

CIVA DS and Simulation for Data Science



CIVA USERS
COMMUNITY EVENT



Outline

- **Introduction to CIVA DS**
- Key Features of CIVA DS
- Use Case Examples
- Roadmap and Future Perspectives

Introduction to CIVA DS

- **AI and Data Science are transforming industries:** *The rise of AI and machine learning is revolutionizing industrial applications, including Non-Destructive Testing (NDT).*
- **NDT can greatly benefit from AI:** *AI-assisted diagnosis and automation can improve detection capabilities and efficiency.*
- **Challenges remain:** *Key obstacles to overcome include :*
 - Data availability,
 - Reliability,
 - Explainability

Why CIVA DS ?

- **High-quality synthetic data generation:** CIVA can provide well-labelled data through fast and advanced simulations, reducing reliance on costly experimental data.
- **Flexible numerical experiments:** Users can design and test AI models with controlled parameters, allowing fast iteration and optimization, on large datasets
- **Powerful post-processing tools:** thanks to CIVA generic functionalities, CIVA DS allows data extraction, signal and images processing, advanced reconstruction tools, and structuring adapted for AI applications.
- **Connects simulation and real-world use:** CIVA DS helps ensure AI models are applicable and practical in production environments through its compatibility with acquisition system file formats and the ability to process performance demonstration studies considering a wide set of variables

Outline

- . Introduction to CIVA DS
- . **Key Features of CIVA DS**
- . Use Case Examples
- . Roadmap and Outlook

DESK of CIVA DS



Data Science

Data bases & Metamodels	Data bases Fusion	Python Notebooks
Classification	Outliers Detection	Inversion

- User manual
- Video
- EXTENDE CIVA
- About CIVA

Import and Manage Data

- Import data from multiple sources (.var, .cck, .txt, .hdf5, experimental data).
- Customize inputs and outputs for better structuring
- Data visualization for easy interpretation and verification

1. List of imported databases
2. List of inputs
3. List of outputs (data) and selection of output visualized
4. Global Visualization (Table or Parallel Plot are available)
5. 1D Visualization of outputs
6. 2D Visualization of outputs (available only for outputs with 2 or more dimensions)

The screenshot displays the software's main interface with several key components:

- 1:** A tree view on the left showing the hierarchy of imported databases.
- 2:** A table listing the imported databases with columns for Color, Name, Nb. Inputs, Nb. Outputs, Nb. In., and Size.
- 3:** A detailed view of a selected database showing its inputs and outputs.
- 4:** A large data table with multiple columns and rows, representing the global visualization of data.
- 5:** A 1D line plot showing a single data series over time or iterations.
- 6:** A 2D heatmap visualization showing data across two dimensions.

The 'Editor of extraction' dialog box contains a Python script for processing data:

```

    Tag [0]
    Label
    Type Script/Python
    Dispatch
    [ ]

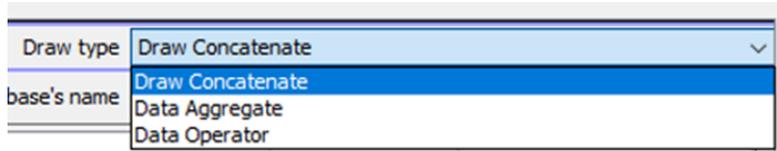
    linpute = get_input()
    x1 = get_output("RESI000")
    x2 = get_output("RESI")
    return x1 + sp.max(abs(x2)) - linpute[0]
  
```

Below the script is a 'SCRIPT' button with a double arrow icon, and 'OK' and 'Annuler' buttons at the bottom.

Process your data using a simple formula or **CIVA Script** (python)

Data Fusion

- Combine multiple datasets for *enhanced learning*.
- Merge experimental and simulated data.



Combine your Data

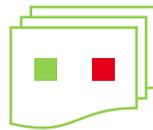
Draw concatenation allows to concatenate samples from several databases in one.



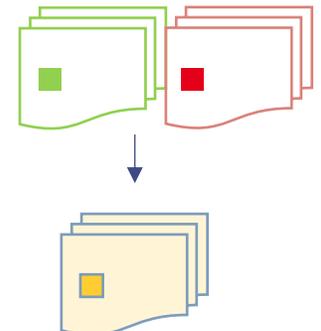
Data concatenation coming from different databases in one new output result of a concatenation of all outputs selected



Data Aggregation allows to concatenate outputs linked to same set of inputs values from several databases in one database



Data fusion allows to merge data from several databases.

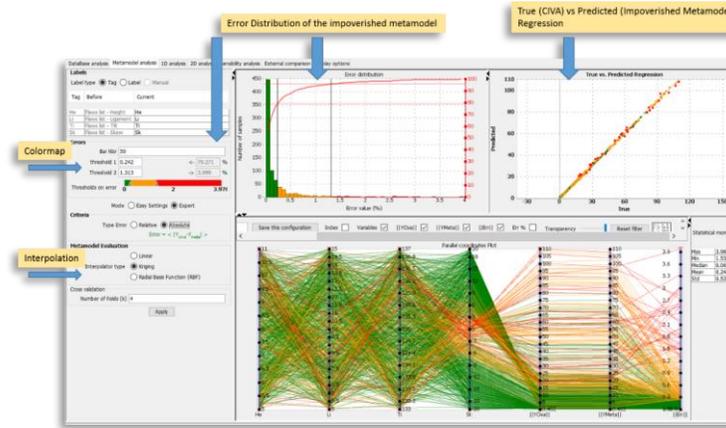


Metamodeling

Data bases & Metamodels

- Build a metamodel from any database (customized outputs, fusion, experimental data)
- Access to all metamodels feature (Accuracy Analysis, Sensitivity Analysis, POD, Prediction Accuracy)

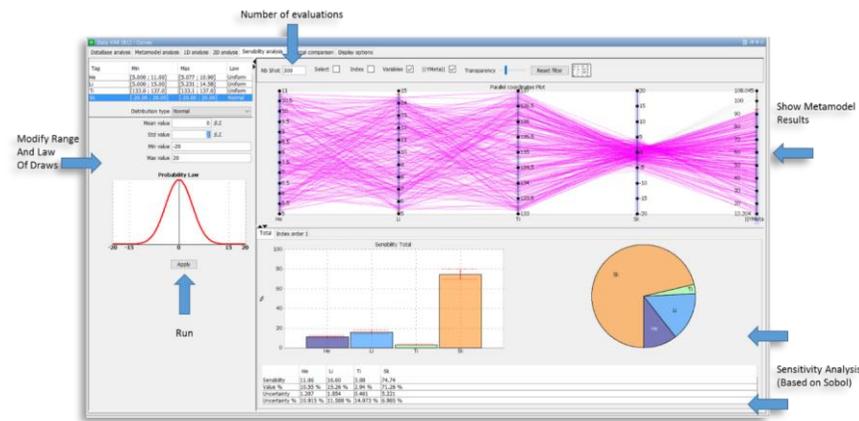
Build Metamodels from any type of Data



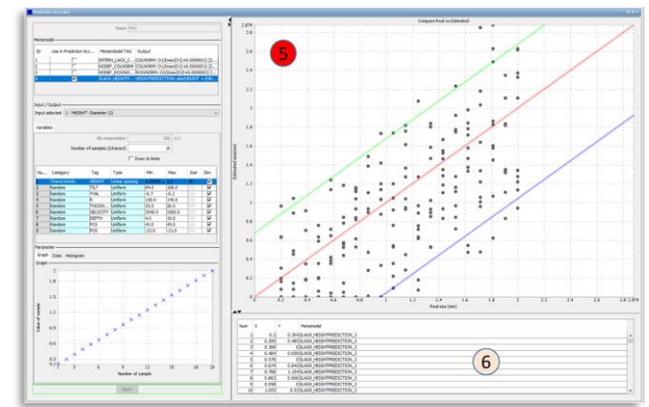
Metamodel Accuracy Analysis



POD



Sensitivity Analysis



Prediction Accuracy

Machine Learning for Detection



- Detect deviations from standard defect-free conditions.
- Binary discrimination for anomaly detection
- ROC curve analysis for performance evaluation.

The screenshot shows the 'Outliers Detection' software interface. It includes a table of defect types (1-9), a 'Method' dropdown set to 'Support Vector Machine', 'Energy normalization' checked, and 'Annotations' for training and testing samples. A 'Criterion and Threshold Analysis' window shows an ROC curve with an AUC Score of 0.94. A 'Data and Zones Analysis' window shows a heatmap of features over samples, with a corresponding 'Curves' plot showing amplitude over features.

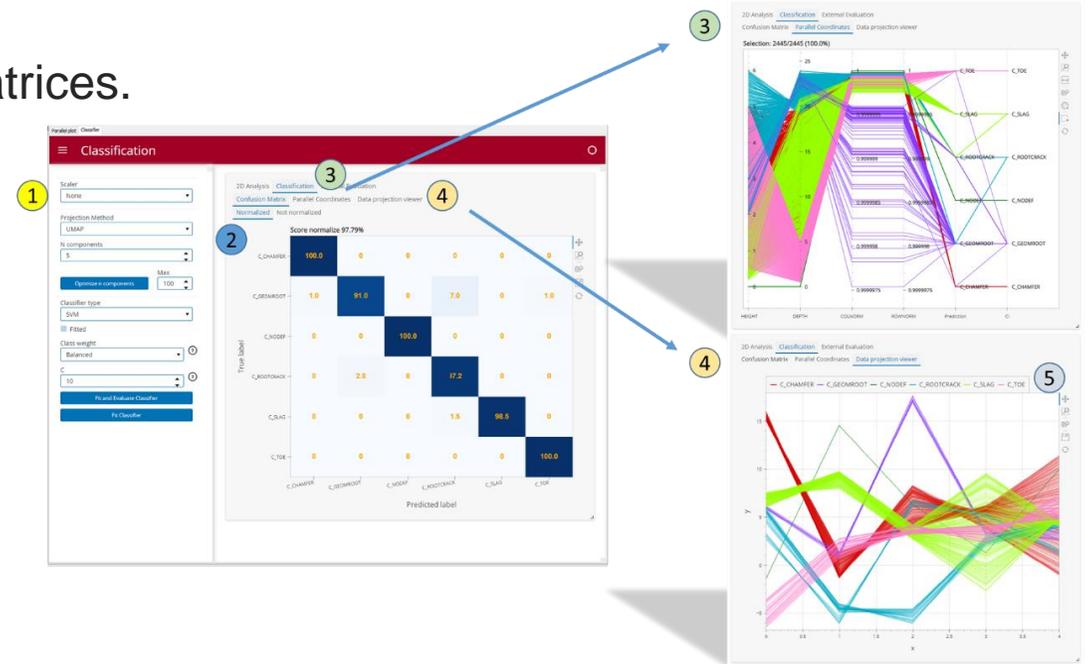
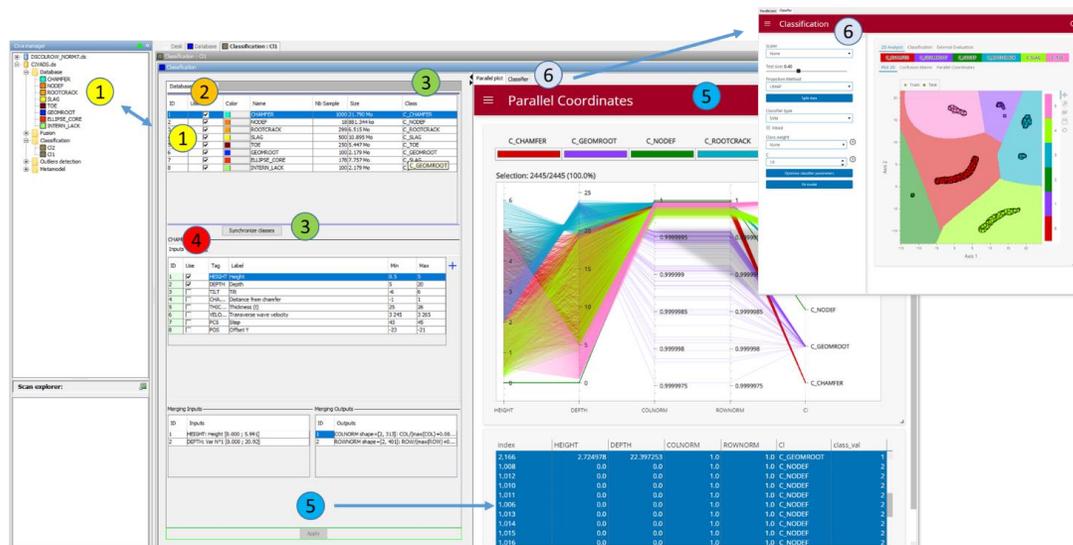
ID	Use	Color	Name	Nb Sa...	Size	Class
1	<input type="checkbox"/>		CHAMFER	1000	21,790 Mo	Standard
2	<input checked="" type="checkbox"/>		NODEF	38,881,344 ko		Standard
3	<input type="checkbox"/>		ROOTCRACK	299	6,515 Mo	Standard
4	<input type="checkbox"/>		SLAG	900	10,895 Mo	Standard
5	<input type="checkbox"/>		TOE	250	5,447 Mo	Standard
6	<input type="checkbox"/>		GEOMROOT	100	2,179 Mo	Standard
7	<input type="checkbox"/>		ELLIPSE_CORNE	178	7,797 Mo	Standard
8	<input type="checkbox"/>		INTERL_LACK	100	2,179 Mo	Standard
9	<input checked="" type="checkbox"/>		NODEF90	59	881,744 ko	Unknown

The outliers detection module allows to use databases for a “single class” training. The main idea is to learn “one situation” (for example “no defect situation”) and to detect outlier cases (for example “with defect situation”)

Machine Learning for Classification

Classification

- Select and structure data for classification models.
- Organize your data by class
- Compare algorithms for optimal performance.
- Visualize results using parallel plots and confusion matrices.



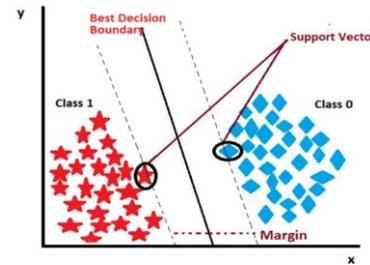
Unified Algorithm Testing Environment: A single platform for evaluating both standard algorithms and specific advanced algorithms with Plug-in capabilities

Machine Learning for Classification

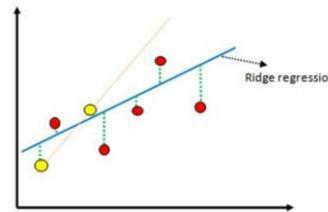
Classification

Classification Algorithms available in CIVA DS

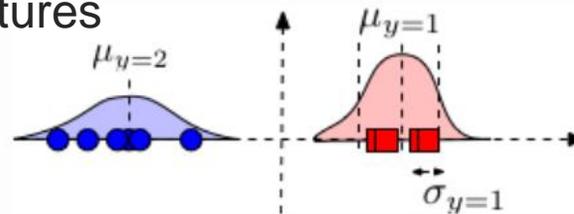
- **SVM** : Support Vector Machines
Find the optimal hyperplane that separates the data into different classes, while maximizing the margin between the classes.



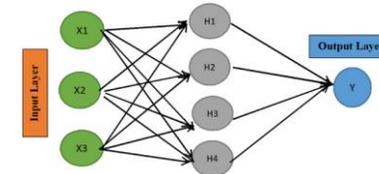
- **Ridge** :
Converts the label data into $[-1, 1]$ and solves the regression method's problem



- **Naïve Bayes** :
Look at how likely each feature is to belong in that category, assuming those features don't affect each other

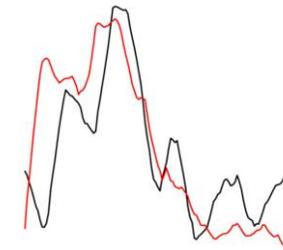


- **MLP** : Multi Layer Perceptron
Artificial neural network that learns to make predictions by adjusting connections between layers of artificial neurons



- **Time Series Correlation**

finds how much two sets of data that change over time are related

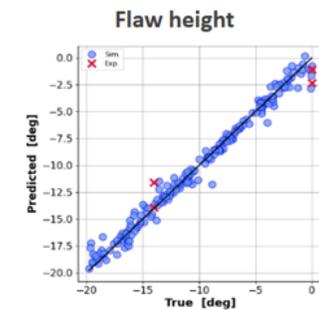
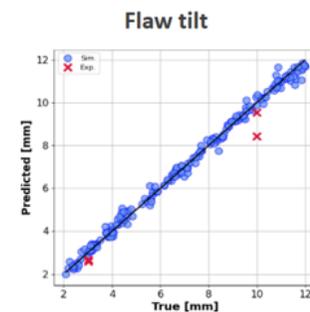


Machine Learning for Inversion / Regression

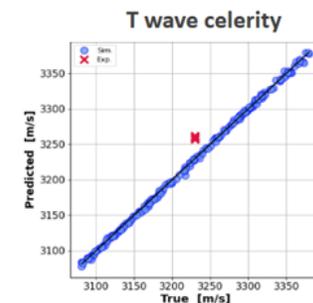
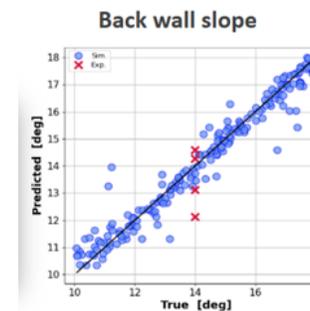
*Train regression models to predict continuous values.
Utilize simulated or experimental data to create accurate estimators*

- Select parameters to invert
- Select and structure data for inversion models.
- Compare algorithms for optimal performance.
- Visualize results using scatter plot with 95% and 5% percentiles of residuals.

■ Inversion results



In progress ...



Notebooks

The CIVA DS Python notebooks allow to offer a simple access for user to specific and advanced features not available through CIVA GUI
User can define its own notebook using CIVA Python API for easy data access , post processing and data base generation

Python Notebooks

SCRIPT

```
ret = []
if not np.isnan(array).any():
    array = extract(array,maxwindow,size,display) #applying treatment
ret.append(array)

#writing transformed data
with DbAPI.create_database_writer(tofile) as writer:
    n = np.shape(ret)[0]
    for i in range(n):
        writer.push_entry(i)
        writer.push_output(ret[i])
```

Input file definitions

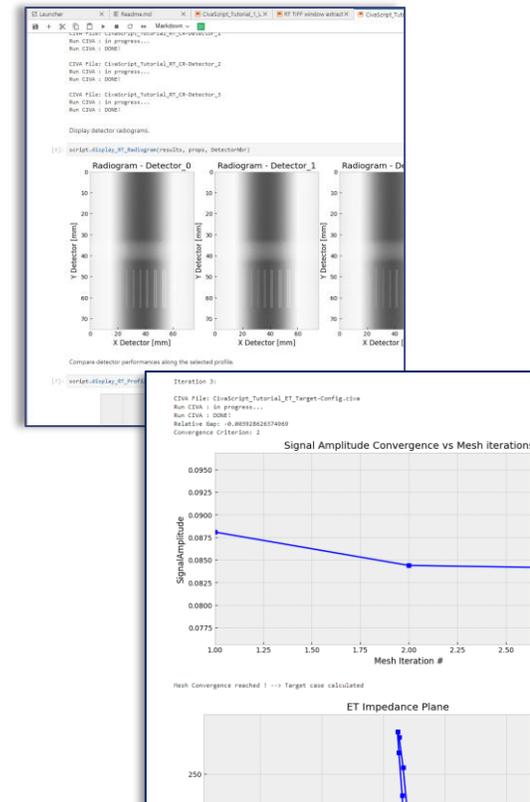
```
[4]: filelist = [r"resources\Attenuation.tiff"]
import os
outdir = r"C:\Temp"
```

Main program

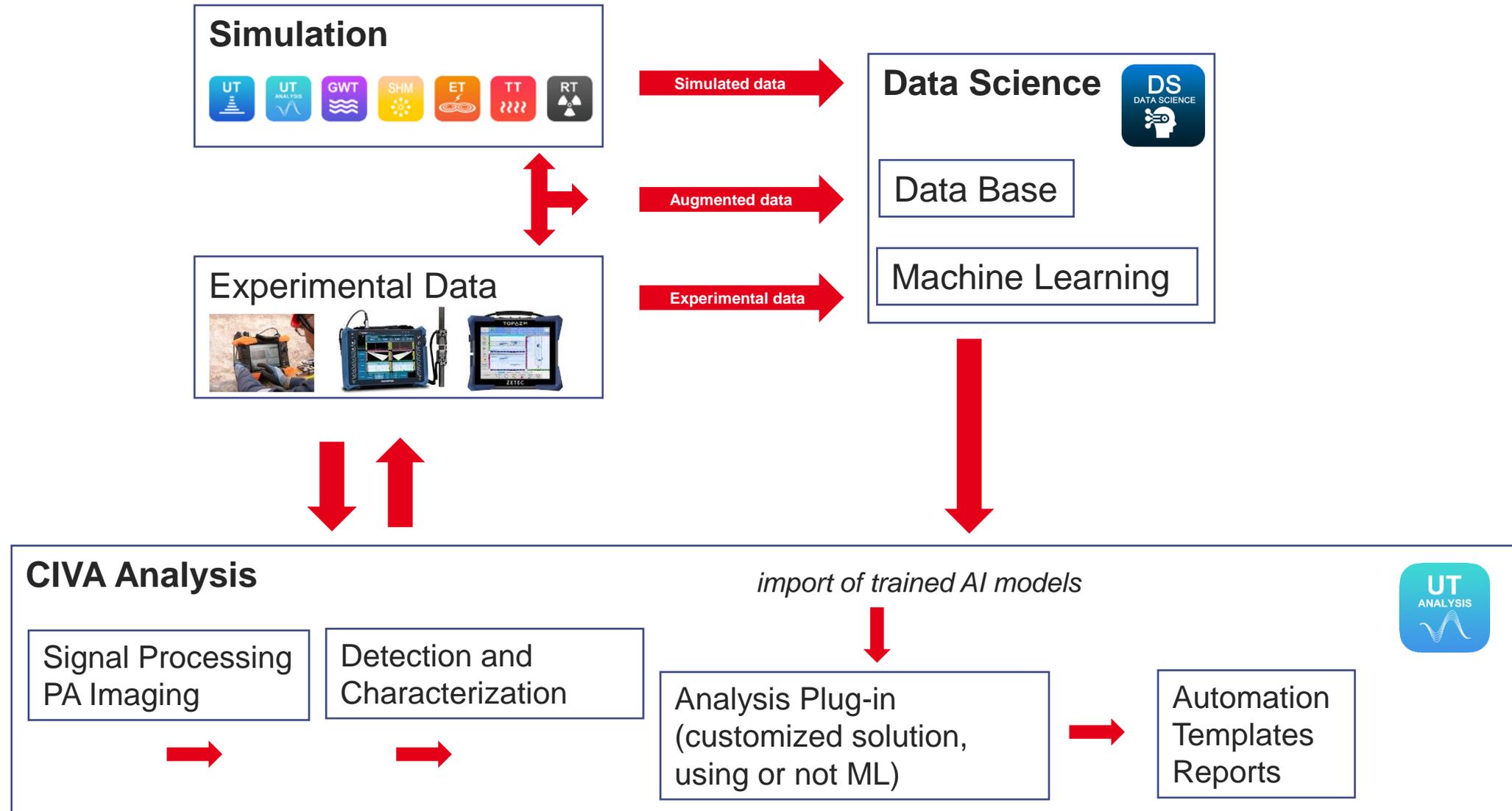
```
[5]: import os

disp=True
extract_window_size=48
window = (200,450,100,400)
if not os.path.exists(outdir):
    os.mkdir(outdir)
for filename in filelist:
    dirname = os.path.dirname(filename)
    basename = os.path.basename(filename)
    output = os.path.join(outdir,basename[:-5]+'_extract.cck')
    extract_file(filename,output>window,extract_window_size,display=disp)
```

The notebook interface shows a file browser on the left with a list of files including 'resources', 'CivaScript...', 'Readme.md', and 'RT TIFF win...'. The main area contains code cells and two plots: a line plot showing a sharp peak at approximately x=250, and a heatmap showing a vertical band of high intensity at the same x-position.



In Summary



Outline

- . Introduction to CIVA DS
- . Key Features of CIVA DS
- . **Use Case Examples**
- . Roadmap and Outlook

AI and simulation-based diagnosis



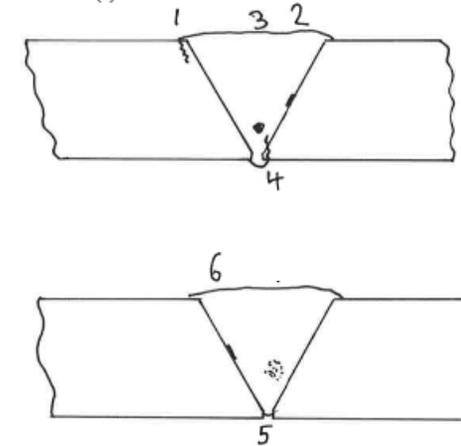
V-Weld 30°
Thickness 25.4mm
Steel tube
Diameter 300mm

NDE INSPECTION REPORT

ULTRASONIC

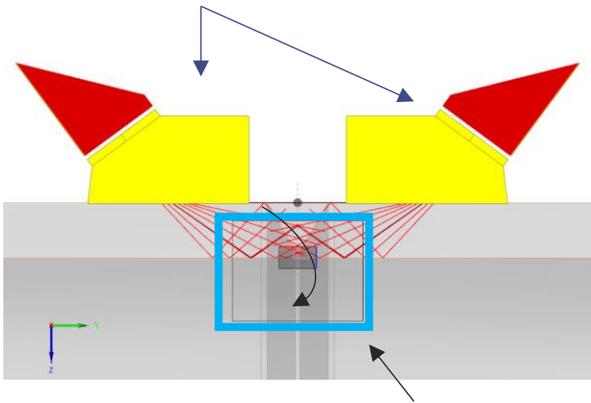
Customer	CEA Saclay	Date	28/03/19
Specimen ID	P 28163	Specimen Type	Pipe
Dimensions	25 THK. x 300 DIA.	mm	Acceptance Spec. SI/08/88

Weld/Specimen Cross Section(s)



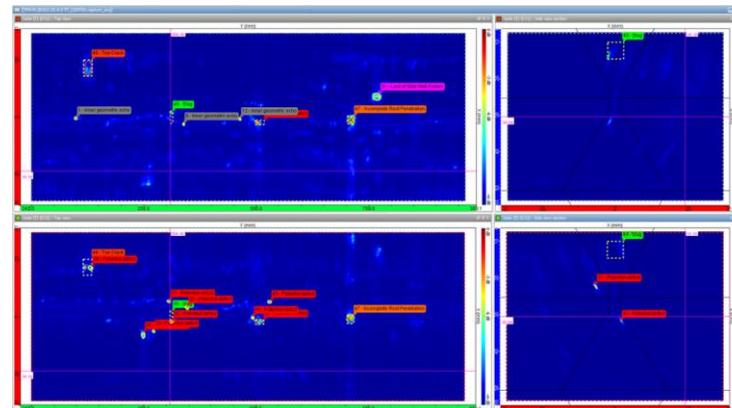
To Scale

2 PA Probes (US and DS) : 5MHz 64 elements : 38 mm x 10 mm
TFM-PWI inspection 43° to 79°, 6 angles



Optimized TFM zone for direct reconstruction (the echoes from the CAP zone will be detected after reflection => visible at the bottom of the zone).

Detection Threshold-12dB



Flaw No	Flaw Type	Flaw Length mm	Distance from 0 mm	Max UT Indication	
				dB	Angle
1	Toe Crack	24	113	+ 0	45
2	Lack of Side Wall Fusion	22	248	+ 11	60
3	Slag	16	307	- 6	60
4					70
5	incomplete root Penetration	18	709	+ 10	70
6	Lack of Side Wall Fusion	15	766	+ 11	60

Comments:

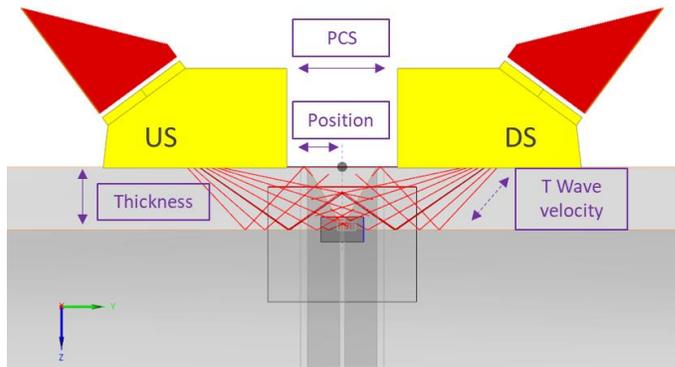
Misalignment noted.

Inspector Sam Berriman Signed



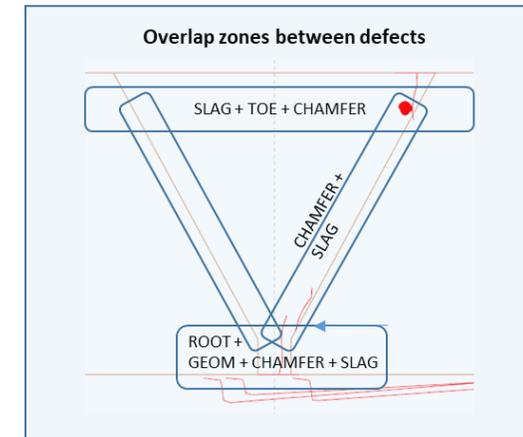
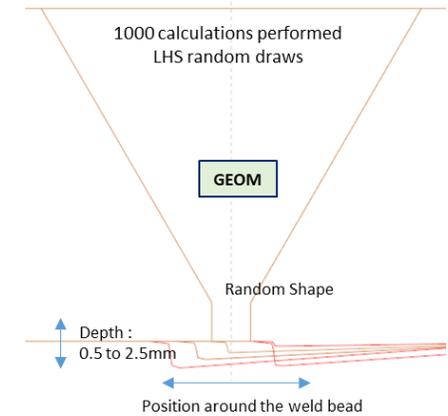
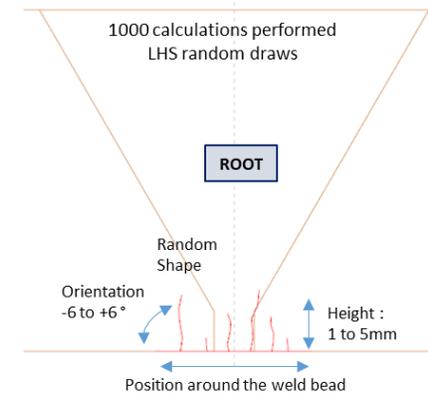
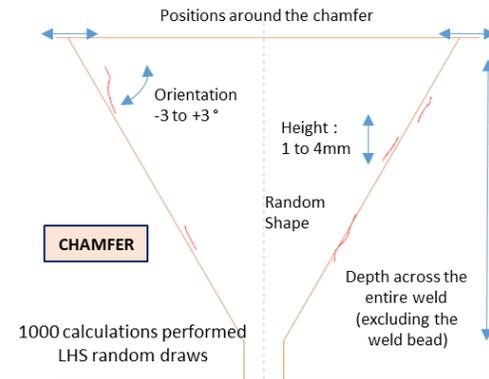
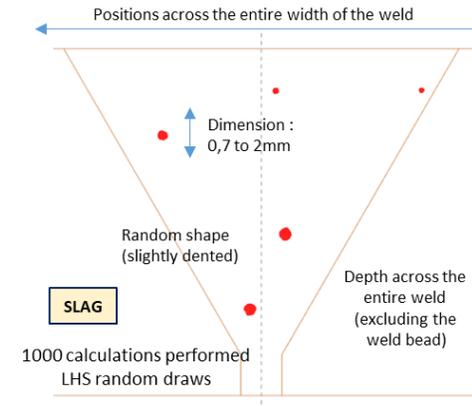
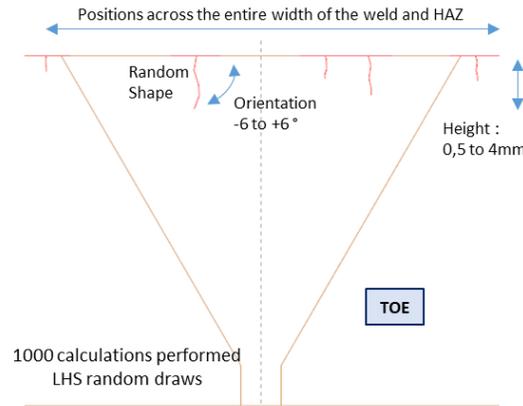
AI and simulation-based diagnosis

Learning from simulated data
=> Numerical experimental design

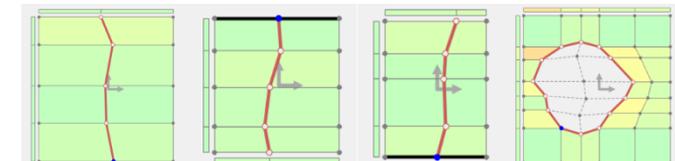


Essential Variables

- PCS : +/- 2mm
- Probe misalignment with respect to the weld: +/-2mm
- Thickness : +/- 1mm
- T-wave velocity : +/- 20 m/s



Defects (types, dimensions, positions, orientations, random shape)



AI and simulation-based diagnosis

Data Importation in CIVA DS

The screenshot displays the CIVA DS software interface. On the left, a table lists various components with columns for ID, Ch., Nom, Nb. E., Nb. E., Nb. T., and Taille. Below this is a section for 'Entrees Extractions' with a table of ID, Exp., Tag, Forme, and Donnees. The main area shows a 3D visualization of a complex structure with a grid of points and lines. On the right, there are settings for 'Extraction' and 'Labels', including a table for 'Labels' with columns for Tag, Origine, Courant, and Visible.



Consolidation and preparation of training data

Fusion of Upstream and Downstream channel (using Fusion module)

Normalization of the data [0;1] through "Output Formula Tool"
In this case, we decided to normalize the data in order to bypass the calibration steps between simulation and experiment, and to eliminate the dependency on the defect size (which significantly impacts the amplitude) in a classification context

Selection and labeling (name of the class) for each dataset

ID	Use	Color	Name	Nb Sample	Size	Class
1	<input checked="" type="checkbox"/>	Light Blue	CHAMFER_DSUS	300	2.634 Mo	C
2	<input checked="" type="checkbox"/>	Blue	CHAMFER_USDS	300	2.634 Mo	C
3	<input checked="" type="checkbox"/>	Light Green	ROOTGEOM_DSUS	100	898.438 ko	R
4	<input checked="" type="checkbox"/>	Green	ROOTPLAN_DSUS	300	2.634 Mo	R

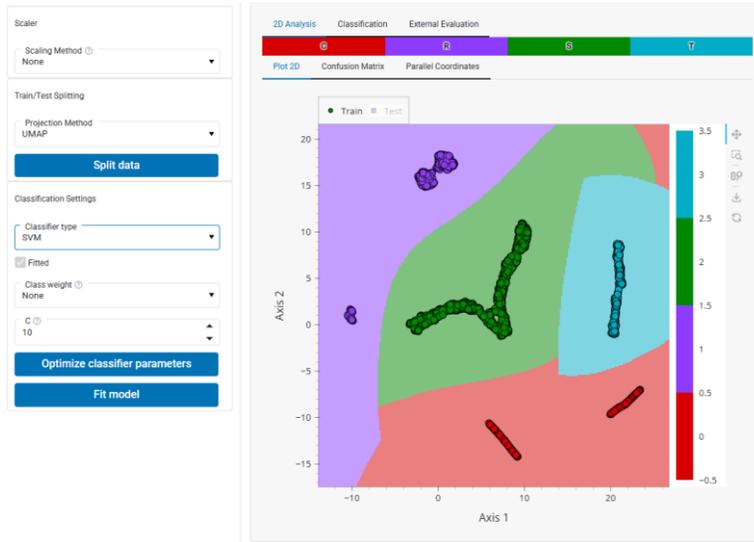
Training datasets
(Simulated data)

ID	use	Color	Name	Nb Sa...	Size	column Class...
4	<input checked="" type="checkbox"/>	Orange	EXP_C_244_255	12	107.531 ko	C
5	<input checked="" type="checkbox"/>	Orange	EXP_C_268_277	10	89.609 ko	C
6	<input checked="" type="checkbox"/>	Orange	EXP_S_310_316	7	62.727 ko	S
7	<input checked="" type="checkbox"/>	Orange	EXP_RG_337_341	5	44.805 ko	R
8	<input checked="" type="checkbox"/>	Orange	EXP_RG_344_350	7	62.727 ko	R

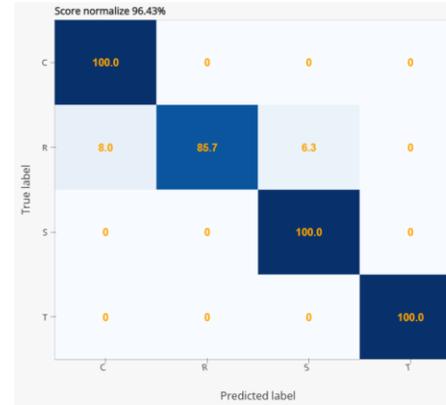
Test datasets
(here experimental data)

AI and simulation-based diagnosis

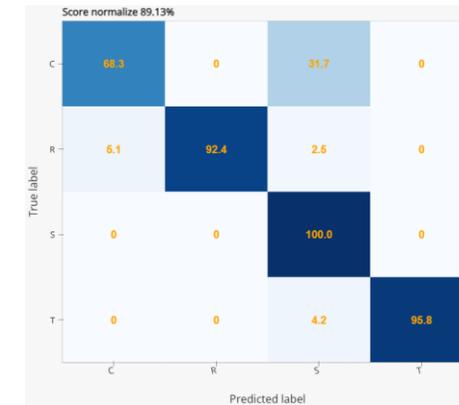
Previsualization (2D Projection)



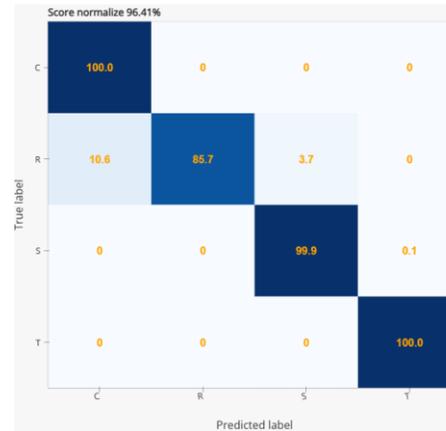
Classification and evaluation



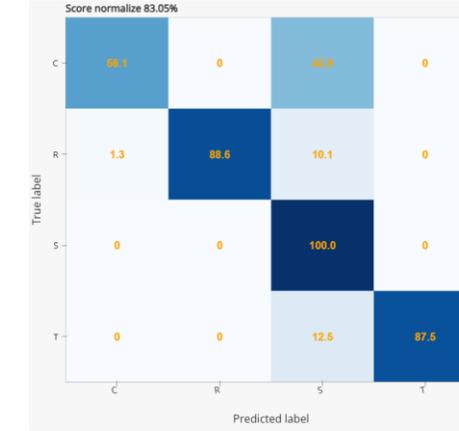
SVM CROSS-VALIDATION



SVM on TEST BASE (EXP.)



MLP CROSS-VALIDATION



MLP on TEST BASE (EXP.)

Search for the best parameters and algorithmic choices

This first step allows to check the separation capabilities of the data
Evaluation is applied by cross validation

Prediction assessment of trained models using cross-validation techniques

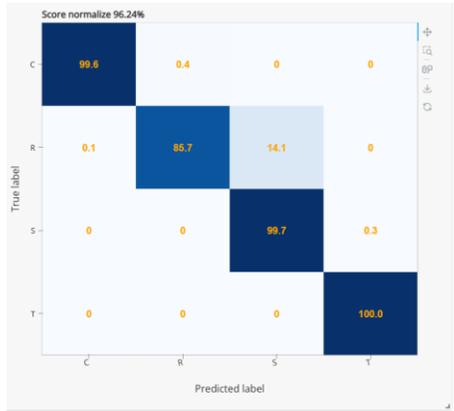
Prediction assessment of trained models using a dedicated test base

AI and simulation-based diagnosis

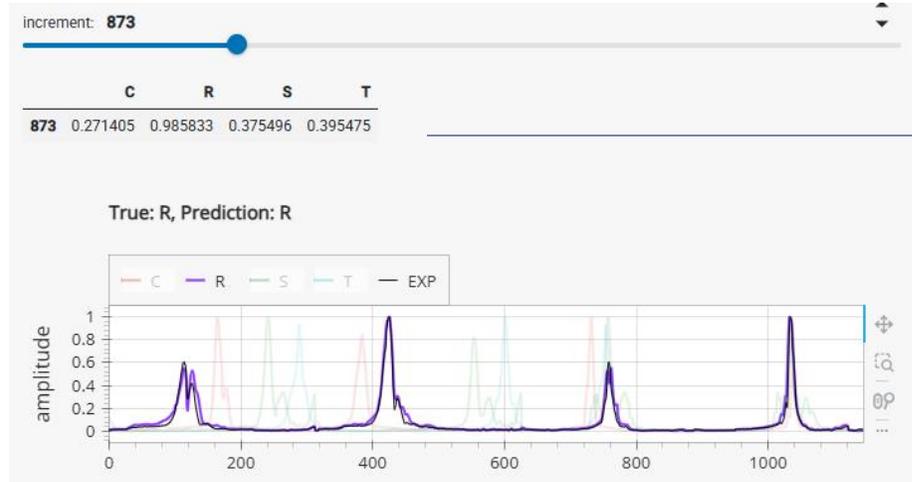


Focus on Time Series Correlation (TSC) (CIVA 2025)

Nearest neighbor search using a similarity measure based on signal cross-correlation.

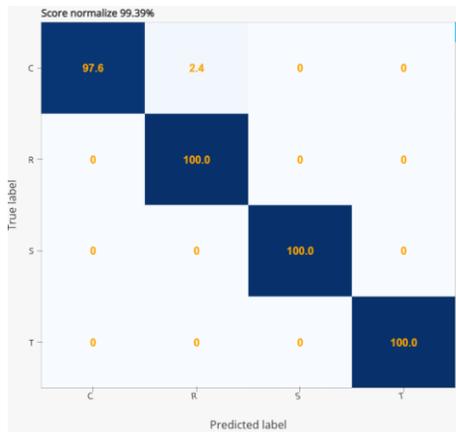


TSC CROSS-VALIDATION

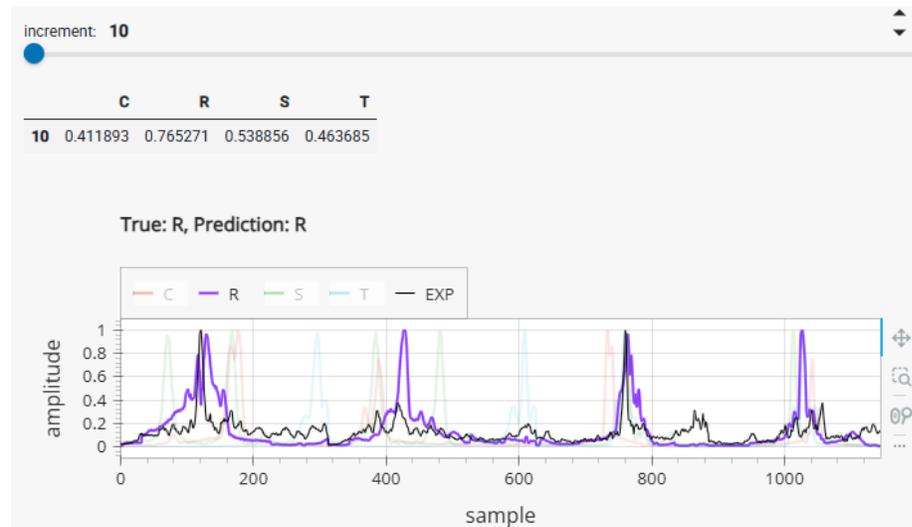


SCORE PER CLASS FOR EACH PREDICTION

TSC CROSS-VALIDATION – SUPERPOSITION DATA TO EVALUATE VS CLOSEST TRAINING DATA FOR EACH CLASS



TSC on TEST BASE (EXP.)



TSC on TEST BASE (EXP.) – SUPERPOSITION DATA TO EVALUATE VS CLOSEST TRAINING DATA FOR EACH CLASS



Augmented Simulation for ML

Objective : most realistic simulation results

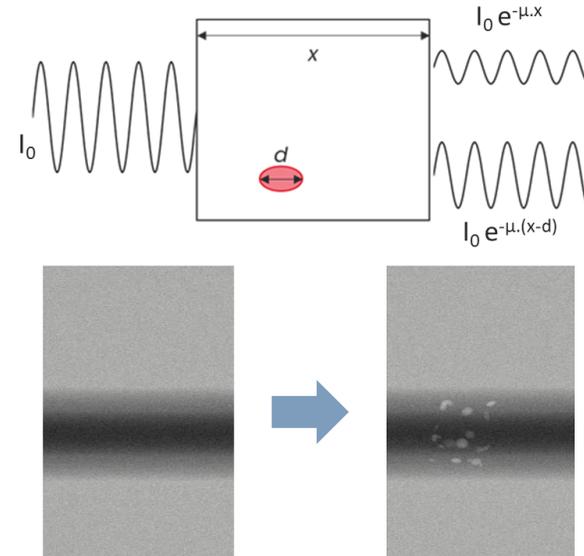
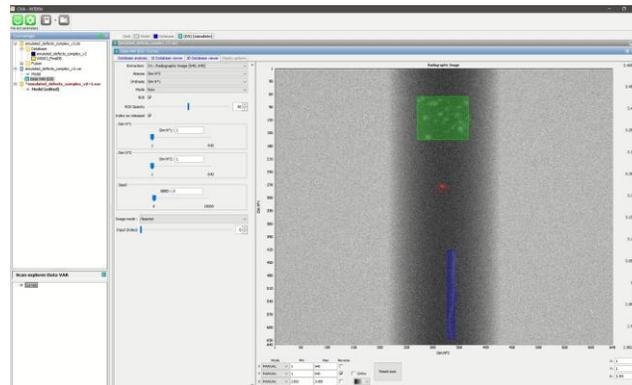
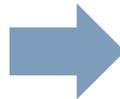
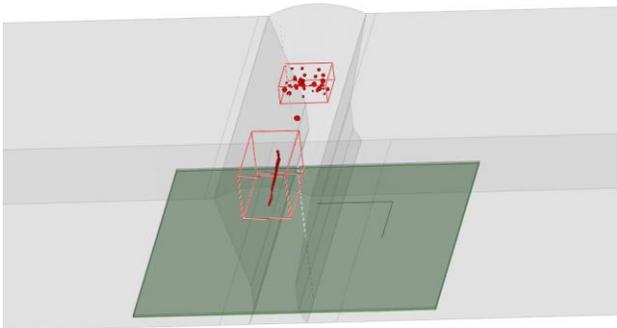
RT Use Case

- Experimental image without defect $I_{\text{exp}} = I_0 \cdot e^{-\mu \cdot x} + B$
- **Recalibration** : $B \rightarrow$ max range of thicknesses

$$I_{\text{aug}} = (I_{\text{exp}} - B) \cdot e^{\mu \cdot d} + B$$



Parametric Study for data base generation



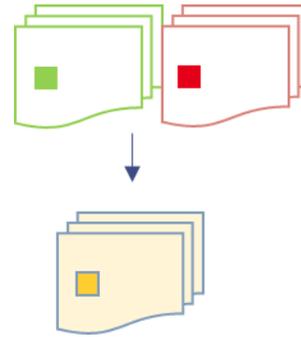
Augmented Simulation for ML

CIVA DS Fusion :

Parametric study + set of experimental images



For each simulated image, random drawing of an experimental image and fusion

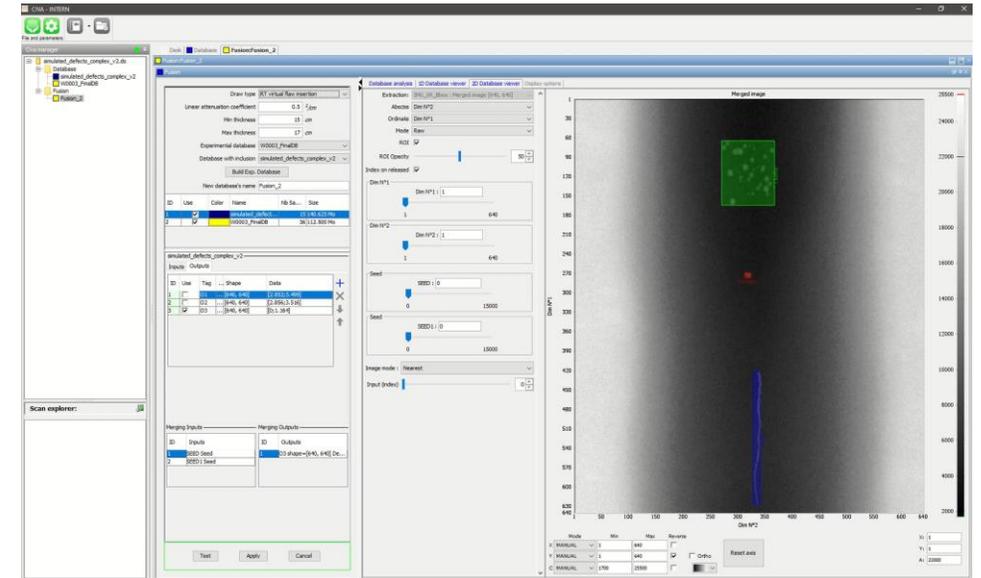


Realistic data base with annotations
(region of the defect, class name)



Database Ready to be used for Machine Learning (Object Detection on Image)

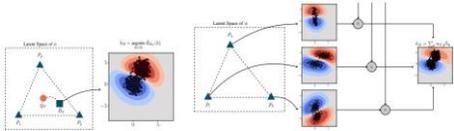
Possibility to adapt the format in Python Notebook thanks to CIVA Script

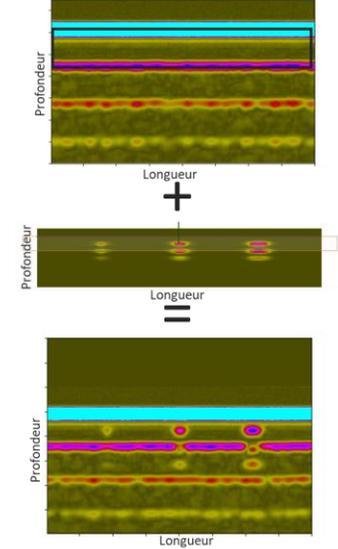


Outline

- . Introduction to CIVA DS
- . Key Features of CIVA DS
- . Use Case Examples
- . **Roadmap and Outlook**

Roadmap

- Inversion and Optimization Modules with AI Integration
- Augmented Simulation will be extended to other techniques (UT, ET, TT...)
- AI domain adaptation 
- *New algorithms based on Explainable AI (XAI) and Trusted AI* 
- Bridge to Convolutional Neural Networks in Deep Learning
- Export of AI models for
 - CIVA Analysis Plug-in
 - External use (including real-time embedded integration)



UT Augmented Simulation



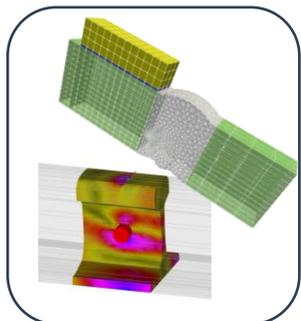
Embedded AI



CIVA USERS
COMMUNITY Suggestions are welcome !

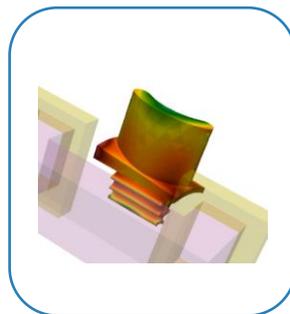
CIVA Strategic Roadmap

MULTI-MODEL



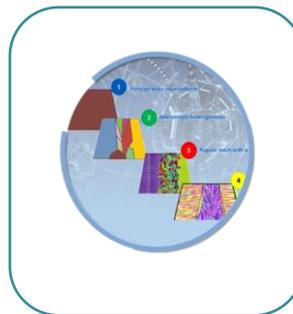
- Semi-Analytical
- Digital
- Hybrid
- Materials

MULTI-TECHNIQUE



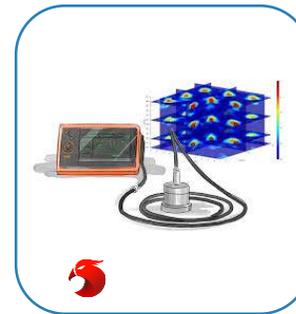
- Thermography
- UT Laser
- Additive Manufacturing
- Trusted and Reliable Simulation

MULTI-SCALE



- Materials
- Simulation for Analysis
- Cloud Computing

DATA AND PROCESSING



- Data Science
- Trusted AI (ExpressIf)
- Augmented Imaging
- Reconstruction

DIGITAL TWIN



- Digital Twin (DT)
- Real-time Simulation
- Planning (ExpressIF)

2020

2022

2024

2026

2028

2030

CIVA 2021



Simulation for SHM (numerical model)



Hybrid models (SA/EF)



Metamodels



Robotic control for Nozzle inspections

CIVA 2023



Thermography



Numerical calculation for ET



Demonstration of performance



Data Science

CIVA 2025



FE Computation



RT complete model
Fluorescence
Bremsstrahlung



Augmented Simulation



Data Science

CIVA 2027



UT : Complex structures and materials



TT and ET: on 3D CAD Specimens



Characterization of materials



Trusted Simulation

CIVA 2030



CIVA Robotics and DT



Real-time simulation
And AI diagnostics



Additive manufacturing



UT laser