of reactive systems design.

[1]Ranging from ML dataflow graphs and linear algebra specifications down to affine loop nests and optimized (tiled, vectorized...) low-level code.

Contribution. Relying on this syntactic and semantic proximity, we extend the SSA-based MLIR framework to open it to synchronous reactive programming of real-time applications. We illustrate the expressiveness of our extension through the compilation of the pure dataflow core of the Lustre language. This allows us to model and compile all data processing, computational and reactive control aspects of signal processing and machine learning applications. In the compilation of Lustre, following an initial normalization phase, all data type verifications, buffer synthesis, and causality analysis can be handled using existing MLIR SSA algorithms. Only the initialization analysis specific to the synchronous model (a.k.a. clock calculus or analysis) requires specific handling, leading to significant code reuse.

The MLIR embedding of Lustre is non-trivial. As modularity based on function calls is no longer natural due to the cyclic execution model, we introduce a *node instantiation* mechanism. We also generalize the usage of the special *undefined/absent* value in SSA semantics and in low-level intermediate representations such as LLVM IR. We clarify its semantics and relate it to the notion of absence and the associated static analyses (*clock calculi*) of synchronous languages.

Our extension remains fully compatible with SSA analysis and code transformation algorithms. It allows giving semantics and an implementation to all correct SSA specifications. It also supports static analyses determining correctness from a synchronous semantics point of view.

The resulting language allows the modeling of signal processing and deep neural network inference in the (closed) loop of feedback-directed control systems. With only a minor time investment in using MLIR's optimization support, generated code surpasses in speed state-of-the-art synchronous language compilers.

Trust in AI: increasing acceptance of robotics and artificial intelligence agents through impression design

Author: Cyrielle Chappuis, PhD.[1]

[1]Consultant Engineer at Capgemini Engineering (Data Driven Solutions), *cyrielle.chappuis@capgemini.com*

Abstract:

Social robots that assist, engage and interact with people are seen by many as the future of human-centered artificial intelligence. One of the big challenges of this evolution is to facilitate acceptance of robots in human habitats. Managing the impressions of social robots is a first step towards ensuring trust, acceptance and integration of AI.

Introduction

As robots and intelligent agents are expected to become more prevalent in everyday contexts, public reception of Robotics and Artificial Intelligence (RAI) is the cornerstone of the acceptance, uptake and research funding of such technology. However, algorithms are becoming more complex and opaque by the day, inducing a negative impact on the trustworthiness of a system (Linegang et al., 2006; Stubbs et al., 2007). The research field of Explainable Artificial Intelligence tackles this shortcoming by developing innovative solutions aiming at increasing user understanding, and thereby trust, of artificial intelligence systems. An effective way of doing so is via the design of embodied virtual agents communicating about the systems' results and processes in a natural and human-like way (Weitz et al., 2019). However this solution comes at a risk of sliding on the disturbing side of the uncanny valley by being too close to human-likeness yet too far from an industrial robot. The Uncanny valley theory expects that such a design could trigger a negative emotional response towards the virtual agent, thus impairing rapport bonding and trust. For instance, people have reported feeling "unsafe" or "uncomfortable" when interacting with a robot. So how can virtual agents or robot design be controlled as to manage impressions and trigger positive emotions?

It is hypothesized that people perceive autonomous intelligent systems by applying human traits to them (Graaf & Malle, 2017). Research shows that humans tend to behave towards robots in a similar way they would with another human: for example as evidenced by a robot "black sheep effect", they distinguish ingroup from outgroup members depending on the characteristics of the robot (Steain et al., 2019). Some researchers have even observed unexpected bullying and abusive behaviors towards robots (Brščić et al., 2015), suggesting that HRI elicit strong affective responses that certainly were not controlled nor wanted by the engineers. Conversely, there is evidence of human prosocial behavior (Connolly et al., 2020) and even empathy (Slater et al., 2006) within HRI when the robot expressed emotional reactions.

Impressions are fast and structured affective responses: they occur within 100-ms of meeting a person for the first time (Willis & Todorov, 2006). More importantly, they are central to action tendencies (approach or avoidance) and decision making. In fact, many of our high-impact decisions are based on zero-acquaintance judgments such as criminal sentencing (Dumas & Testé, 2006), political voting (Chen et al., 2014), salary (Fruhen et al., 2015), economic choices (Rezlescu et al., 2012).

What impressions do we form?

When a new person is met for the first time, people judge them on two main dimensions: trustworthiness-warmth ("is this person trustworthy? Are their intentions good?") and dominance-competence ("is this person capable of acting on their intentions ?"). Evidence shows that perceived competence and intelligence in a robot influences trust behaviors towards it (Haring et al., 2013). Such judgments occur on extremely fragmentary sources of information such as facial traits, vocal cues, clothing, and even body shape and size.

Trustworthy faces are characterized by higher eyebrows and a larger space between eyes and eyebrows, pronounced cheekbones, wide chins and thin nose sellion. Untrustworthy face configuration is associated with an increased response of the (right) amygdala, a neural correlate of relevance and threat detection (Todorov et al., 2008).

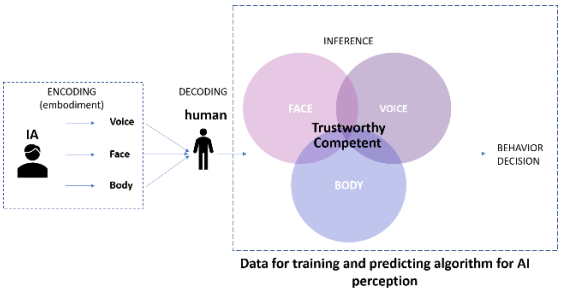


Figure 1. A sequential model of impression formation: data for training of Capgemini Engineering's testing solution is based on natural reactions.

Methodology: a testing solution for AI trustworthiness

"Emotions are largely seen as interfaces between an organism and its environment […] they constitute a sort of detection mechanism for personal relevance of events and stimuli in the organism's surroundings. They have been found to interact with several, if not all, cognitive mechanisms, such as attention, memory, judgment, etc." (Scherer & Moors, 2019)

Our project aims at casting light on the features of embodied AI that elicit emotional responses and influence behavior in human-machine interaction. To this end, we are developing a protocol for evaluation of affective perception, useful for the detection of trust and behavioral prediction.

Capgemini Engineering's machine learning algorithm trains on multi-channel measures of facial expression, vocal expression, and physiological activation. It is able to analyze a combination of responses on several sub-systems of the organism implicated in the emotion event and predicting behaviors towards an item.

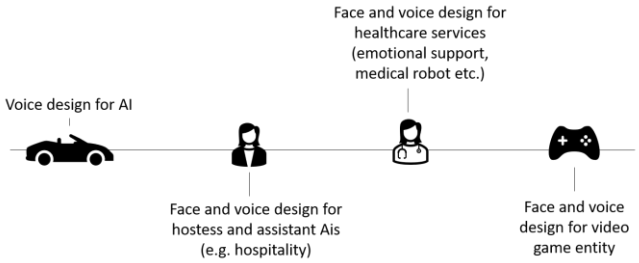


Figure 2. Several use cases for development of trustworthy and competent embodied AI.

Conclusion

Human proclivity to form impressions of their environment can be leveraged to design more effective and more trustworthy AIs.

References

Brščić, D., Kidokoro, H., Suehiro, Y., & Kanda, T. (2015). Escaping from Children's Abuse of Social Robots. *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, 59-66. https://doi.org/10.1145/2696454.2696468

Chen, F. F., Jing, Y., & Lee, J. M. (2014). The looks of a leader: Competent and trustworthy, but not dominant. *Journal of Experimental Social Psychology*, *51*, 27-33. https://doi.org/10.1016/j.jesp.2013.10.008

Connolly, J., Mocz, V., Salomons, N., Valdez, J., Tsoi, N., Scassellati, B., & Vázquez, M. (2020). Prompting Prosocial Human Interventions in Response to Robot Mistreatment. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction* (p. 211-220). Association for Computing Machinery. https://doi.org/10.1145/3319502.3374781

Dumas, R., & Testé, B. (2006). The Influence of Criminal Facial Stereotypes on Juridic Judgments. *Swiss Journal of Psychology*, *65*(4), 237-244. https://doi.org/10.1024/1421-0185.65.4.237

Fruhen, L. S., Watkins, C. D., & Jones, B. C. (2015). Perceptions of facial dominance, trustworthiness and attractiveness predict managerial pay awards in experimental tasks. *The Leadership Quarterly*, *26*(6), 1005-1016. https://doi.org/10.1016/j.leaqua.2015.07.001

Graaf, M. M. A. de, & Malle, B. F. (2017, octobre 9). How People Explain Action (and Autonomous Intelligent Systems Should Too). *2017 AAAI Fall Symposium Series*. 2017 AAAI Fall Symposium Series. https://www.aaai.org/ocs/index.php/FSS/FSS17/paper/view/16009

Haring, K. S., Matsumoto, Yoshio, & Watanabe, Katsumi. (2013). *How do people perceive and trust a lifelike robot*. *1*, 23-25.

Linegang, M. P., Stoner, H. A., Patterson, M. J., Seppelt, B. D., Hoffman, J. D., Crittendon, Z. B., & Lee, J. D. (2006). Human-Automation Collaboration in Dynamic Mission Planning : A Challenge Requiring an Ecological Approach. *Th ANNUAL MEETING*, 5.

Rezlescu, C., Duchaine, B., Olivola, C. Y., & Chater, N. (2012). Unfakeable Facial Configurations Affect Strategic Choices in Trust Games with or without Information about Past Behavior. *PLoS ONE*, *7*(3), e34293. https://doi.org/10.1371/journal.pone.0034293

Slater, M., Antley, A., Davison, A., Swapp, D., Guger, C., Barker, C., Pistrang, N., & Sanchez-Vives, M. V. (2006). A Virtual Reprise of the Stanley Milgram Obedience Experiments. *PLOS ONE*, *1*(1), e39. https://doi.org/10.1371/journal.pone.0000039

Steain, A., Stanton, C. J., & Stevens, C. J. (2019). The black sheep effect : The case of the deviant ingroup robot. *PLOS ONE*, *14*(10), e0222975. https://doi.org/10.1371/journal.pone.0222975

Stubbs, K., Hinds, P., & Wettergreen, D. (2007). Autonomy and Common Ground in Human-Robot Interaction : A Field Study. *Intelligent Systems, IEEE*, *22*, 42-50. https://doi.org/10.1109/MIS.2007.21

Todorov, A., Baron, S. G., & Oosterhof, N. N. (2008). Evaluating face trustworthiness : A model based approach. *Social Cognitive and Affective Neuroscience*, *3*(2), 119-127. https://doi.org/10.1093/scan/nsn009

Weitz, K., Schiller, D., Schlagowski, R., Huber, T., & André, E. (2019). « Do you trust me? » : Increasing User-Trust by Integrating Virtual Agents in Explainable AI Interaction Design. *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, 7-9. https://doi.org/10.1145/3308532.3329441

Willis, J., & Todorov, A. (2006). First Impressions : Making Up Your Mind After a 100-Ms Exposure to a Face.

Psychological Science, 17(7), 592-598. https://doi.org/10.1111/j.1467-9280.2006.01750.x