Lessons learned from TAILOR benchmarks/challenges

Sebastien Treguer, Marc Schoenauer

Foundations of Trustworthy AI – Integrating Reasoning, Learning and Optimization
TAILOR
Grant Agreement Number 952215

Lessons learned from TAILOR benchmarks and challenges

Document type (nature)	Report
Deliverable No	2.4
Work package number(s)	2
Date	Due 30 June 2024
Responsible Beneficiary	INRIA, ID 3
Author(s)	Sébastien Treguer and Marc Schoenauer
Publicity level	Public
Short description	This report synthesises the outcomes of all challenges and benchmarks organised with the participation of TAILOR partners, from the point of view of Trustworthy AI (TAI) on the one hand, and Learning, Optimization and Reasoning (LOR) on the other hand.

History			
Revision	**Date**	**Modification**	**Author**
First version	17/11/24	-	Sébastien Treguer and Marc Schoenauer

Document Review		
Reviewer	**Partner ID / Acronym**	**Date of report approval**
Umberto Straccia	#2 / CNR	15/11/2024
Philipp Slussallek	#26 / DFKI	14/11/2024

Table of Contents

Executive Summary

This report synthesises the outcomes of all challenges and benchmarks organised with the participation of TAILOR partners, from the point of view of Trustworthy AI (TAI) on the one hand, and Learning, Optimization and Reasoning (LOR) on the other hand.

All concerned challenges have been described in detail in Deliverables 2.3 and 2.6. They are hence only very briefly presented here. But the specific features of the best performing solutions (most of the ones that received a prize) are analysed from both the LOR and the TAI points of view, and partial lessons are derived. The report ends with more global lessons that can be possibly generalised from these challenges, and a conclusion proposing a few further directions for future challenges, in the line of hybridization between Learning, Optimization and Reasoning, or/and favouring trustworthy AI.

Introduction: A Brief history of TAILOR Data Challenges

For the sake of completeness, we start by presenting again the arguments that led us to propose to run AI Challenges within TAILOR, and were first given in the Introduction of Deliverable 2.3, submitted in July 2022..

Challenges have been a strong drive in Artificial Intelligence for more than 30 years now, from the very first SAT competitions in 1992 (still on-going) to the series of Visual Recognition Challenges in the early 2010's that definitely demonstrated the incredible effectiveness of Deep Learning approaches. The introduction of Deliverable 2.2 of this project gives a more detailed historical survey of challenges in AI, that will not be repeated here.

In the absence of strong theoretical results in most AI fields, challenges and open benchmarks are the only way to test and compare algorithms on different types of situations in a fair and reproducible way. The success of the historical pioneer Kaggle challenge platform, and its 800000+ AI experts users, led Google to buy it in 2017, in order to "continue democratising AI", as advocated by Fei-Fei Li in the official announcement. Whatever the actual motivations of Google for such a move, this shows, if at all needed, the importance of challenges in the AI world. However, many AI practitioners, in particular in Europe, have turned to other platforms to organise their challenges, to avoid disclosing their data (and expertise) to this US BigTech company. This boosted other more open and transparent Open Source platforms such as AIcrowd or the university-operated Codalab, that was chosen in the TAILOR proposal to run TAILOR challenges not only because it is a reliable and completely transparent tool, but also because its scientific coordinator is Isabelle Guyon, a pioneer in challenge design and setup, through the Chalearn organisation, and a member of the TAILOR INRIA team (partner #3).

Organising a challenge requires quite some work, and here we refer again to Deliverable 2.2 of this project, where the whole process is detailed and recommendations are given, with specifics related to Codalab. Furthermore, the challenges organized within TAILOR should address TAILOR-related topics, something that is completely problem-dependent and could not be described at the general level in the Deliverable.

The chronological history of TAILOR challenges is the following. The initial plan for TAILOR was to organise one academic and one industrial challenge per year (during the three years initially planned for the project). The academic challenges would be gathered from the 45 TAILOR academic partners, while the industrial challenges would preferably be proposed by the 10 TAILOR industrial partners, plus the analysis of the results of the Theme Development Workshops organised in the context of WP8.

We hence issued a call for challenge topics/data during the Kick-Off meeting (Sept. 29. 2020), for both types of competition, as well as during all meetings of WP8, for industrial competitions. Things started well: we rapidly received two propositions from TAILOR partners: an industrial competition from EDF (together with a consortium of large French industries), regarding Smarter Mobility (optimisation of charging stations for Electric Vehicles) and an academic competition from Fraunhofer (Prediction of Inductive Links). Unfortunately, for many reasons, including of course the Covid pandemic and the absence of physical meetings, but also the inertia of the industrial consortium around EDF, things progressed very slowly, and these challenges are still in the pipeline, hopefully to be launched next Fall for the latter. Also, the Theme Development Workshops only started in

Fall 2021, i.e. AI in the Public Sector (Sept. 7 and 9 2021), Future Mobility – Value of Data & Trust in AI (Oct 28 2021), and AI for Future Healthcare (Dec. 16 2021), but no concrete challenge spontaneously emerged from them. Two other were held in Spring 2022, i.e. AI: Mitigating Bias & Disinformation (May 18 2022), and AI for Future Manufacturing (May 10. 2022), for which the reports are still to come.

It became obvious that we would not be able to organise the promised number of challenges on our own, limited to inputs from TAILOR partners. Therefore, we identified existing challenge series, linked to TAILOR topics, that we could contribute to. We started with the activities of INRIA's TAU group on the Codalab platform, led by Isabelle Guyon, and TAILOR officially joined the organisation and the lists of sponsors of the Meta-Learning challenges[1], and the Learning to Run a Power Network challenge (L2RPN). TAILOR contribution consists of human power (for all projects, Sébastien Treguer, hired part time on TAILOR budget, Marc Schoenauer, and of course Isabelle Guyon, plus interns and PhD students), advertisement over TAILOR network and affiliates, and financial contributions: to Codalab storage, with cash prizes for the winners of the Meta-Learning challenges.

The above introduction was written in July, 2002. It is still valid, but since then, things have progressed fast, as reported in Deliverable 2.6, resulting in nine Data Challenges[2] having been run with TAILOR contributing to the organisation one way or another. All are described in detail in Deliverable 2.6, sometimes pointing to the detailed discussion of Deliverable 2.3 for the Data Challenges that were already presented there.

Nevertheless, we should keep in mind that most of these Data Challenges were designed without specific scientific concern related to TAILOR, and generally led by non-TAILOR scientists, as TAILOR partners failed to answer our call for possible Challenge Data. One noticeable exception was the Inductive Link Prediction challenge. Unfortunately, because of staff change at Fraunhofer, this challenge was abandoned as a TAILOR challenge, and will not be discussed here. All details regarding this side of the TAILOR Challenges story have been given in Deliverable 2.3.

The goal of this Deliverable, as described in the initial Description of Work, is to study the links between the Data Challenges that were run within the project, and TAILOR scientific research axes, i.e., from both the Learning, Optimization and Reasoning (LOR) aspect of the winning approaches, and their Trustworthiness as AI algorithms (TAI). However, as discussed above, none of these Data Challenges were designed with LOR or TAI as a target. Hence this Deliverable should be understood as **a post-hoc discussion** based on the results of these Challenges. Indeed, the TAILOR vision is that AI should not restrict to Machine Learning (the « L » of LOR), and that further progress in AI should and will involve hybridizations between Machine Learning and Reasoning, involving Optimization at all stages. Furthermore, when it comes to AI research, at least in Europe, emphasis is now clearly on Trustworthiness of AI algorithm, in order to increase their efficiency and accuracy

[1] beyond INRIA, TUE (Technical University Eindhoven, TAILOR partner #12), and University Leiden, (TAILOR partner #7) were already participating to the organisation

[2] To avoid confusion with « intellectual challenges », and as suggested in the review we received to Deliverable 2.2, and immediately implemented in Deliverable 2.6, we will here also, and from now on, use the terms « Data Challenges », even when Data is not central in the challenge, to avoid confusion with the scientific hurdles that the word "challenge" generally means, too – except where there is no possible ambiguity, e.g., when written together with the name of the challenge.

(see e.g., the unavoidable hallucinations of LLMs at this point in time in the history of AI). But mainly for ethical and societal reasons, to highlight the fact that the benefits of AI surpass the drawbacks and fight the fears AI creates, often for good reasons in the absence of safety and trustworthy bounds.

This Deliverable will now survey in turn the results of the nine TAILOR challenges that were run during the 4 years of the project, adopting successively (or together) both LOR and TAI points of view, and discussing “local” lessons that were learnt for each of these challenges. EAch challenge will be very briefly described, and the links to the challenge related papers or repository, already presented in Deliverable 2.6, will be recalled for the sake of completeness, eventually completed by more recently published links. The final section will discuss some general lessons that can be drawn from these real-size experiments, and whether TAILOR vision starts to become reality.

Smarter Mobility Data Challenge

The goal of this industrial Data Challenge was to predict the status (available, charging, waiting with charged vehicle attached, other[3]) of charging stations for electric vehicles in the Paris area, at station level, area level (four areas covering Paris) and global level. The data was historical data of all Paris stations during one year (see Deliverables 2.3 and 2.6).

The teams were ranked according to performances at the three levels. The six best teams had to write a small report, and do an oral presentation in front of a jury chaired by Cédric Villani. The final ranking was decided by the jury based on the performance rankings and the quality of the presentations, and three teams were awarded the three prizes – team1, team2 and team3 for simplicity here - the only teams which outperformed the baseline (see below).

Results

The data

The data here is real-world data: it is noisy, with quite a few missing data in the time series. Furthermore, the size of the data is rather small, at least far from deserving the name of “Big Data”: as a matter of fact, some of the competing teams even complained that the provided data was too small to run efficient algorithms. As a consequence, the competitors (at least the 3 winning teams, the only ones who wrote a report detailing their approach) all started by studying the available data closely.

Team1 used a validation set to choose the approach to handle missing data, considering mean by station, forward and backward filling, simple moving average, weighted moving average, and exponential moving weighted average – choosing the latter as best performing on the validation set.

Team2 discovered a large distribution shift corresponding to the start of Covid19 regulations. They also found out that the data was noisy (failure of stationarity test), and used some sliding window averaging to cope with the noise. They also investigated several ways to handle missing data, and finally decided to simply drop the corresponding time slots. They also performed some original “data augmentation”, adding columns (features) rather than lines: More precisely, they added a Boolean value indicating that the day was French holiday, and sine and cosine transforms of the day and month in the year.

Team3 also noticed the change of distribution in October, though came up with different explanations. They complained that the data did not even cover a full year. As a consequence, they decided to give more weight to the most recent data in the time series.

A detailed knowledge of the data (e.g., knowing about Covid, or the French holidays) was necessary here to handle it properly, to clean it, and shows the importance of understanding the data before starting any learning.

[3] the plug can be out-of-order, disconnected, occupied by the wrong vehicle, etc

The ML algorithms

Regarding the Machine Learning model, the organisers had provided a baseline that was trained using CatBoost, a tree-based Gradient Boosting algorithm specialised in regression for categorical data. As a consequence, most competitors (and at least the top 3 winners) also used CatBoost. But only three of them (as mentioned above) succeeded in outperforming the organisers' results. Several teams also turned, at the station level, the prediction problem of the 4 states of the three plugs into a classification problem with 20 classes, the possible values of the 4 states (number of plugs in that state) that must sum up to 3 (there are 3 plugs per station).

Team1 started with a model selection (as they did for choosing the data handling algorithm), and decided CatBoost obtained the best performance on the validation set among SARIMAX, LSTM, XGBoost, and random forest (also because it handles categorical data seamlessly). At the station level, they solved the classification problem above (using CatBoost). At the area and global levels, they used some sequentially chained regression, predicting the availability state first, then using this predicted value to predict the charging state, then the waiting state, and finally the last state.

Team2 trained one tree-based regression model using an autoregressive XGBoost with 100 estimators (scikit-learn implementation) for each station and each possible state. They also trained a classification XGBoost with 300 estimators to choose among the 20 classes described above. They also trained a standard ARIMA model that performed well, though it output almost-constant values. The very different characteristics of these models led them to finally propose a linear aggregation of these three models with weights proportional to their performances on the public data set.

Team3 experimented using a sound Training/Validation procedure with twelve different settings regarding the depth of the trees and the number of gradient iterations for the CatBoost algorithm as provided by the organisers. Ultimately, they chose the model with exponential decay of past data importance, depth 5 and 200 iterations, and qualified this model trained on the whole training set.

A post-challenge study was made by **the organisers**, and used the three winning solutions together. It is well known that Aggregating uncorrelated estimators (here, the results of the top 3 teams), even naively (uniform averaging) can improve over each of the base estimators alone. This was the case here, and even better results were obtained when the weights of the aggregation were **optimised** by some gradient procedure over the whole training set: The resulting model outperformed all other models in all criteria (station, area, global) and all state prediction (available, charging, waiting, other).

LOR

All three winning teams spent a lot of effort on data pre-processing, coming up with smart ways of handling missing data, and denoising this real-world data using some average over a sliding window. Such an approach can be seen as some kind of expert reasoning on the available data - though it was not done automatically, and only as a pre-training phase.

All teams used some Machine Learning algorithm at the heart of their approaches, as the main goal was prediction from labelled data, i.e., supervised learning. However, because of the small size of the data, they used “classical” machine learning approaches (gradient boosting, decision trees, ARIMA),through their robust implementations from scikit-learn or classical statistical packages.
However, they also used optimization algorithms when doing sound model selection and hyperparameter tuning, using standard AutoAI optimization approaches on a validation set selected from the training set.

In summary, an interesting lesson here: applying CatBoost blind gave a pretty good baseline for this supervised learning problem, and only the addition of knowledge in data processing, and the optimal choice of models and hyperparameters in a sound way led to some improvement.

TAI

The final winners have been determined by a jury after providing both a written and an oral presentations of their project, not only based on forecasting accuracy, but also on:

- clarity of written and oral presentations
- usefulness of the documentation
- rigour of the proposed approach
- interpretability of the models
- practicality to the industry.

Note that only the third-ranked team recognized in their report that tree-based models as the ones resulting from using CatBoost are more **interpretable** than the other models (ARIMA, CatBoost, etc). However, efficiency to handle categorical data was the main drive for choosing CatBoost, not the explainability of the resulting models.

Links

- Data, code and tutorials on Gitlab.
- The paper describing the challenge and the results (published in DMLR - Data-centric Machine Learning Research journal in Sept. 2024)
- The codalab page of the challenge

Learning to Run a Power Network (L2RPN)

This challenge, co-organized by Inria TAU (TAILOR partner #3) and RTE France (the operator of the French national Power Grid, which is Europe largest grid operator), was the last one of a series of challenges aiming to control a Power Grid under more and more complex contexts. This edition was subtitled “Energies of the future and carbon neutrality”. Its goal was to control the Power Grid (meet the demand and make sure it stays within security bounds) while taking into account renewable energies, which are by nature intermittent. Different scenarios were proposed, and the behaviour of the Grid, given the inputs and operational commands, was simulated using the RTE Python module Grid2Op.

The setting of the L2RPN challenges is that of Reinforcement Learning: at each time step, the agent receives information about the state of the Grid, and must make the corresponding decision regarding the Grid topology (disconnect or connect lines, split nodes, etc) to ensure correct and secure behaviour. It can simulate the result of its action with Grid2Opt and decide based on its results.

Results

We will now briefly describe the approaches chosen by the best three teams.

Maze RL (winner)

This team used an Alpha-Zero RL approach for topology optimization (as described in detail in their ArXiV paper). However (they are experts in the field), they also used domain knowledge, like a contingency-aware redispatching, curtailment and storage controller. More precisely, they manually designed the value function of the algorithm, whereas basic AlphaZero learns this value function as one Deep Network. Also, though several simplifications were made to the original AlphaZero algorithm to decrease the computational cost (in particular, some early stopping was added to limit simulations), the amount of compute of this approach is still huge, much larger than that of the other participants.

Richard Wth

Interestingly, this team developed two agents: One using PPO (Proximal Policy Optimization), state-of-the-art Reinforcement Learning technique, and a Brute Force agent, that does topology optimization by brute force searching among actions and substations, and optimising the generators and storage by linearizing the Optimal Power Flow problem (the well known DCOPF approach). However, the brute force agent outperformed the PPO agent and was in the end the one they submitted.

Team HRI

Along similar lines, this team developed a stochastic greedy agent that picked up the best among 1000 randomly chosen actions, and backed up to line disconnection (again, greedy choice) if none of the 1000 actions tried before resulted in a correct behaviour. This agent was a purely reactive agent, involving no learning.

LOR

These three results somehow demonstrate that

- The problem might have been too easy, if brute force or systematic trials in the search space could arrive at reasonable solutions, better than PPO in generalisation (unsurprising, as generalisation is not an issue for brute force approaches). These approaches rely purely on optimization, plus some hidden (human) reasoning when domain knowledge is involved.
- On the other hand, the sophisticated AlphaZero approach, plus some domain knowledge, obtained the best results: There was some gain to be made but using a compute-intensive approach coupled with symbolic human knowledge. Ablation studies would be needed to see what is the real source of performance here.

Altogether, this challenge illustrates the power of adding symbolic knowledge to even the most powerful Learning algorithm (AlphaZero here), but that Optimization could somehow "replace" Learning (also coupled with symbolic knowledge) to obtain reasonable results, if not optimal.

TAI

There was no particular concern here about Trustworthiness. Trust in the agent was enforced by an alert system that had been added in the 2021 edition of the Challenge, and was part of the context without any emphasis put there (e.g., no part of the objective function was related to this alert).

Links

- The description of the setup
- The Codalab web site of the competition, with the leaderboards of all 4 phases.
- Detailed description of the results

Meta Learning from Learning Curves 2 (MetaLearn 2022)

This challenge belongs to the MetaLearn series, and is described in detail in Deliverables 2.3 and 2.6. During Meta-learning, for many different datasets, beyond the meta-features of the dataset and those of the algorithm (including its hyperparameters), the learning curves for the training, the validation and the test sub-datasets are available to the participants for all algorithms of the portfolio. The main originality of this challenge is that these Learning Curves represent the performance of the algorithm as a function of the proportion of the whole training set that is actually used for learning (see the figure in Deliverable 2.6). At meta-testing time, new datasets are presented to the agent, that must designate at each time step one algorithm from the portfolio together with the proportion of the dataset to be used in the following episode, given the performance at previous time step on both the training and the validation sets. The episode stops when the computing budget (unknown to the participants, to favour "any-time" behaviour) is exhausted. The Area under the Learning Curve is then computed on the test set (that was never seen by the participants) using at each time step the model with the best performance on the validation set – and taking the worse of three runs with different random seeds.

Results

Rank	Team	Average ALC score seed = 1	Average ALC score seed = 2	Average ALC score seed = 3
1	dragon_bra	0.39	0.39	0.39
-	ddqn_baseline	0.37	0.36	0.35
2	diaprofesser	0.32	0.34	0.36
3	carml	0.36	0.31	0.35

Final results - Areas under the Learning Curve.

First of all, it should be noted that only the winning team succeeded in outperforming the baseline DDQN provided by the organisers – DDQN (Double Deep Q-Network) being the current state-of-the-art in reinforcement learning, available off-the-shelf. This somehow demonstrates that the problem is rather difficult. Nevertheless, the first three teams were awarded a prize (sponsored by TAILOR).
Furthermore, to assess the robustness[4] of the proposed approaches, three runs of the submitted algorithms were run with three different random seeds, and the worst of the three was retained to compute the final score. In the table above, the results of the three runs for the three winning teams and the baseline are displayed.

Dragon_bra (winner)

This team used a well grounded approach. They first transformed the problem into a 0/1 knapsack problem, a well-known combinatorial optimization problem. But they could not directly apply dynamic programming, in particular because the total compute budget is not

[4] in the sense here of stability of the results with respect to the stochasticity of the algorithms, see the discussion in the final Section.

known a priori, as said (in order to favour the any-time behaviour of the solutions). Hence they used a greedy algorithm that is known to well approximate the solution of the knapsack problem, not relying on any known Machine Learning algorithm. As a consequence, their results are indeed stable, because of the deterministic nature of the greedy algorithm (once the parameters of the greedy algorithm have been determined during the meta-training phase). Interestingly, their approach was innovative enough that they could publish it in a conference paper at ICIC 2024 (Advanced Intelligent Computing Technology and Applications), where it is described in much more detail than in the Fact Sheet they submitted after the challenge.

Diaprofessor

This team ranked second, just below the DDQN baseline on average, but with some good results depending on the random seed. They use a pure supervised learning approach: They define features describing the training meta-data, combining the dataset meta-features, the data sizes, and the learning curve (score vs percentage of dataset). They then learn an ensemble of regression models (using Random Forest, AdaBoost, GBDT and ExtraTree) to predict the performance and the training time for different training sizes. At test time, they use this ensemble to select several models with highest performances for different data sizes, sorted by training time, and choose one randomly with higher probability for faster training time.

Carml

This team is also using a supervised learning approach, but pertaining to “learning to rank” and using a neural model. They use a contrastive loss function to perform end-to-end learning to predict the ranking of pairs of algorithms from their meta-descriptions, conditioned by the meta-data of the dataset. At test time, they use this neural model to rank the algorithms for the new dataset, for increasing proportion of the dataset, but perform re-ranking every time the proportion increases, using the latest results obtained on the current dataset.

LOR

By definition, Meta-Learning involves both Learning and Optimization, Learning as the main goal, and Optimization as the means to obtain the best Learning results. But here, the winning team made a critical use of Reasoning by turning the meta-learning problem into a combinatorial optimization problem, on which they applied classical optimization algorithm (here, a very basic greedy algorithm). One more example where the coupling of Learning and Reasoning did obtain excellent results - even if the standard RL approach (DDQN) was almost as good, and in any case better than the other two winners who stuck to supervised learning approaches.

TAI

Again, Trustworthiness was nowhere to be seen in the definition of the challenge, and hence did not appear explicitly in the results. However, because they used Reasoning by turning the Meta Learning into a combinatorial problem and later using a deterministic optimization

algorithm, the winning team obtained as a side effect the maximal stability offered by reproducibility, and stability one of the bricks of trustworthiness.

Links

- The Codalab web page of the Data Challenge.
- An ArXiV paper describing the setup of this Data Challenge (and the results of the first round).
- Results - a public report
- The ICIC 2024 paper describing in detail the solution of the winning team
- A recent paper (Sept. 2024) summarising both “Meta-Learning from Learning Curves” challenges published in Pattern Recognition Letters.

Cross-Domain MetaDL (MetaLearn 2022)

This academic Data Challenge was described in detail in Deliverable 2.3, and its results quickly surveyed in Deliverable 2.6. Furthermore, the whole challenge with a detailed analysis of the results has been published by the organising team (and co-authored by the winning teams) in the NeurIPS 2022 Competition track. We will only briefly recall here the basic traits of this challenge and its results, focusing on the lessons with respect to TAI-LOR that can be drawn therefrom.

As a reminder of the context, the introduction of this Challenge in Deliverable 2.6 reads
The context is that of Computer Vision, and the first round MetaDL challenge focused on transferring knowledge between tasks of the same domain so only small data is needed to learn new tasks (aka within-domain few-shot learning). The aim was to efficiently learn N-way (number of classes in a task) k-shot (number of examples per class) tasks, for given N and k. This second competition challenges the participants to solve "any-way" and "any-shot" problems drawn from various domains chosen. Last but not least, these domains were chosen for their humanitarian and societal impact (healthcare, ecology, biology, manufacturing, …).

Results

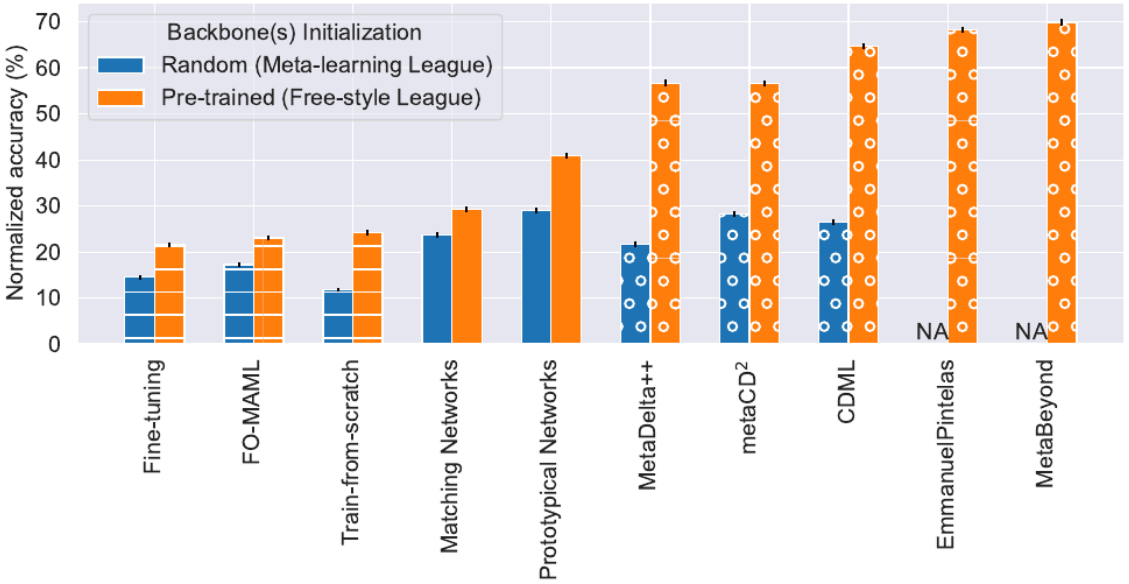
First, remember that there were two main leagues[5] Free-Style league, in which the use of pre-trained backbones was allowed, and the Meta-Learning league, where all weights of all layers of all subnetworks of the candidate architecture had to be initialised randomly.

The figure below gives a global overview of the results. The y-axis represents the balanced accuracy (bac) (also known as macro-averaging recall), normalised with respect to the number of ways N - the metric that was used throughout the challenge to take into account the any-way any-shot objective of the challenge (formal presentation in the NeurIPS 2022 paper). Blue bars are the results of the Meta-Learning league, and orange bars those of the Free-Style league.

From left to right on the figure, are the six baselines that were provided by the organisers, including MetaDelta++, the winner of the first edition of this challenge, and the four winners, that were the only four methods (out of 47 teams who submitted their solutions) to outperform MetaDelta++ in the Free-Style league, while no application succeeded in outperforming the Prototypical Network baseline in the Meta-Learning league (some teams did not even submit there)!

The first three baselines use a linear classifier, next two use a nearest centroid classifier, and the rest (small circles) is based on MetaDelta++, the winner of the first MetaDL challenge (including MetaDelta++ itself as a baseline, best of the baselines for the Free-Style League). Note that all baselines are based on a REsNet-18 architecture (and backbone when allowed), except MetaDL++ which uses a ResNet50 architecture.

[5] There were three other leagues thought as incentives to participate for ill-represented communities (women, poorly-represented countries, new-in-ML).



Another look at the global results is given in the Table below, that displays in particular the results of the 3 runs for each method, and for both the Free-Style et the Meta-Learning leagues (plus the prizes, provided by TAILOR). Remember that the final ranking was based on the worst performance out of three independent runs with three different random seeds.

League	Rank	Team	Average Normalized Accuracy seed = 1	Average Normalized Accuracy seed = 2	Average Normalized Accuracy seed = 3	Potential Prize
F	1	MetaBeyond	0.700 ± 0.007	0.700 ± 0.007	0.699 ± 0.007	400€
	2	EmmanuelPintelas	0.682 ± 0.007	0.687 ± 0.007	0.686 ± 0.007	250€
	3	CDML	0.646 ± 0.007	0.647 ± 0.007	0.650 ± 0.007	150€
	4	metaCD^2	0.566 ± 0.007	0.568 ± 0.007	0.569 ± 0.007	
M	1	metaCD^2	0.291 ± 0.007	0.286 ± 0.007	0.283 ± 0.007	400€
	2	CDML	0.265 ± 0.006	0.275 ± 0.006	0.267 ± 0.006	250€

Let us now take a quick look at those four winning solutions (more detailed descriptions, written by the team members themselve, are available in the NeurIPS 2022 paper).

MetaBeyond (winner)

This team designed a Light-weight Task Adaptation Network (LTAN) by integrating multiple lightweight task-adaptation modules with two generalizable pre-trained backbones (ResNet-50 and PoolFormer). During meta-training, they add an MLP-based classification layer after the backbones, and fine-tune only the layers upstream. Then, they replace the MLP classifier with a prototype-based classification head during meta-testing. Additionally, to

deal with the domain gap between the 10 domains of the competition, the backbone models remain frozen at meta-test time, and only some task-adaptive parameters attached to each backbone were learned. A great strength of this method is that these task-adaptive parameters are different for each backbone. Furthermore, to reduce the computational cost and running time, LTAN uses Automatic Mixed Precision in the task-adaptation process.

EmmanuelPintelas

This team introduced two innovations: a new Augmentation and Validation Optimization Pipeline scheduler to improve the training performance of **any** CNN-based model; An ensemble of Distance-based and Linear-based ML models. During meta-training, to force the backbone (SE ResNet152D) to generalise knowledge, Circular Augmentations are applied, i.e., in each epoch, a different subset of transformation functions are applied to the batch of images in a looping way. Aso, to avoid ending up in local minimum solutions, they apply simple backtracking optimization: in the end, they keep the parameters of the ones that lead to the highest validation score during training. One of the biggest strengths of this solution is the ensemble of Distance-based and Linear-based ML models at meta-test time based on the task configurations.

CDML

This team improved MetaDelta++ baseline by fine-tuning three models pre-trained on ImageNet during meta-training: ResNet50, SE ResNeXt50, and SE ResNeXt101 with anti-aliasing filters. Unlike MetaDelta++, this solution does not include random cropping as part of the data augmentation techniques since the random cropping may result in losing critical information. Additionally, one of the key points of this method is that for fine-tuning the two SE ResNeXt models, a combination of a supervised cross-entropy loss and a triplet margin loss is employed to improve the feature representations. During meta-testing, the feature representations of each backbone are concatenated and then transformed using the self-optimal transport algorithm. Finally, the transformed features are classified by a soft k-means algorithm.

metaCD^2

This team enhanced MetaDelta++ baseline for the Meta-learning league by using an attention-based contrastive spatial contrastive loss, motivated by the state-of-the-art, where it is shown that this loss can improve the generalisation ability of the models. However, due to the nature of contrastive learning, this approach can lead to over-clustering the features of the same class. Therefore, in the Free-style league, they use a contrastive distillation approach to compensate for the excessive disentanglement induced by the attention-based spatial contrastive loss and help the model stabilise. Moreover, they also improve regularisation by computing the predictions of the teacher model from weakly-augmented versions of the meta-training instances, trying to match them to the strongly augmented versions of the teacher model.

LOR

First of all, remember that this challenge is a Learning challenge -- even if concerned with Meta-Learning.It is hence clear that the main focus of the challenge, and of all applicants, is learning.
However, the NeurIPS paper, beyond giving the results of the final phase, also presents the results of ablation studies regarding four components that the four winning teams all used, one way or another: pre-training of backbones, data augmentation, domain adaptation techniques, and optimization, gradually introduced one after the other in that order. They also draw some lessons from these results - from the point of view of Meta-Learning, and not that of Learning, Optimization and Reasoning. Nevertheless, these lessons give us some insight about how the winning teams incorporated some LOR ingredients in their solutions.
The main conclusions of the organisers are three-fold. First, using pre-trained backbones is essential, improving the results of all teams by around 40 points (in percent). This is clearly an ingredient pertaining to pure Learning.
The second very positive conclusion is that the Optimization part of the winning pipelines is also critical -- some smart optimization procedure applied where needed at the end of the training process.
Finally, Data Augmentation and Domain Adaptation can have mixed results, and are beneficial only when applied in a smart way (e.g., through cyclic handling of transformations). Though not explicitly Reasoning, such a step calls upon background knowledge and clever applications of standard tools (data augmentation, domain adaptation) that require some human reasoning to be triggered right.

TAI

Trustworthiness was never an explicit topic in this challenge However, improving the cross-domain performances of AI models clearly goes in the direction of more trustworthy models. And this will be even more true in the next challenge, yet to be launched (and without TAILOR, that ended before it could have started), as it will get closer to real-life situations by proposing different domains during meta-training and meta-testing phases -- another step toward an AI you can trust, even in out-of-distribution domains, and even if this series of challenges only deal with image recognition tasks.

Links

- The Codalab page of the Data Challenge
- A didactic tutorial that runs on Google Colab
- An ArXiV paper describing competition design and baseline results
- The results presented at NeurIPS 2022

Brain age prediction challenge

Brain age prediction from brain recordings is a powerful indicator of several neuro-psychiatric diseases, but is generally done from structural and functional magnetic resonance imaging (MRI). This challenge addresses the problem from EEG signals, much more affordable. A more detailed description of the challenge, together with the datasets used and the evaluation process is available in Deliverable 2.6.

Results

The Brain Age Prediction from electroencephalogram (EEG) recording data Challenge, concluded with impressive participation and results. Over 180 competitors engaged in this intense machine learning competition, generating more than 500 submissions as they attempted to estimate the brain age of 2,000 subjects

The challenge culminated in a high-stakes final phase, where over 20 top-performing competitors were tasked with predicting the age of 400 subjects, limited to a single submission. This format tested not only the accuracy of their models but also their ability to provide clear explanations of their approach, with code and instructions to replicate the results.

Additionally, a special jury prize has been set to recognize interesting work that might have otherwise gone unnoticed. The jury prize recipient's story is particularly noteworthy. Despite ranking last on the final leaderboard due to a submission error, their model's performance would have secured second place if submitted correctly. This underscores the importance of both model accuracy and careful execution in competitive machine learning challenges.

This competition has demonstrated the potential of AI in predicting brain age from EEG data, which could have significant implications for the development of computational psychiatry diagnosis methods. The success of this challenge highlights the growing intersection of neuroscience and machine learning, paving the way for innovative approaches in understanding brain development and potential early diagnosis of neurodevelopmental disorders.

Final results
1st place - 1000$: team **tsneurotech** (MAE score: 1.156811)
2nd place - 500$: **MethodA** of team **State++** (MAE score: 1.600948)
3rd place - 250$: team **thatsvenyouknow** (MAE score: 1.603094)
jury prize - 250$: team **robintibor** (MAE score*: 1.4453888)

The leaderboard for the final phase is available on Codalab
https://codalab.lisn.upsaclay.fr/competitions/8336#results

We will now give a brief overview of the two winning approaches.
Both teams performed a number of problem-specific expert data-handling operations before any learning was applied, and did some model selection before choosing their final learning method. Team State++ also reported some unsuccessful tentative, something useful for

others, even if not providing any increase of performance (and sometimes even degrading the performances!).

tsneurotech (winner)

Preprocessing

- Dropped recordings with more than 30 bad channels
- Applied bandpass filter between 0.1Hz (low), and 45Hz (high)
- Downsampled the input signal to 100Hz
- Interpolated remaining bad channels, which worked better than dropping the channel locations

Model Selection

Different types of models were tried:

- "Classical" machine learning: Support Vector Regression (and its Riemanian variant), Gradient Boosting Regressor.
- Deep Learning, from the ConvNet architectures: EEG Resnet (a 2D ConvNet), Dilated ConvNet, Conv1DNet

Classical machine learning models bottlenecked at an MAE above 2 (see also next team). The 2D ConvNet required too much GPU memory to be practical.

Hence the 1D ConvNet was chosen, and gave the best results overall with the following key features

- A 3-layers architecture was enough – additional layers did not bring any improvement
- 1D ConvNet encodes temporal priors but not spatial priors, which would allow learning relationships between channel locations that are not directly adjacent to each other: Such relationships might not be critical to learn.
- Dilated kernels proved to be the best choice, with a dilation rate of 3, to increase the receptive field without increasing the number of parameters
- Max pooling and adaptive average pooling for temporal information aggregation gave the best results
- GELU activation functions displayed an improved performance over ReLU. GELU is based on a Gaussian cumulative density function, which provides smoothness and differentiability across the entire real line.

State++, MethodA

This team performed a number of problem-specific expert data-handling operations before any learning was applied. They also reported some unsuccessful tentatives, something useful for others, even if not providing any increase of performance (and sometimes even degrading the performances!).

Data Augmentation

- In addition to the dataset provided for the challenge, they used resting state EEG recordings from the open-access "Health Brain Network" dataset.

Data Cleaning

- Dropped recordings with more than 30 bad channels

- Applied interpolation to remaining bad channels
- Applied bandpass filter between 0.1Hz (low), and 45Hz (high)
- Downsampled all recordings to 100Hz
- Split data into epochs

Data Transformation

- Converted data to Power-Spectral Density (PSD) representation, using Welch's method, which improves upon the standard periodogram approach by reducing noise in the estimated power spectra, at the cost of reduced frequency resolution
- Scaled all signals to zero mean and unit scale

What didn't worked

- Downsizing the data to a subset of the 64 "best" channels, i.e. with the least occurrences of being bad for all subjects.
- Dimensionality reduction by applying a Principal Component Analysis (PCA)
- Reducing noise by selecting oscillatory patterns over non-periodic parts of the signals, by using the FOOOF library, which is a physiologically-informed tool to parameterize neural power spectra.

Model Selection and Optimization

- Chose Support Vector Regressor (SVR), which delivered better results over Random Forest and other classical machine learning models
- Utilised nested cross-validation in combination with grid-search for hyperparameter optimization.This approach is computationally expensive; suggested alternatives include:
    - Iterative testing on small subsets of hyperparameters
    - Decreasing the number of inner folds to increase speed

LOR

Before any learning step, all teams, even the ones not detailed above, applied preprocessing steps to clean the input data, extract a clearer signal from the noise and make it easier for the machine learning step. This allowed them to then use simple machine learning approaches. Furthermore, whereas the winning team tsneurotech had to turn to Neural Networks, though using an elementary 3-layers ConvNet in the end, the second team State++ did much more sophisticated pre-treatment (including data augmentation and expert data transformation) and obtained their best results using Support Vector Regressors. But we don't know if the 3-layers ConvNet would have allowed them to beat the winner thanks to their sophisticated pre-treatment.
In any case (human) reasoning as a pre-processor was key to success here.

Regarding Optimization, both model selection and hyperparameter tuning pertain to optimization, and were used by all ranked teams.

TAI

In addition to submitting their results on the private test set, participants had to provide their code, with a description and explanations of their approaches. Their explanations and the explicability of their approach were taken into account by the jury to make a final decision about the ranking.

Links

- The Codalab web site of the competition, with the leaderboard of both development and final phases.
- The website of NeuroTechX Global Hackathon 2022

Sleep States

The competition addressed the growing need for accessible, consumer-grade devices capable of providing reliable sleep monitoring and analysis. Participants were required to develop models that could accurately classify sleep stages using Electroencephalography (EEG) data collected from IDUN Guardian Earbuds, bridging the gap between clinical-grade EEG analysis and consumer-friendly devices.
A more detailed description of the challenge, together with the datasets used and the evaluation process is available in Deliverable 2.6

Results

The competition saw active participation from only nine teams, showcasing the interest, but also the high level of skills required to address it. The winning team achieved an impressive F1 score of 0.55, demonstrating the effectiveness of their approach. Unfortunately, some technical issues arose, that prevented all teams from submitting their fact sheet, and only the winning team was recorded on the Codabench web site of the challenge, that was also the only one to provide access to their github repository.

We will now give a quick overview of the winning approach.

Aashish Khilnani (winner)

For this challenge, the winning team has used techniques from signal processing to extract features from the EEG signal.

Decomposition of the EEG signals into time-frequency domain features:

They used Empirical Wavelet Transform (EWT), an adaptive multi-resolution analysis based on some ad hoc wavelet dictionary, to decompose the signals into three Intrinsic Mode Functions (IMF). These IMFs offer superior time-domain resolution compared to traditional methods. From each IMF, the team extracted 11 distinct features: variance, skewness, kurtosis, Point to point range, Root mean square, Standard deviation, number of zero crossings, hjorth mobility and hjroth complexity, Petrosian fd and permutation entropy. This process resulted in a total of 33 features (11 features × 3 IMFs) in the time-frequency domain, providing a comprehensive representation of the EEG signals.

After decomposing the EEG signals using EWT, they further enhanced their feature set by using the Welch method to extract Power Spectral Density (PSD) across various frequency bands, as well as the ratios of power across bands, providing insights into the relative strengths of various brain wave types. This process yielded 9 more frequency domain features per signal.

Then, to capture the relationship between the two EEG signals, they calculated the correlation coefficient, adding one more feature to their set.

Final Feature Set:

The comprehensive feature extraction process resulted in a total of 85 features:

- 33 time-frequency domain features × 2 signals = 66 features
- 9 frequency domain features × 2 signals = 18 features
- 1 correlation feature

Machine Learning model

The team experimented with various classifiers to determine the most effective model, including SVM, Random Forest, Gradient Boosting and others. Among all tested classifiers, Random Forest proved to be the most successful to classify sleep states based on the feature set they built, yielding the highest F1 score.

LOR

Before applying any learning approach, some preprocessing steps of the input signal can make the learning task easier for machine learning algorithms. These steps include:

1. Cleaning the signal from noise and artefacts, treatments for missing values, default values, detection and removal of outliers, etc.
2. Features extraction, which can in some cases be done more efficiently with techniques and transforms from signal processing or other “classical” approaches based on expert knowledge than with deep neural networks.
3. Data augmentation, useful to provide more diversity of possible cases in the input data used for learning and compensate for potential biases in the input data.
4. Model ensembling. It can be interesting to combine several learning models with ensemble approaches like bagging, stacking, or boosting, to improve the performance and regularise the behaviour of the final model as well as to provide better generalisation abilities.

Such pre-processing steps are based on domain knowledge and can be viewed as symbolic reasoning.

TAI

Again, trustworthiness was not directly addressed by this challenge. However, Filtering methods from signal processing and data augmentation have been used to compensate for potential biases.
Data augmentation provides more diversity to the data, fulfilling areas of the density distribution that were under-represented, with new data points coherent with the already existing input data. This technique thus contributes to increase the confidence of the resulting trained model.

Links

- The Codabench web site of the competition, with the leaderboard of the development phase.
- github repository of the winning team

Automated Crossword Solving: WebCrow

This challenge is based on the WebCrow 2.0 platform, the most recent version of the WebCrow project, that has been going on in Prof. Marco Gori's team at Siena University (TAILOR partner #37) since the early 2000s. Below are the links that have been put in Deliverable 2.6 to the complete information about WebCrow.
Let us first note that WebCrow itself is a perfect illustration of a successful hybridization of Learning, Optimization and Reasoning. This is particularly obvious in the 2.0 version, which is totally modular, and allows the programmer to easily add modules in the pipeline. The basic version starts with a Word Embedding module, that learns a specific word embedding dedicated to crossword puzzles, then does a Web Search helped by a Knowledge Graph, generating a list of possible answers that is passed on to a grid filling module that uses some specific Constraint Programming Solver to optimise the grid filling. Hence it uses Learning (of the Word Embedding), Symbolic knowledge (the Knowledge Graph), and Optimization and Reasoning with the CP solver. And it is very clear that only the combination of these modules allowed WebCrow to deliver such astonishing performances.

However, WebCrow challenges are very different from all the other TAILOR challenges as described in Deliverable 2.6, in that they are Human-Machine competitions, in which Humans try to compete with the WebCrow program (or different instances of it). All the other challenges are data-based challenges in which the human participants compete against one another and try to come up with the best possible solution to solve the problem at hand.

To sum up, WebCrow is a typical LOR platform, developed during many years, that beats the average human crossword solver (but not the human experts … yet). And the concept of Trustworthiness is irrelevant, in the sense that the proof here is in the pudding: A crossword is successfully solved or not. Hence no particular lesson to retrieve from WebCrow as a TAILOR challenge regarding TAI.

Links

- The main WebCrow page
- The competition platform webcrow arena
- The WCCI'2022 competition page
- The github link for the Webcrow 2.0 agent platform
- Clue-answer datasets (English, Italian)
- An ArXiV paper detailing the French version

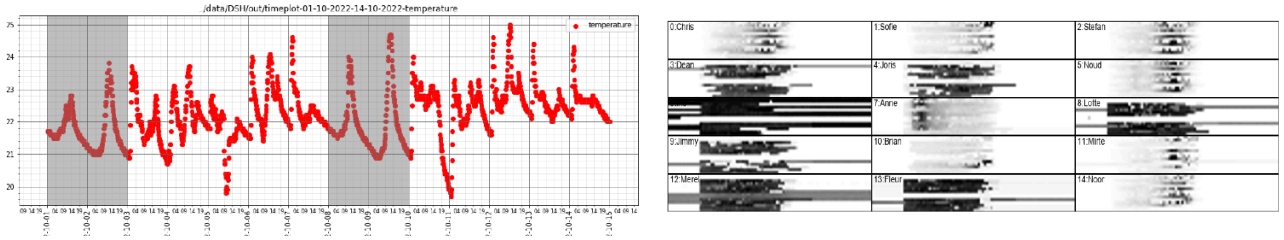
Mind the Avatar's Mind

The actual challenge that took place in March 2023 is described in detail in Deliverable 2.6.
It was more a hackathon than a challenge, spread over 3 evening sessions, and in fact entitled *Mind your Building*.
However, the initial and more ambitious goal of this challenge was related to *the Theory of Mind* - hence its title, *Mind the Avatar's Mind*. In that respect, this challenge is, remarkably, the only TAILOR challenge that was designed from the very beginning with TAILOR principles as goals, and exclusively by TAILOR partners: as discussed in the introduction of Deliverable 2.3, the first Deliverable describing TAILOR challenges, and reminded in the introduction of Deliverable 2.6, describing all TAILOR challenges, very few TAILOR partners answered our call for Challenge Data – and one was the Inductive Links Prediction challenge that turned out not to be run by TAILOR partners in the end due to some changes in Fraunhofer staff.

From the context of TAILOR WP6, Social AI, two TAILOR partners, TNO, DFKI studied the topic of computational Theory of Mind, and initiated the idea of combining their theoretical efforts with an industrial use case in the field of urban energy sustainability. Together with INRIA, they set up a data mining contest involving a multitenant smart building. TAILOR enabled the three organisations to think about such a context from three different angles : data modelling, industrial relevance and organising an event.
During the preparation and organisation they encountered their own challenge: mastering the topic such that they could be a frontrunning guide for the intended participants. Their solution was to split the idea into two parts: one part feasible enough to organise as a master of AI learning mechanisms and potentially a second part in which the approaches had to be further discovered. The first part ('Mind your Building') was straightforward to organise and the second part (intended 'Mind the mind') appeared to be challenging; it involved fundamental questions on how to formulate the problem, create validations with respect to candidate implementations of Theory Of Mind models. The second part of the challenge was therefore not announced by the organisers and postponed as a future candidate challenge.

LOR

The challenge (or hackathons) used as a starting point some real-world sensor data, such as the ones in the figure below (single sensor data on the left, aggregated sensors on the right,



to allow the user to get a quick notion of possible patterns. Because the first step of the challenge was a visual exploration of the data, i.e. some human reasoning allowing the user

to spot impossible values (e.g., temp>1000), and periods of “black-out” (during which no data was recorded, that could go from 1-3 days to up to 20 days).

The following step was some statistical Machine Learning, here mostly based on correlation analyses to detect, for unknown sensors, other sensors to which they are highly correlated and hypothesise that they are located in the same zone. So the whole process of the winning teams here can be seen as some hybrid between Machine Learning and Human Reasoning. But though tightly coupled with the Learning part, the Reasoning part was not in the algorithm itself, but came as a pre- and post-treatment of the Learning. Note that the second part of this task, to predict the occupancy, could not be addressed by lack of time, as the data was not labelled and the needed manual labelling was too time-consuming to be done on site during the hackathon itself.

While thinking of the follow up part, entitled *Mind the Mind* came, the organisers came up with ideas to let each zone of the building to be represented by an active digital twin, referred to as an Avatar, that can interact with other avatars (zones) and humans present in the building. “They create and use local machine learning models for learning and optimising their own tasks as well as choosing when and with whom they interact”.
Such an implementation would have implemented the LOR concept in a very original manner, but had to be postponed to a later occasion.

TAI

During the challenge, questions from both organisers, as well as participants, arise on the trustworthiness of sensor data, privacy of the data and the ways one could retrieve related information out of this data.
Though the visual data exploration and the removal of absurd measurement of real-world data pertains to increasing the likelihood of the data, and hence the robustness of the results. On the other hand, the Mind the Avatar’s Mind part explicitly addressed the explainability of the results. It is expected that one of the advantages of the Theory of Mind point of view compared to more traditional Learning approaches would even allow for more explainable results. This is why Explainability will be part of the evaluation of the provided outcomes of the teams and will enter their final ranking, along two more axes than pure performance.

Lessons

To this end, the value of the TAILOR project manifested itself not solely on the results of a winning team with a nice worked out solution, but more importantly.
As a matter of fact, the pressure of organising a challenge, forced the involved TAILOR researchers to collaborate and dive deeper in the topic of computational Theory of mind. As a result, this has led to three master-student projects on the application of computational Theory of Mind in different industrial areas.

By means of thinking about Challenges one is forced to think and rethink problems thoroughly. Working on a topic as a scientist doesn't mean one is able to organise a challenge around it. An overall inside TAILOR brought to us all.

Machine Learning for Physical Simulations

ML4PhySim and ML4CFD challenges

Two challenges are grouped under this title. The first one, ML4PhySim, started Nov. 2023 and ended March 2024, and is detailed in Deliverable 2.6. For the follow-up challenge ML4CFD (Machine Learning for Computational Fluid Dynamics), we decided to narrow the focus to the CFD use case, more familiar to all potential participants, to do a much wider advertisement, and to use the results of ML4Phyim as baselines. Note that these are the first lessons learned from ML4PhySim! It turns out that ML4CFD has been accepted as an official NeurIPS 2024 competition (see NeurIPS 2024 Competition Track paper). But it only started on July 1. 2024, hence after Deliverable 2.6 was delivered, and will end Oct. 31, 2024 - after this Deliverable has been delivered. Hence this section, while discussing the TAI-LOR aspects of both challenges topics and organisation, that are very similar, will only discuss the results of the ML4PhySim challenge.

Both challenges were run on Codabench, and the interested reader can check their respective Web pages (see the ML4PhySim Codabench page and the ML4CFD Codabench page).

Results

As briefly presented in Deliverable 2.6, the ranking was based on an aggregation of different metrics, computed on a standard test set, and on an OoD test set, namely ML-related quantities (accuracy of the computed fields on the whole domain, and speedup of the inference compared to the Finite Element Method OpenFOAM) and Physics-related quantities (Spearman correlations and mean relative errors for drag and lift computed from the fields around the profile). The results of the top 5 winners are summarised in the following table, where each criterion is first binned into 3 absolute qualities, and displayed as a colored dot: green for great value, orange for acceptable value, and red for unacceptable value.

		Criteria category							
		ML-related (40%)		Physics (30%)	OOD generalization (30%)				Global Score (%)
	Method	Accuracy(75%)	Speed-up(25%)	Physical Criteria	OOD Accuracy(42%)		OOD Physics(33%)	Speed-up(25%)	
		$\overline{u}_x$ $\overline{u}_y$ $\overline{p}$ $\overline{\nu}_t$ $\overline{p}_s$		C_D C_L ρ_D ρ_L	$\overline{u}_x$ $\overline{u}_y$ $\overline{p}$ $\overline{\nu}_t$ $\overline{p}_s$		C_D C_L ρ_D ρ_L		
	OpenFOAM	🟢 🟢 🟢 🟢 🟢	1	🟢 🟢 🟢 🟢	🟢 🟢 🟢 🟢 🟢		🟢 🟢 🟢 🟢	1	**82.5**
	Baseline(FC)	🔴 🟠 🔴 🔴 🔴	750	🔴 🟠 🔴 🟠	🔴 🟠 🔴 🟢 🔴		🔴 🔴 🔴 🔴	750	**32.85**
Rank	Preliminary Edition : Top 5 solutions								
1	MMGP [13]	🟢 🟢 🟢 🟢 🟢	27.40	🟢 🟢 🟠 🟢	🟢 🟢 🟠 🟢 🟠		🟢 🟢 🟠 🟢	28.08	**81.29**
2	GNN-FC	🟢 🟢 🔴 🟢 🟠	570.77	🟢 🟠 🟠 🟢	🟠 🟢 🔴 🟠 🔴		🟢 🟠 🟠 🟠	572.3	**66.81**
3	MINR	🟢 🟢 🟠 🟢 🟠	518.58	🟠 🟠 🔴 🟠	🟢 🟢 🔴 🟢 🔴		🟠 🟠 🔴 🟠	519.21	**58.37**
4	Bi-Trans	🟢 🟢 🔴 🟢 🔴	552.97	🟠 🟠 🔴 🟢	🟠 🟠 🔴 🟠 🔴		🟠 🔴 🔴 🟠	556.46	**51.24**
5	NeurEco	🟢 🟢 🟠 🟢 🔴	44.93	🟠 🟠 🔴 🟠	🟢 🟢 🔴 🟢 🔴		🟠 🟠 🔴 🟠	44.78	**50.72**

An immediate conclusion is that the overall winner, MMGP, achieves the best accuracies and physics-related results at the cost of speed, and hence might not be the preferred method if speed is the main issue.

We will now very briefly survey the approaches of these winners, focusing on the type of Machine Learning technique used, and the links with the symbolic part of the challenge (the PDE, the mesh, …). Note that GNN-FC, ranked second, has not yet sent the description of their approach, while NeurEco, ranked fifth, had said from the beginning that they did not

wish to publish their method, and it will hence remain private. We will hence only introduce MMGP, Min-R and Bi-Trans, which were ranked first, third and fourth respectively.

MMGP - Mesh Morphing Gaussian Process

The approach is described in detail in the corresponding NeurIPS 2024 paper. Its main characteristics are the use of Gaussian Processes and not of Deep Networks, like most other competitors), and a very tight intrication with the problem at hand.
The latter is based on some very specific mesh morphing technique to transform all the different meshes of the different examples into the same mesh on the same domain. They are first morphed to the same bounding box. Then the profiles are all morphed onto the profile of the first example (arbitrary choice), identifying intrados and extrados, and rotating the wake to be horizontal, then eventually projecting again everything on the bounding box. Finally, using standard FEM interpolation, all examples are projected on the same mesh, allowing all examples to be described by vectors of the same size that can be compared easily. Standard PCA (in the generic MMGP approach, replaced by POD in the case of airflow simulation) is then used for dimension reduction of all input and output fields. Gaussian Processes can then be trained to learn the input/output mapping in the reduced-dimensional spaces, and the resulting output is finally decoded by the inverse PCA (POD), and projected back onto the original mesh to obtain the complete desired output fields.
The whole process can be run on GPUs. However, the inference part involves the computation of the morphing and the projection on the same mesh than that obtained during the training phase, and this has a cost, resulting in the rather poor speedup when compared to the competitors. On the other hand, all operations are “close” to the physics, which probably explains the good results in terms of accuracy and, more impressively, of physics-related metrics.

Min-R - Multiscale Implicit Neural Representations

This method is a slight variant of the INFINITY approach (Implicit Neural Fields for INterpret-Ing geomeTry and inferring phYsics) as described in the corresponding NeurIPS 2023 workshop paper.

INFINITY is based on a latent encoding of all fields defined on the geometrical domain into a latent compressed representation, and a surrogate model mapping the geometrical and physical input data onto the physical outputs. The encoding extends Implicit Neural Representation to vector fields, using some Fourier basis vectors to better capture the different scales of the signals. The encoding is made of two parts, one that is common to all examples, and is modulated by a second one, which is specific to each example. The decoding is learned by solving an inverse problem, the target being the output fields of all examples (aka auto-decoding).

In the present context of airfoil simulations, the geometrical data are the coordinates (x,y) of points transformed into a signed distance to the profile and the directions of the normal projection on the profile, the physical input data are the coordinates of the velocity field at the entrance of the domain, and the physical outputs are the five physical fields defined above.

Also, the proposed approach, termed "multi-scale", slightly deviates (improves?) over the original INFINITY methodology: the frequencies of the Fourier base vectors are randomly drawn from a centred Gaussian distribution, and tuning the standard deviation \sigma of that distribution is critical with respect to the over- or under-fitting issue. Furthermore, each output field might require different values of \sigma. Hence the idea of using one \sigma per output field, generating intermediate representations that are later fused into the final latent representation.
Note however that no indication is given regarding how the different values of \sigma are chosen for each output field.

Sub-Sampled Bi-Transformer - Bi-Trans

This approach is described by the authors as “purely end-to-end machine-learning based”. Indeed, no physical consideration is taken into account here, and each example is considered as a set of point data, regardless of any global structure. They leverage the power of transformers to account for structure and interactions at all lengths - from local to global. However, the number of tokens here is the size of the mesh (the number of nodes), and the computational cost of a ‘standard’ transformer architecture with self-attention is quadratic in the number of tokens. In order to keep this computational cost reasonable, they use cross-attention between the full mesh and a ‘skeleton’, obtained by sub-sampling the full mesh respecting the local density of the nodes. Note that such sub-sampling has to be done anew for each example, as the meshes are a priori all different.
For meshes of size around 200,000 nodes, they typically use a skeleton of size 1000, 3 blocks of cross-attention with transformer MLP of [32, 64, 64, 64, 32], an encoder MLP of size [7, 64, 64, 32] and a decoder MLP of size [32, 64, 64, 32, 16, 4]. All activation functions are ReLU.
There is no constraint on the batch size, or the batching of the sets of points, and the typical batch-size used is 50,000.

LOR

The target of the challenge is to learn a surrogate model that will allow a fast inference of the solution of the RANS PDEs for any given geometrical (airfoil) and physical (inlet velocity) input. However, among the objectives, classically involving the accuracy with respect to the known outputs of the samples, are the lift and drag along the airfoil, that require some physical relevance of the results that cannot be guaranteed by the global accuracy of the velocity and pressure fields. One could say that these objectives require some reasoning (computing the lift and drag from velocity and pressure) as well as some optimization on top of that. Such a point of view might look a little twisted toward TAILOR moto, but it does have some consequences on the choices made at least by the winning approaches.

Interestingly, the degree of intrication of standard ML tools with problem-specific features (reasoning!) decreases with the ranking of the approach. The winning MMGP comes down to an airfoil-specific approach, and makes intensive use of the mesh and the FEM toolbox. The mIn-R method is more generic, in that it can be applied to very different PDEs where 2D (and probably 3D) fields are the quantities of interest. Note that it also explicitly uses an optimization algorithm to solve the inverse problem of the decoding. Last but not least, the Bi-Trans approach uses transformers and claims to ignore any structure of the examples,

considered as simple sets of points. However the authors did take into account the balance between local and global interactions between points, in particular close to the boundary (the airfoil), which requires this sub-sampling operation to avoid the quadratic cost of the self-attention mechanism.

TAI

MMGP achieves impressive results in terms of physics-related metrics, in particular for OoD examples. And when dealing with physical simulations, physical relevance of the solutions is mandatory, and hence a critical part of trustworthiness : no engineer will ever use a model that can give erratic results as far as physics is concerned (e.g., not respecting the conservation of mass or of energy). And because OoD test cases are unavoidable (it is virtually impossible to predict all possible scenarios in real-world situations), physical relevance is even more critical in OoD cases. The speedup only comes as a secondary criterion here: the weighting of the difference criteria in the challenge do respect such common sense arguments (though it was of course proposed before any result was available).
However, it should be clear that even 18 green points in the ML-related and physics-related criteria would not guarantee physical relevance in yet-to-come OoD examples. At this point in time, no formal guarantee is to be expected from the ML point of view, and further deep theoretical studies are still needed. On the other hand, the same guarantees than the FEM approaches could be obtained by re-injecting the solution proposed by the ML system into the FEM solver: if it is accurate enough, only very few iterations of the FEM solver would be needed, decreasing slightly the speedup for a huge gain in trustworthiness. More work is needed to find the optimal balance between FEM- and ML-models, and we are still far, from the trustworthiness point of view, of ML-models that would not need an FEM solver in that respect.
It should be noted, however, that the winning solution MMGP, being based on Gaussian Processes, also provides an estimation of the uncertainties of its results: This is a first step in another direction of trustworthiness: At least, the user is advised of the level of uncertainty the proposed solutions are likely to exhibit.

Links

- The ML4PhySim Codabench page
- The ML4CFD Codabench page
- The NeurIPS 2024 Competition Track paper

General Lessons

Most challenges run during the course of the TAILOR project are Machine Learning challenges, as Machine Learning is today the most visible and prominent part of the AI iceberg. But the main goal of TAILOR was precisely to demonstrate that AI is not limited to Machine Learning. And this was, too, the purpose of the Challenges as described in Tasks 2.3 and 2.4 of the original TAILOR proposal.

Unfortunately, and this is the first lesson learned, though para-scientific more than scientific, running a challenge is a demanding and difficult task, and almost no TAILOR partner or even second circle friend could find both the data and the human power to make it possible to create a “pure TAILOR” challenge. There were three exceptions, though, which were clearly emphasising TAILOR specificities in their goals and proposed means: the Inductive Link Prediction challenge, proposed by Fraunhofer soon after the kickoff meeting – but staff mobility at Fraunhofer decided the end of this adventure, at least as far as TAILOR was concerned; The Mind your Building challenge, proposed by TNO and DFKI – gathering the data proved a much more time-consuming task than expected, and the challenge was changed into the Mind the Avatar’s Mind hackathon and the ambitious idea of the full challenge is still pending; The WebCrow Crossword Solving challenge run by University of Siena – but this was a challenge for humans to compete with the WebCow platform, and not a regular challenge for AI expert programmers to tackle and benchmark the state-of-the-art on some cutting-edge AI problem.

All other challenges run with the help of TAILOR and detailed in Deliverables 2.2 and 2.6 were "classical" Machine Learning challenges. In particular, none had as main focus some hybridization of Learning, Optimization, and Reasoning. And very few directly addressed trustworthiness: Some of them, though, emphasised explainability of the proposed solutions in the final ranking of proposed models. But explainability had to be estimated by a human jury, and was considered together with standard efficiency, attested by some accuracy metric on some hidden test datasets.

The third type of link with TAILOR motos has come a posteriori, in the design and implementation of the best performing solutions (and of many others, less performing), that very often demonstrated the power of the LOR paradigm, and sometimes a posteriori also favoured trustworthiness.

LOR

It is clear that Learning is today the most prominent part of AI, and most challenges targeted a learning problem, aiming at improving the state-of-the-art in some specific aspect of Learning

It is also clear that Optimization and Learning are not separable: learning is nothing more than finding the optimal model that explains the data. But things are even more intricate when it comes to model selection and hyperparameter tuning: Indeed, most of the best-performing solutions of most of the Challenges co-organized by TAILOR (the Smart Mobility challenge, both NeurotechX challenges, both ML for Physical Simulations

Challenges) make a routine use of AutoML methods, from basic grid search for hyperparameter tuning to advanced Bayesian Optimization, as it has also become standard practice for all the practical ML implementations for real-world applications. Optimization is also the method of choice of the winner of the Meta-Learning from Learning Curves, that does not use Learning at all, even though addressing a Machine Learning challenge, but transformed the original Meta-Learning problem into a combinatorial optimization one, thanks to expert human reasoning. Also, all winning approaches of the Cross-Domain Meta-Learning Challenge have added a post-training Optimization step in their pipeline. There is hence no doubt that Optimization is a critical ingredient in all these approaches.

Things are not yet as clear when it comes to Reasoning. Nevertheless, though no challenge explicitly addressed the coupling of Learning and Reasoning, and no solution of the proposed challenges implemented some hybrid approach liaising Learning and Reasoning, most best performing solutions relied on **human reasoning**, based on **domain knowledge**, in the pre-processing of the data, from basic data wrangling to expert transformations of the data into new representations. We are not yet at the “robot scientist” level in this respect, but this clearly shows a way to go for further research (and further challenges), possibly automatically looping such type of reasoning and the learning process that follows.

TAI

Trustworthiness was only directly addressed in the objectives of the challenges through the lens of **explainability**, a first step toward transparency. Some challenges (e.g., the Smart Mobility Challenge, and the original Mind the Avatar’s Mind challenge) had announced that explainability would be a secondary criterion to help ranking the candidates – empirically judged by a human jury. Because there are no recognized metrics allowing to automatically rank algorithms with respect to their explainability ability, and proxies are not very satisfactory (e.g., the number of parameters of the solution, or the length of dedicated explanations), which makes the design of challenges addressing explainability almost impossible.

Another component of trustworthiness is **robustness**. However, there are several different points of view on robustness. For stochastic algorithms, a first point of view on robustness deals with their **stability** with respect to stochasticity, and is close to the idea of **reproducibility**: Practically, the variance of the results when the algorithm is run with different random seeds should be as small as possible. Another aspect of robustness is with respect to the input data: results should change gradually to the change in the input data. For learning systems, the latter covers the **generalisation** ability of the trained model to correctly predict the output for data drawn from shifting distribution, or Out-of-Distribution generalisation.

Robustness with respect to stochastic stability was partially addressed by both MetaLearning challenges in their design, as they considered the worst of three runs with different random seeds of the algorithms for their final ranking. Such practice could be easily extended to most challenges, and more sophisticated statistics could easily be proposed.
Along the same line, but a posteriori, the deterministic approach of the winner of the Meta-Learning from Learning Curves, that turned the problem into a Combinatorial

Optimization problem and solved it using a standard solver, reaches maximal stability, as it is not stochastic.
On the other hand, robustness with respect to the input data was addressed by definition by the Meta-Learning challenges, in which the evaluation function precisely rewards the performance on several domains, hence the ability to generalise over these different domains. However, only the next step of the Accross Domain Meta-Learning challenges (with completely new domains at meta-test time) will reward true OoD generalisation performances.

Conclusion

Several challenges addressing different aspects of AI (but mainly Machine Learning) have been proposed during the course of TAILOR, and even though none actually involved the hybridization of (at least two out of) Learning, Optimization and Reasoning in its goals, the analysis of the best performing solutions proposed by the competitors demonstrated that LOR is very often the path to choose to achieve the best performances – even though the reasoning part is "limited" to human interventions, and not really part of the algorithmic solution.
There is a critical need for designing a "pure TAILOR" challenge, and at least two directions should be followed. First, goals pushing toward neuro-symbolic approaches (e.g., around learning the multiplication tables from written exemples). Unfortunately, no TAILOR partner had the motivation and the human-power to fetch the necessary data (as discussed in the global context of TAILOR in the previous Section).
Also, interesting challenges could be proposed regarding the reasoning capabilities of LLMs. But the coming-of-age of LLMs and the experiments about their reasoning skills arrived too late in the course of TAILOR: Remember that TAILOR proposal was written before ChatGPT even existed – GPT1 had been launched recently, but initially thought to be of interest only to NLP people; and the actual LLM explosion started with ChatGPT in Nov. 2022, while TAILOR was initially supposed to finish end August 2023.

The trustworthiness point of view was also only lightly addressed in these challenges, as said in previous Section, and no challenge did address other aspects of trustworthiness such as fairness, privacy, transparency or accountability – all as difficult to quantify as explainability, as discussed above.

Another aspect of TAILOR Challenges is that, at least for those run on Codalab/Codabench, they de facto became Benchmarks at the end of the competition, with the opening of the so-called Legacy Phase: all datasets become public, and the leaderboard is updated with new submitted results. Hence, based on the present analyses, it should be possible to make new experiments involving the Open Source solutions (most of the winning ones) with a strong basis for comparison, e.g., some ablation studies that are sometimes missing to find out the important components of the proposed approaches (e.g., in L2RPN), or adding some automatic reasoning part to the existing approaches.