Meet DebiAI: A Versatile Open-Source Tool for Streamlined Data Analysis, Visualization, and ML Model Evaluation

Tom Mansion
IRT SystemX
91120 Palaiseau, France
tom.mansion@irt-systemx.fr

Raphaël Braud
IRT SystemX
91120 Palaiseau, France
raphael.braud@irt-systemx.fr

Faouzi Adjed
IRT SystemX
91120 Palaiseau, France
faouzi.adjed@irt-systemx.fr

Ahmed Amrani
IRT SystemX
91120 Palaiseau, France
ahmed.amrani@irt-systemx.fr

Sabrina Chaouche
IRT SystemX
91120 Palaiseau, France
sabrina.chaouche@irt-systemx.fr

Fady Bekkar
IRT SystemX
91120 Palaiseau, France
fady.bekkar@irt-systemx.fr

Yoann Randon
IRT SystemX
91120 Palaiseau, France
yoann.randon@irt-systemx.fr

Martin Gonzalez
IRT SystemX
91120 Palaiseau, France
martin.gonzalez@irt-systemx.fr

Loïc Cantat
IRT SystemX
91120 Palaiseau, France
loic.cantat@irt-systemx.fr

***Abstract*—We present DebiAI, a powerful open source tool crafted to streamline data analysis, visualization, and the comprehensive evaluation and comparison of Machine Learning models. It serves as a versatile companion throughout the entire machine learning workflow, from project data preparation to model performance assessment. With its intuitive and feature-rich graphical interface, DebiAI enables users to effortlessly visualize, explore, select, edit, and annotate both data and metadata. The tool is also equipped for bias detection and contextual evaluation of ML models, ensuring a thorough and fair analysis. Built on a flexible, generic data model, DebiAI is adaptable to a wide range of ML tasks, including classification, regression, and object detection in images, as well as a variety of tasks for time-series and more. Released under the Apache License, Version 2.0, it offers an accessible and linearly scalable solution for ML practitioners of all levels. The code for the proposed tool is publicly available at https://github.com/debiai; and other information and user guidelines are available on the dedicated website: https://debiai.irt-systemx.fr.**

***Index Terms*—Data Analysis; Data Visualization; Bias Detection; Human-Centered Machine Learning; Trustworthy AI.**

I. Introduction

This work is an extension of our previous work published in the ICAS conference [1] dealing with data analysis and visualization in Machine Learning (ML) projects. They are playing a crucial role in a typical ML process, and they are not only contributing in the data preparation phase, but also during and after the model building. In short, data visualization contributes to the whole ML life cycle, from its specifications and data acquisitions until its deployment and monitoring. This involved to create an emerging research topic, which combines several interactive systems and domains for ML processes, focused on human interaction and collaboration [2] to constitute a new field that is named Human-Centered Machine Learning (HCML) or Human-Centered Artificial Intelligence (HCAI) interaction [3], [4]. Thus, a typical HCML framework allows an interactive visual analysis and evaluation of data and ML models [5]. As a result, efficient tools are essential to support users in the most user-friendly manner throughout the entire ML process, including tasks like data preparation and quality inspection prior to training, as well as monitoring model performance and deployment quality after training. Specifically, as highlighted by Caple et al. [4], the HCML/HCAI process encompasses five key areas: (i) Explainable and Interpretable Artificial Intelligence (AI), (ii) Human-Centered Approaches to AI Design and Evaluation, (iii) Human-AI Collaboration, (iv) Ethical AI, and (v) AI Interaction.

An effective tool should support the iterative ML process across multiple stages: from data preparation, analysis, anomaly detection, and annotation, to the evaluation and analysis of model results and performance. This helps identify model weaknesses and uncover issues at the data level. In real-world ML projects, such as those in industry, data is often enriched with metadata, including operational context and expert knowledge, which provides deeper insights into the raw data and enhances the learning process. This, in turn, improves the quality of model training and predictions. Furthermore, having such tools increases the trustworthiness of the ML algorithms in use.

In ML-based engineering systems, it is crucial to guarantee key properties like accuracy, robustness, explainability,

Research funded by the French Government under the “France 2030” program.

fairness, privacy, among many other primary values of AI trustworthiness. Current research and development challenges of deploying trustworthy ML solutions are covered by wide programs such as Confiance.ai [6], the French AI flagship program to industrialize trustworthy AI-based critical systems [7], [8] and the TAILOR [9] network at the European level.

DebiAI has been developed by the IRT SystemX in the framework of Confiance.ai program to contribute in ensuring trustworthiness by data, and serves as the main interface to view, analyze, select, edit and/or annotate any type of data and metadata.

The remainder of the paper is structured as follows. Section II provides a brief review of the state-of-the-art HCML tools. Section III outlines the methodology developed in this work, building on our previous contribution [9]. Section IV details the implementation, presenting the main architecture and various functionalities. In Section V, we describe the application of the tool to real-world use cases, including images and time series. Section VI discusses evaluation analysis and usage recommendations, while Section VII addresses limitations. Finally, conclusions and future perspectives are presented in Section VIII.

II. Literature review

Data visualization is the practice of representing information using graphical representations, employing technology-driven tools and software. Its fundamental objective is to enhance pattern recognition, improve understanding of complex concepts and facilitate in-depth exploration, thereby fostering the generation of new insights. Well-designed data visualizations can help in understanding large datasets and establishing connections between ideas, concepts, and processing stages. Therefore, visual analysis can contribute to the optimization of AI approaches by actively participating in all aspects of the model building process [10], [11]. Similarly, Hohman et al. [12] highlight that successful ML applications often require iterations in data handling and continuous adjustments of the model. The authors introduced CHAMELEON, an interactive tool designed to attribute data iteration, thereby enhancing model performance, data validation, and the overall quality of ML projects. To facilitate the interaction between machine learning experts and final users, Françoise et al. [13] proposed a toolkit addressing an interactive machine learning workflow to permit a collaboration between machine learning experts, designers and end-users through a unified tool. To improve data quality, Kandel et al. [14] presented Profiler, a tool using data mining to automatically detect issues and recommend coordinated visualizations for context-based assessment. Profiler offers methods for integrated statistical and visual analysis and view suggestions.

Grafana [15] is an open source web platform, in its recent version, used for data visualization and tracking in real time for data science field. This tool has multiple functionalities for data monitoring, whether user's data is connected to Prometheus [16], InfluxDB [17] and others database. Thus, the platform offers an increasing array of data analysis and generative AI features, such as creating alerts, predicting capacity needs, and detecting anomalous activities [18]–[20]. Streamlit [21] is a Python library that enables data scientists to swiftly and effectively create robust web applications. The advantage of Streamlit lies in its simplicity of using python scripts to build customizable dashboards, whereas DebiAI has the advantage of offering a wide range of ready to use widgets available.

ScrutinAI [22] is a Visual Analytics tool specifically tailored for enhancing the comprehension of deep neural network (DNN) predictions. Its primary objective is to identify and investigate potential weaknesses within models. To facilitate this, ScrutinAI provides interactive visualizations of input and output data, along with interactive plots and data filtering for comprehensive analysis of predictions. This tool is specifically designed for object detection and semantic segmentation, whereas DebiAI is applicable to a wide range of use cases. Zhang et al. [23] presented Manifold, a visual analytics platform designed for comparing and debugging ML models. The platform empowers users to categorize instances based on the model's accuracy and confidence, identify symptomatic instances that generate incorrect results and continually help to enhance the model's performance. As DebiAI, Manifold is created as a generic tool that operates independently of the internal logic of the ML model. It focuses on the input and the output. Similarly, Uni-Evaluator [24] is also independent of the model but focuses on evaluating computer vision tasks, such as classification, object detection, and instance segmentation, with tailored visualizations for these specific uses cases. However, DebiAI can perform a more general evaluation across different types of ML tasks thanks to its flexible data model.

To improve model performance and understand their limitations, it is not enough to just rely on the overall results from the test and training sets. To overcome this limitation, the ModelTracker [25] tool provides instance-level result visualization, allowing users to inspect each instance individually. The tool has been applied to the binary classification task, and in [26], the authors introduced Squares, which extends the approach to multi-class classification. Additionally, Squares facilitates the estimation of common performance metrics and provides instance-level result visualization, guiding practitioners in troubleshooting performance issues while offering direct access to relevant data. They used Parallel Coordinates Plots (PCP) to visually represent the multi-class predictions for a subset of instances. In line with this approach and to improve the understanding of models results, the proposed DebiAI extends the analysis by enabling the exploration of model outcomes at various levels of granularity, including instance, subset, and dataset. This functionality has been applied to multiple tasks such as regression, classification, object detection, etc. In the same way as outlined in [26], DebiAI utilizes PCP to analyze model results. The implementation of PCP within DebiAI is flexible, enabling its use not only for result analysis but also for assessing attributes. Table I summarizes an overview of HCML tools presented in this section by

TABLE I: Summary of Tools in AI and Data Visualization

Tool	Purpose	Key Features
CHAMELEON [12]	Enhances ML model performance and data quality through iteration	Interactive tool for data attribution, model performance enhancement, and data validation
Marcelle [13]	Facilitates interaction between ML experts, designers, and end-users	Unified toolkit for interactive machine learning workflows
Profiler [14]	Improves data quality using data mining and visualizations	Integrated statistical and visual analysis, view suggestions for context-based assessment
Grafana [15]	Real-time data visualization and tracking for data science	Supports multiple databases (e.g., Prometheus, InfluxDB), alerting, predictive analysis, and anomaly detection
Streamlit [21]	Simplifies creation of data-driven web applications	Python-based, customizable dashboards, fast prototyping
ScrutinAI [22]	Enhances understanding of DNN predictions	Interactive visualizations, data filtering, tailored for object detection and semantic segmentation
Manifold [23]	Compares and debugs ML models	Instance categorization based on accuracy/confidence, generic tool for input-output analysis
Uni-Evaluator [24]	Evaluates computer vision tasks	Tailored visualizations for classification, object detection, and instance segmentation
ModelTracker [25]	Provides instance-level result visualization for binary classification	Instance-level inspection and analysis
Squares [26]	Extends ModelTracker to multi-class classification	Instance-level result visualization using PCP
DebiAI	General evaluation across different ML tasks	Interactive instance, subset, and dataset-level analysis, flexible PCP implementation for results and attributes analysis

highlighting their purposes and their main key features.

III. Methodology

DebiAI is a web-based visual analytics application designed to support ML and data analysis. Its emphasis lies in two crucial phases of the ML pipeline: pre-model and post-model building. As shown in Fig. 1, DebiAI facilitates the development of ML models by assisting in data analysis during data curating and processing stage and enabling models performances comparison.

In the pre-model construction phase, DebiAI serves as a key resource for data scientists and ML engineers during project preparation. It enables them to visually identify biases and errors in data inputs, detect anomalies and outliers throughout the data life cycle, assess data quality and domain coverage through relevant metrics and select and analyze subsets of data to improve the quality of ML models.

In the post-model building phase, DebiAI serves as a visual analytics solution, simplifying the interpretation of the ML model's outputs. Its primary objective is to present the model's results in an intuitive and easily understandable manner, ultimately enhancing user confidence in the model's predictions. Additionally, DebiAI offers features to identify model's weaknesses, comparing performances, and evaluating model's effectiveness according to the project contextual data such as weather or gender biases. This comprehensive approach fosters ongoing model refinement, tailored to the specific needs of the use case.

In both phases, DebiAI provides tools for creating and sharing statistical visualizations of the project data and results with collaborators (team or/and clients). In summary, DebAI reflects its name from "Debiasing AI" towards mitigate AI bias [27].

A. Functional Description

DebiAI is an intuitive visualization tool designed to simplify the creation of interactive dashboards, empowering users without little to no programming skills. It offers a diverse set of

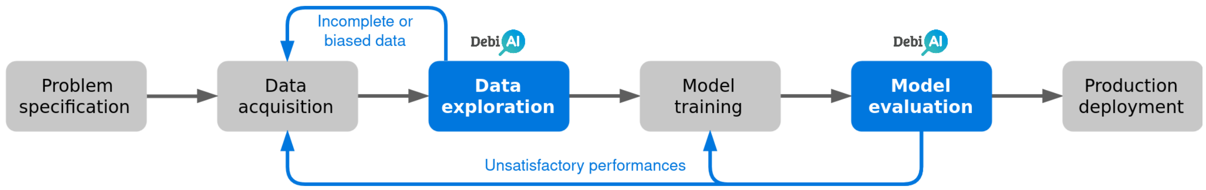


Fig. 1: The role of DebiAI within the different stages of an AI project. This diagram illustrates a comprehensive machine learning pipeline from problem specification to production deployment. DebiAI intervenes during the data exploration stage and the contextual comparison stage of AI project models. It helps provide intelligence on incomplete or biased data and on model performance, thereby accelerating the resolution of feedback loops.

graphical widgets, including charts, tables, parallel categories, parallel coordinate plots, interval plots, night star plots, and sample arrays. Moreover, DebiAI provides a user-friendly and flexible solution for interactive dashboard design, allowing users to effortlessly configure, adjust, resize, and position these widgets within their dashboards, ensuring the utmost customization of data presentation. This includes the ability to generate and share statistical visualizations of project data with team members or clients, fostering collaboration and informed decision making by providing clear insights into the data.

One of DebiAI's standout features is its dynamic data selection and filtering capabilities, which encourages continuous exploration. Users can effortlessly create data subsets (selections) and apply filters based on various variables and contexts. This ensures that the dashboard consistently presents the identified subset of data. Furthermore, DebiAI assists users in identifying biases and inaccuracies in inputs, results, project data contexts, or ground truths, thereby improving data integrity.

DebiAI facilitates the evaluation and comparison of ML model's performances within the whole dataset or a specific data subset. It enables analysis of results across multiple levels of granularity. Indeed, the model's performances are calculated at the level of each instance. Consequently, it is possible to identify the contexts or a combination of contexts in which the model encounters difficulties. It also simplifies the generation and organization of datasets, supporting in-depth analysis and potential retraining.

DebiAI relies on a generic data model that facilitates seamless application across various datasets, data types, and use cases while maintaining consistent data processing practices. This essential feature provides DebiAI with flexibility, allowing it to transition between various datasets or the results of the model. In addition to its visualization capabilities, DebiAI incorporates implementations of statistical measures such as correlation analysis using Pearson or Spearman coefficients. To support these visualizations, DebiAI also integrates techniques for discretizing continuous variables. In addition, it enables the use of internal or external algorithms to compute metrics or indicators on the data. Consequently, these metrics can be calculated either before integrating the data into DebiAI or during the data analysis phase. Various types of calculations can potentially be carried out by these algorithms, including the computation of new features, the assessment of model's results quality, as well as indicators of data quality and distribution. In addition to that, at each step, DebiAI provides an easy solution to transform the dashboard of widgets and comments into a markdown file with a PNG image format for each widget.

IV. Implementation Description

In this section, we describe DebiAI's global architecture and dive into the details of each component. We start by an overview of the architecture details, then we proceed by presenting the data model, generic and multi-dimensional, followed by data integration process.

A. *DebiAI Technical architecture*

The DebiAI architecture is divided into two main environments as shown in Fig. 2, which are the project environment (data and algorithms) and the application environment (backend and dashboard).

The project environment consists on the following:

- **Project Data:** This is the source of data that the user intends to analyze. It may originate from various sources and formats, such as CSV or JSON.
- **Data-Providers:** These are the services created by the project members to enable DebiAI to fetch data and model results directly from the project's sources. Creating a Data-provider allows DebiAI to always fetch up-to-date data without duplication. A Web Data-Provider can be developed using any programming language, access data from any type of database, and be hosted on any server. The only stipulation is that it should implement and expose a specific REST API according to a defined contract. DebiAI allows users to add as many Data-Providers as they require, allowing them to analyze different projects with mixed data sources.
- **Python Scripts and DebiAI Python Module:** Using the DebiAI Python module, users can adapt their existing scripts and workflows to create selections and insert data and model's results into DebiAI. This Python module is a simpler alternative to creating a Data-provider, but it requires data to be duplicated and the module to be called at each project data update, which is time consuming.

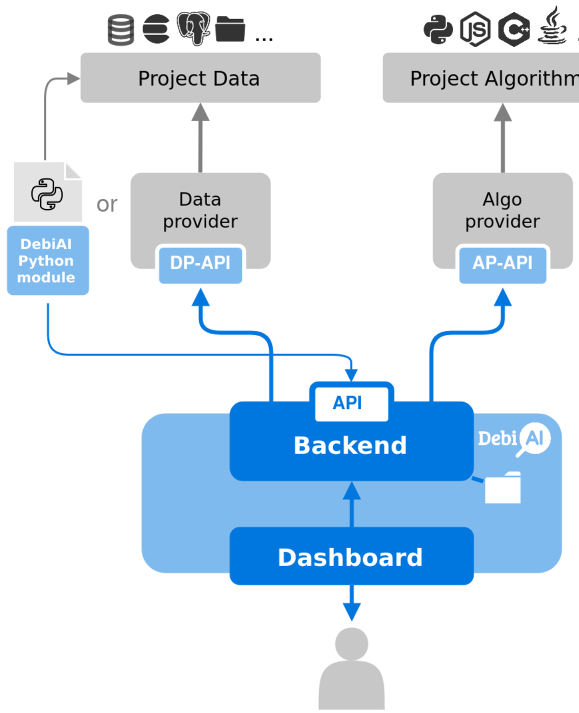


Fig. 2: DebiAI system architecture overview illustrating the division between the project environment, where users manage their data sources, data providers, algorithms, algorithm providers, and the DebiAI application environment, which includes the backend, web dashboard, and data storage. The DebiAI Python module serves as a simple way to insert data into DebiAI. This architecture enables seamless integration of multiple and mixed data sources, real-time data updates for project analysis, and project algorithms.

The compoarison with Data-provider services is done in Section VI.

- **Algo-Providers:** These services are used to provide specific algorithms required by projects. Once an algorithm is provided to DebiAI, it can be called from the analysis dashboard with the project's data. For example, an algorithm can be used to compute some specific features, model prediction, data quality metrics, etc. The algorithm's results can be displayed, filtered, and analyzed, just like any other dataset. An Algo-Provider can be developed using any programming language, expose any algorithm and be hosted on any server. The only stipulation is that it should implement and expose a specific REST API according to a defined contract. Users can add as many Algo-Providers as they require.

The DebiAI's application environment consist on the following:

- **Backend and API:** This is a Python-powered backend that provides an API and serves the Web dashboard. This API is employed by the dashboard for data retrieval and by the Python module for data insertion. Additionally, it manages communications with the Web Data-providers, processes computational requests made by the dashboard, and calls the Algo-Providers selected from the dashboard.
- **DebiAI Web Dashboard:** This is the user interface of DebiAI, developed using VueJs. It provides users with an interactive platform to manage and view their data, and is hosted and served by the DebiAI backend. DebiAI uses different tools to display plots, the main being the PlotlyJs library.
- **Data storage:** DebiAI uses a folder-based data store that contains data in a JSON format. This data store supports the DebiAI backend by retaining projects created by the Python module and some specific dashboard elements, including layout configurations for project dashboards.

B. DebiAI Generic Data Model

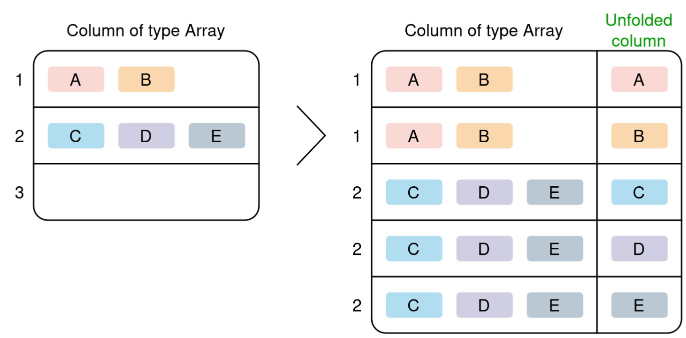


Fig. 3: Vertical Unfolding : DebiAI allows array columns to be **unfolded vertically**, on the condition that the data follows certain formatting conditions. This figure exposes how unfolding an array column vertically will add more lines. Note that because the array for data number 3 is empty, the value for data number 3 is absent from the unfolded column.

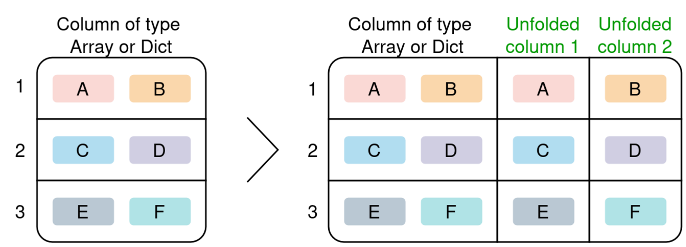


Fig. 4: Horizontal Unfolding: DebiAI allows array and dictionaries columns to be **unfolded horizontally**, on the condition that the data follows certain formatting conditions. This figure exposes how unfolding a dictionary or array column horizontally will add new columns.

One of the most important features of DebiAI is its data model. The main objective is to enable the determination of the format of instances and the relationship between instances, models, models' outputs, and models' evaluation metrics per instance. Syntactically, each instance is composed of attributes, contexts, and annotations. The instance is linked to multiple ML models, where each model produces an output. Evaluation metrics are also associated with the model's outputs.

1) **Data purpose:** This structure is applicable to all types of data and ML tasks such as classification, regression, object detection, anomaly detection, previsions.
2) **Data format:** The data format required must follow a CSV format and/or table structures. Dataframe and NumPy arrays are supported by the DebiAI data insertion Python module. When supplying data through a data-provider (described in Fig. 2), any format is acceptable since it follows the requirements of the data-provider's API.
3) **Primitive data type:** DebiAI supports primitive types such as text, numbers and boolean values. Any data columns with text will be considered as a class column. Class columns are treated differently in DebiAI, for example, the distribution plot will set the bins number to the number of unique values in the class columns, but set a fixed bin number for number columns. Columns with only numerical data can be forced into a class type, this can be useful when numeric values needs to be considered as a class, such as vehicle model number, a year date or age.
4) **Missing data:** Missing, None or NaN data are supported by DebiAI since version 0.29.0. The percentage of missing values is displayed for each column, and missing values can be filtered out or in.
5) **Arrays and lists data:** DebiAI supports columns of values containing lists under the following condition: the column must contain only lists. If the condition is met, DebiAI will be able to unfold the list of values as new lines, changing the scope of the analysis (as explained in Fig. 3). This process is called "vertical unfolding", it is demonstrated in Section IV-C. If all the columns list values have the same number of keys, the horizontal unfolding is available. Unfolding the lists horizontally will treat the list values as individual columns (as explained in Fig. 4). If the list recursively contains more lists, the unfolding operation can be repeated. However, columns containing some dictionaries and some other mixed types won't be able to be unfolded.
6) **Dictionaries and JSON objects data:** DebiAI supports columns values containing dictionaries (a data element composed of values associated with keys) under certain conditions: the column must contain only dictionaries, the dictionaries must have the same keys for all values, the number of keys must not exceed 30. If these conditions are met, DebiAI will be able to unfold the dictionaries values and treat them as individual columns (as explained in Fig. 4). If the dictionaries recursively contain more dictionaries, the unfolding operation can be repeated. However, as vertical unfolding, columns containing other types than dictionaries or dictionaries with different keys won't be able to be unfolded.

C. Multi-Dimensional Data Model in DebiAI

DebiAI's data model is equipped to handle complex, multidimensional datasets, making it particularly valuable in projects like the Woodscape dataset [28] where both images and the objects are connected (as illustrated in Fig. 5): within them are analyzed. The model supports recursive and nested objects, such as lists and dictionaries, allowing users to progressively explore and **unfold** various data dimensions.
In projects like Woodscape, the dataset initially comprises 1,624 images. By unfolding these images into their annotated objects, the analysis scope broadens, covering 72,061 objects.

Fig. 5: Hierarchical structure of the Woodscape dataset: Layer1 represents images, while Layer2 represents annotated objects within those images.

This feature enables a detailed analysis of specific aspects, such as the distribution of object classes (e.g., vehicles, pedestrians) and position of the objects across the dataset. Such a capability is essential for projects with hierarchically structured data, facilitating a comprehensive analysis at multiple levels. In this multidimensional model, users can start with a high-level analysis of the entire dataset, then delve deeper into specific annotations, like object detection labels. This process provides valuable insights into the distribution and characteristics of different object classes, helping to refine model training and evaluation strategies.

Furthermore, DebiAI enhances its selection capabilities by ensuring that any selection made on the unfolded data (such as the Woodscape objects) will also select the corresponding original data (such as the Woodscape images as illustrated in Fig. 7). This feature greatly improves the accuracy and relevance of data selection, allowing users to maintain a consistent context when analyzing both high-level and detailed data aspects. This multi-layered approach enhances the depth and breadth of data analysis, allowing for the identification of biases or gaps in the dataset. DebiAI's capability to analyze data at various levels of granularity makes it an indispensable tool for developing trustworthy AI systems, ensuring a thorough and nuanced understanding of ML datasets.

D. DebiAI Data Integration Process

DebiAI offers two main ways to add data, each suited for different types of users and projects:

1) **Python Module:** This principal method enables seamless integration of project's data into DebiAI via a dedicated Python module. Made for an integration within Python workflows, this approach, for example, facilitates the direct transfer of models' results post-evaluation. This method is especially handy for those who primarily use Python.

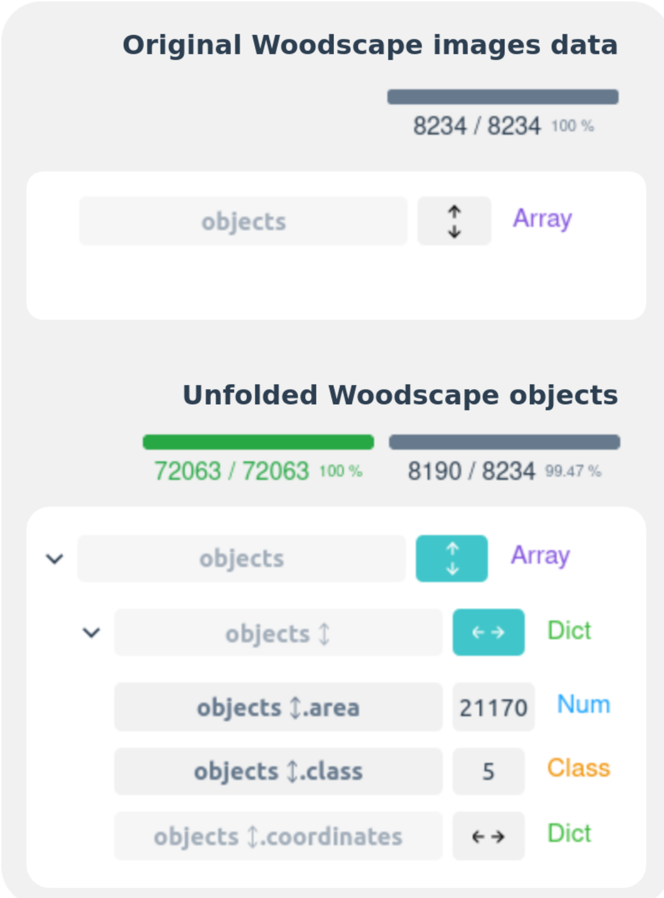


Fig. 6: DebiAI interface showing the number of images and objects in the Woodscape dataset, before and after unfolding the object layer. New columns are available, increasing the analysis depth. Note that 44 (0.53%) images aren't selected after unfolding due to the absence of objects in those images.

2) **Data Providers:** Alternatively, DebiAI can interface with data through RESTful services, termed 'Data Providers'. This method is database-agnostic, allowing DebiAI to directly request project's data, thereby making the data loading process faster and more efficient. Unlike the Python module, it doesn't require DebiAI to duplicate data within its integrated database. Although setting up a Data Provider is more time-consuming than using the Python module, it offers greater efficiency and flexibility. This is particularly beneficial for long-term projects that regularly update their data.

Each method offers distinct benefits, and the choice depends on the specific requirements and scale of the project.

V. DebiAI Application

DebiAI is built upon a generic data model, and does not depend on data type (for instance images and time series), making it pertinent to various use cases across a multitude of datasets. This intrinsic adaptability allows it to be valuable in a wide range of scenarios. It demonstrates its utility in the

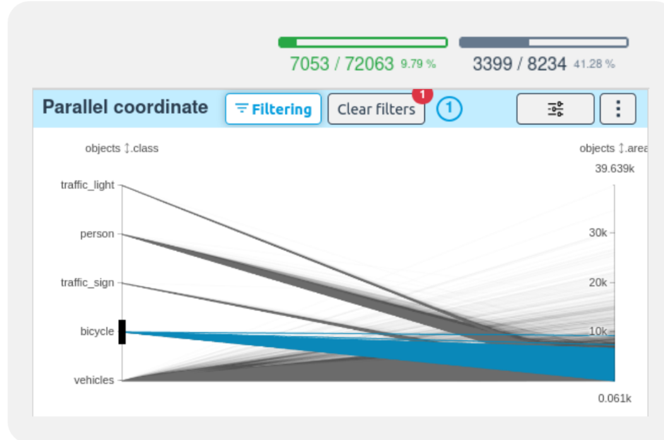


Fig. 7: DebiAI interface showing the selection of objects of type "bicycle" and the corresponding number of selected objects and images. We can see that 7053 objects (9.79%) are bicycles and that 3399 images (41.28%) have at least one bicycle. This feature enhances data analysis by maintaining context across data dimensions.

analysis of time series data, simplifying essential tasks such as regression. Furthermore, its functionality seamlessly extends to computer vision applications. Indeed, DebiAI provides tailored visual support for each stage of the process, enhancing models in tasks such as object detection and image classification.

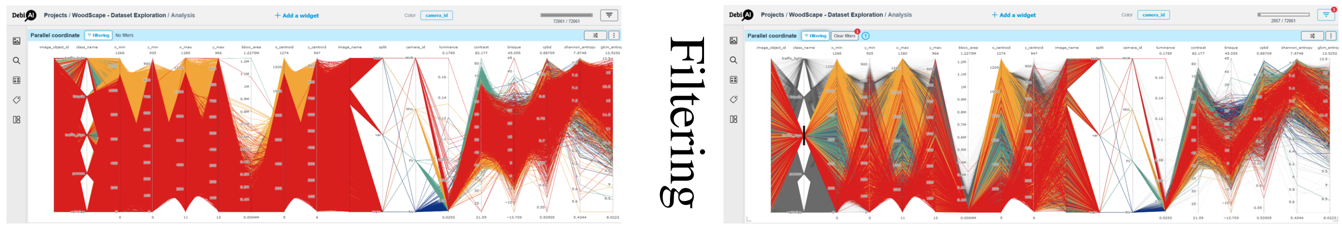
In the following two sections, we present the use of multiple widgets in DebiAI for various use cases and provide an overview of a use case related to 2D object detection.

A. DebiAI Visual Functionalities

As described in Section III-A, DebiAI gives the ability to visualize and create interactive dashboards. Moreover, it can visualize various data types such as time series, point clouds and tabular data and display computed attributes of images. However, for images viewing, it can establish links with external tools. In this section, we review a set of graphics implemented on different datasets with different data types. We also illustrate the main filtering features proposed by DebiAI. Four graphical visualizations are presented by exploring the parallel coordinates, the data distribution, the points plots and the time series widgets enhanced with interactive options. The following figures (8 to 11) described below are illustrating the main function with adapted filter. However, to be able to read and distinguish between selected filters, the dedicated website https://demo.debiai.fr/#/ offers the opportunity to reproduce the same figures.

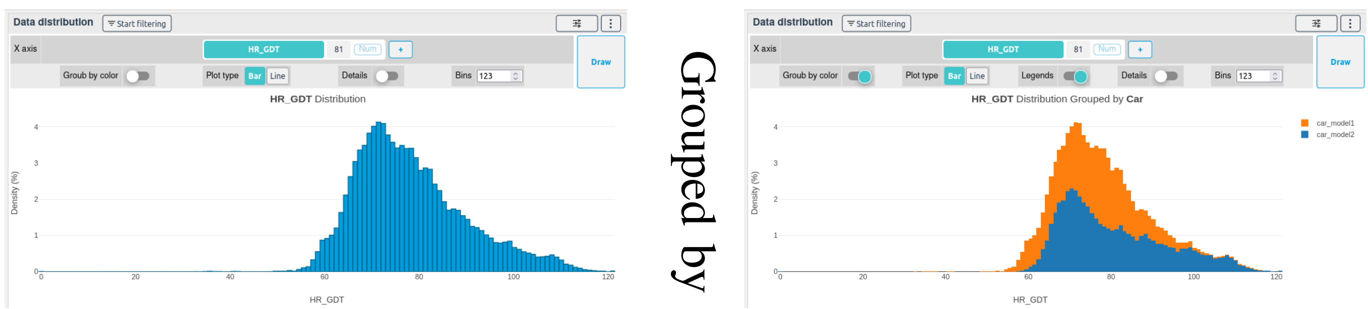
Fig. 8 illustrates a visualization of a dataset by using a parallel coordinates and the possibility to filter directly a set of variables. Another graphic visualization to analyze data distribution variables with the possibility of grouping by other variables is shown in Fig. 9. The third visualization selected from DebiAI is the possibility to apply statistical measures.

Fig. 10 captures a data cloud visualized with its primary statistical measures; an envelope of min and max of the data,



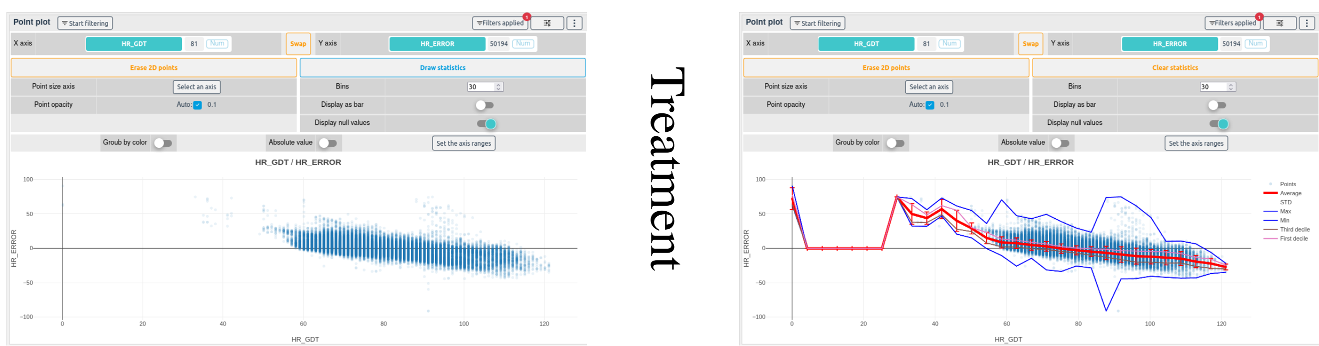
(a) original data → (b) Filtered data

Fig. 8: parallel coordinates widget. (a) represents the original data uploaded and (b) represents the same widget by selecting a subset of variables interactively.



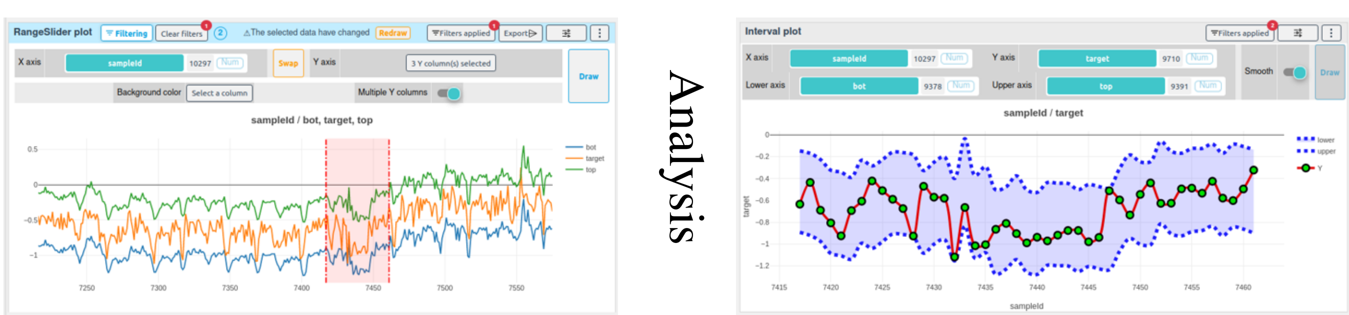
(a) data distribution of the data → (b) data distribution + "grouped by" option

Fig. 9: illustration of data distribution by adding the option of "grouped by". (a) represents an example of data and (b) represents the same data grouped by another variable with two different colors.



(a) original cloud data → (b) cloud data with statistical measurements

Fig. 10: Statistical treatment for an example of cloud data. (a) represents the original data and (b) represents the data by adding a set of statistical measures. Here illustrated measures are: mean, standard deviation, min and max and deciles



(a) original time series → (b) times series selected and analyzed

Fig. 11: Statistical analysis applied on times series example. (a) represents the original time series (b) represents the times series filtered and analyzed by adding a set of statistical measures

the average, the confidence interval $\pm\sigma$, where σ represents the standard deviation and also two deciles of the data. Fig. 11 visually encapsulates the two distinct stages in the statistical analysis of a time series. Initially, Fig. 11a displays the original data over an extended period. Subsequently, in the second stage, Fig. 11b illustrates the time series after a more detailed analysis and filtering, focusing on a shorter timeframe. Among the statistical measures, a noteworthy one to explore when analyzing dataset's variables is examining their correlation, a task effortlessly accomplished using the DebiAI's correlation matrix widget.

Fig. 12 displays the Pearson and Spearman correlation matrices for a set of a dataset's image quality metrics. We observe that contrast is positively correlated to Shannon entropy in both correlation matrices, which suggests a monotonic affine relationship between contrast and Shannon entropy attributes.

It is worth noticing that DebiAI's is part of a broader environment and can host tools such as Data Quality Metrics (DQM) library [29]. A visualization of this integration is provided in Fig. 13.

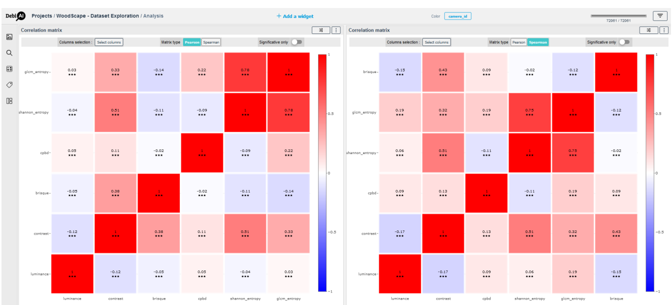


Fig. 12: DebiAI's visualization of correlation Matrices of the WoodScape Dataset under two forms: Pearson (left) and Spearman (right).

B. Images Dataset: 2D Object Detection

In order to illustrate DebiAI's functionalities on images datasets, we conducted some experiments using the WoodScape dataset [28], [30]. WoodScape is a public dataset containing more than 100k images of urban scenes captured using fish-eye cameras for automotive driving tasks from three distinct geographical locations: USA, Europe and China. The images are provided by four cameras with different angles of view (front, rear, middle right and middle left) with 360° coverage and have annotations for a diverse set of computer vision tasks. In addition to different calibration and vehicle information, The dataset provides annotations of 10k images for nine tasks; for instance, 2D object detection, semantic, instance and motion segmentation.

In the scope of this work, we focused on the 2D object detection task for five classes: vehicle, person, bicycle, traffic light and traffic sign. Our selection from the dataset was split into three chunks of $6 : 1 : 3$, namely, train, validation and test, respectively. The study process is divided into two main steps: i) data comprehension and ii) results exploration.

The first step aims to obtain a comprehensive overview of the data distribution, understand its scope and how it can be effectively used in a ML process. This comprehension is crucial, as it helps transform an industrial problem into a ML task and establish the appropriate process for models training and results validation. Fig. 14 displays the train set's final distribution grouped by cameras IDs using DebiAI's Data Distribution widget. By applying the same configuration to display the distribution of each of the three sets (train, validation and test), we observed a similar distribution among the three of them. Nevertheless, the figure highlights an immense imbalance among the distribution of the five classes, which is essential for: i) using the appropriate adaptive training techniques, for instance, a weight sampler, and ii) taking into consideration this imbalance when interpreting the models outputs to avoid biased and skewed conclusions.

In the second stage, we used DebiAI to analyze the results of our models applied on the WoodScape test set and put them back into the context of the dataset and its features. This approach ensures an accurate interpretation of the models' outputs and provides potential improvements directions. In our experiments, we used two versions of YOLO-based architectures, specifically YOLOv5 and YOLOv8. The first model, a YOLOv5 with COCO2017 [31] weights. The second model is a YOLOv5 and the third one is a YOLOv8 both trained on WoodScape dataset. Fig. 15 illustrates the relationship between the precision and the recall of each model using the *night stars plot* widget, which helps to navigate the trade-off between the two depending on the context of the task, for instance, are we prioritizing the detection quality over the quantity and vice-versa.

Fig. 16 shows the distribution of the F1-score of each model grouped by camera IDs, where we can easily spot the gap in performance between the three models: having the two models trained on WoodScape dataset showing higher scores compared to the one pre-trained on COCO2017 dataset, which is expected giving the discrepancy between the two datasets. We can also notice that the YOLOv5 trained on the WoodScape train set has better score on the data coming from the front and rear view cameras (FV and RV) of the vehicle while the YOLOv8 also trained on WoodScape shows a greater score on the middle view cameras (left and right) data. This first observations led to further investigations using DebiAI in an attempt to understand the models' outputs; you can check our tutorial on our website for more details, where a complete tutorial is given in the following link: https://debiai.irt-systemx.fr/tutorials/woodscapeTutorial/.

C. Time-series Dataset: Anomaly detection in time-series

We conduct experiments concerning the usage of DebiAI on anomaly detection on time-series applied to Server Machine Dataset (SMD) [32]. This dataset is commonly used in benchmark dataset for anomaly detection in time-series. It deals with the monitoring of the performance and health of server machines in data centers. This dataset has been collected by measuring 38 several key performance indicators (KPI) and

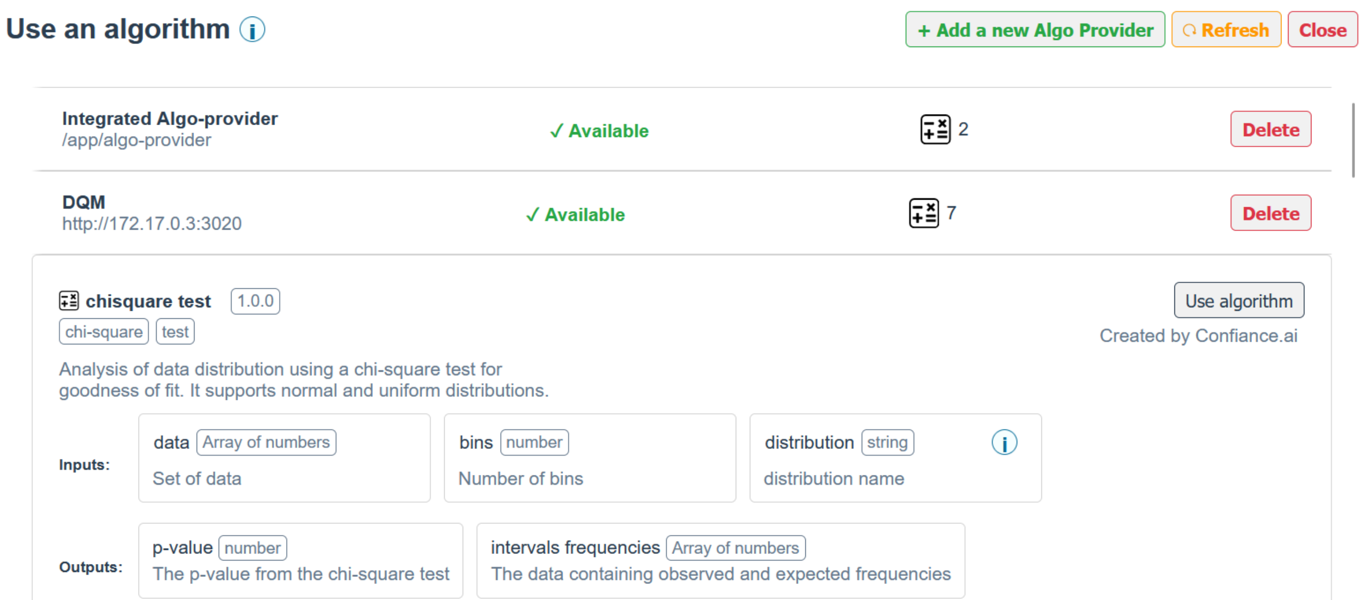


Fig. 13: Illustration of Algo-provider services in DebiAI. Integration of a new Algo-provider named DQM developed in Confiance.ai Program. The new algorithms can use directly filtered/selected dataset from the project.

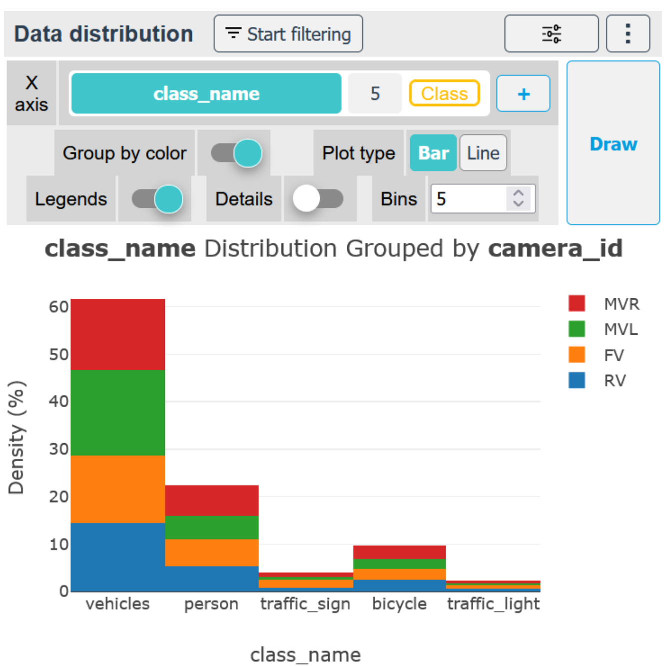


Fig. 14: DebiAI's visualization of WoodScape train's set distribution grouped by Camera ID: MVR (Middle View Right), MVL (Middle View Left), FV(Front View) and RV(Rear View)

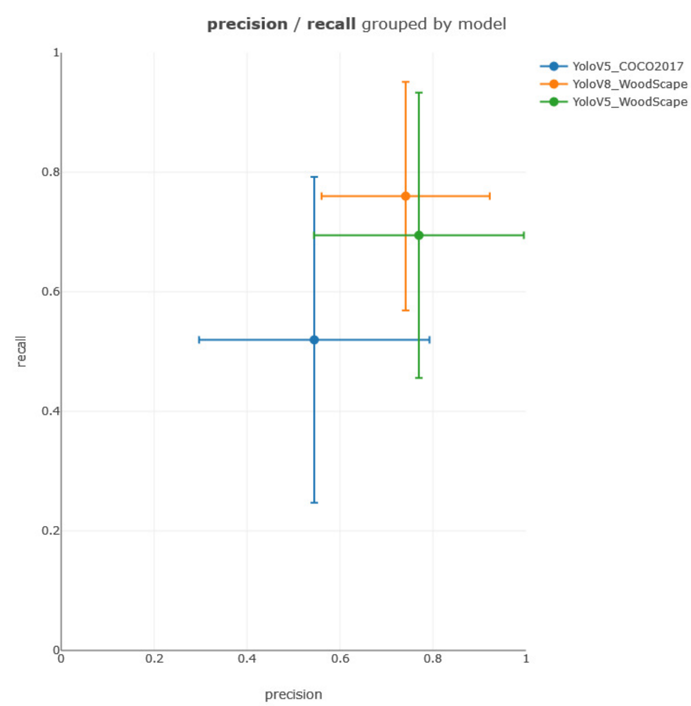


Fig. 15: DebiAI's visualisation of the Precision - Recall relationship of the three models: YOLOv5 on COCO2017, YOLOv5 on WoodScape and YOLOv8 on WoodScape.

anomalies occurrences from 28 different machines during 5 week. KPIs are indicators such as CPU usage, memory usage, system loaded.

Fig. 17 illustrates the capacity of DebiAI on producing clear and insightful SMD representations of 5 machines' first KPI. DebiAI's RangeSlider functionality renders two plots on

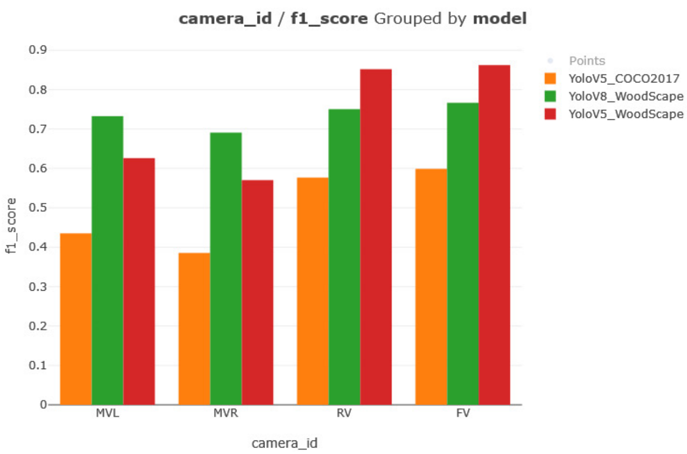


Fig. 16: DebiAI's visualisation of the F1-score results by Camera ID grouped by Models.

a desired range of timestamps, this plot allowing a detailed exploration of zoomed areas of interest while the below plot gives reference on the global time-series visualisation to better situate the user's zoomed explorations by giving them a global picture reference. Both plots present the same annotations: 5 different time-series, one for each machine's first KPI.

We trained a Topological Anomaly Detector (TAD) [33] algorithm on a time series from a single machine in the dataset. Developed by IRT SystemX and available through the TADkit (Time-series Anomaly Detection kit) library, the TAD algorithm is an unsupervised method from TDAAD (Topological Data Analysis for Anomaly Detection) a TADkit's component designed to detect anomalies in time series data. It assigns an anomaly score to each timestamp, with anomalies identified based on a predefined threshold. To evaluate the algorithm's performance, we compared its predictions against the ground truth anomaly labels in the dataset using DebiAI.

DebiAI RangeSplider plot shown in Fig. 18 visualizes the performance of our model by displaying a time series curve overlaid on a color-coded background that distinguishes between different prediction outcomes. The background colors represent the four possible classification results: True Positives (TP) are shown in red, indicating correctly detected anomalies; False Positives (FP) appear in yellow, representing false alarms where normal behavior was misclassified as an anomaly; True Negatives (TN) are colored in blue, marking correctly identified normal behavior; and False Negatives (FN) are highlighted in green, showing missed anomalies. The curve represents the actual data points, and the color-coded sections help illustrate where the model performed correctly and where it misclassified, giving an intuitive understanding of the prediction results over time.

VI. Evaluation analysis and usage recommendations

As presented above, DebiAI offers both a Python module and a web-based Data-provider as alternatives for adding data into a project environment. In this section, we assess these services in terms of scalability, exhibit a specific trade-off between these two approaches and propose user recommendations on how to use them.

To benchmark the performance of DebiAI's two services above, we evaluated the insertion time provided by the application, whether it's the DebiAI Python Module or the Web Data Provider, by inserting randomly generated CSV files with varying sizes : 1 000, 10 000, 100 000 and 1 000 000 samples.

TABLE II: Time, in seconds (s), of data insertion for variable samples sizes along DebiAI's Web-based Data Provider and Python Module services.

Samples	Web Data-Provider	Python Module
1000	0.1s	1s
10 000	0.4s	8s
100 000	3s	62s
1 000 000	33s	465s

The evaluation of DebiAI's services, shows contrasting performance behaviors as dataset sizes increase. On the one hand, we observe a logarithmic trend on the side of DebiAI's Web Data-Provider, highlighting Web Data-Provider's scalability and consistent efficiency, even when handling large datasets. On the other hand, the DebiAI's Python Module demonstrates an exponential growth in insertion time, indicating a sharp increase in computational demands as the dataset size increases. While Python Module handles smaller datasets reasonably well, the exponential rise in insertion time for larger datasets, such as 1 000 000 samples, reveals significant scalability limitations for this alternative as can be easily seen in Table II.

This sheds evidence on a trade-off between accessibility and scalability arising on both DebiAI's services. We suggest to use DebiAI's Python Module as a quick-start approach for usage, and while the amount of data increases in the project environment, switching to the Web Data-Provider service included in DebiAI.

VII. Limitations

DebiAI is currently in its beta phase, which introduces several limitations. Not all functionalities have been fully implemented, such as complete persistence when navigating away from and returning to the analysis page, which may disrupt the user workflow. Additionally, as demonstrated in our benchmarking (Section VI), the platform's performance can degrade when handling large datasets exceeding 1 million samples, indicating a need for further optimizations. Users may also encounter occasional bugs or incomplete features, as the tool continues to evolve with new additions and improvements planned for future releases. Despite these limitations, DebiAI is actively used within our company, where it continues to add significant value to our machine learning workflows and projects.

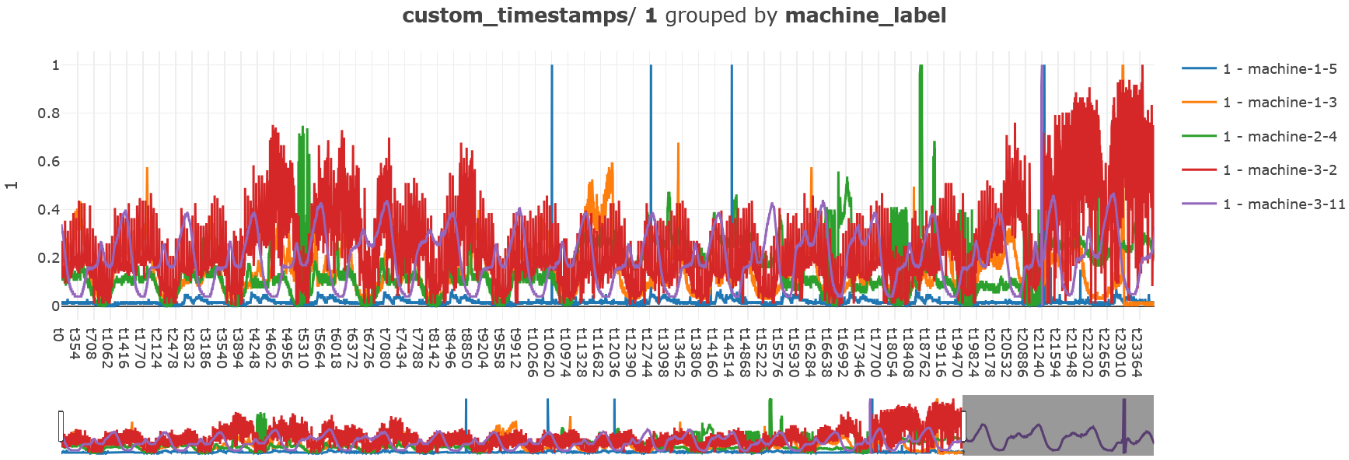


Fig. 17: DebiAI's RangeSlider provides a comprehensive representation of each of the 5 chosen machines' first KPI in the SMD dataset.

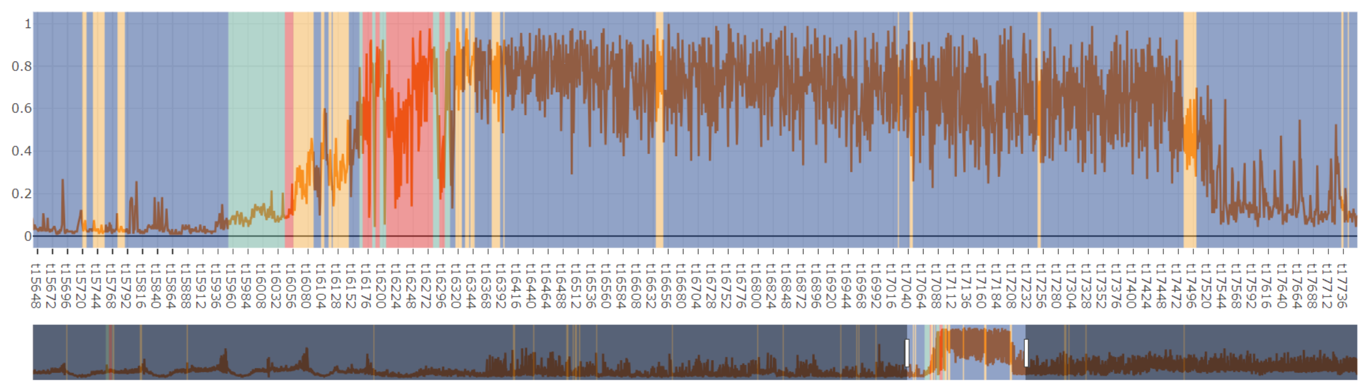


Fig. 18: DebiAI's RangeSlider applied on a single machine timeseries with anomaly detection algorithm performances compared to dataset ground truth using background plot colors on first KPI. TP: red, FP: yellow, TN: blue, FN: green

VIII. Conclusions and Perspectives

In this paper, we introduced DebiAI, a versatile web-based visual analytics tool that enhances data preparation process, quality assessment, model results analysis and comparison in ML projects. Its adaptability to various use cases and user-friendliness make it a valuable asset contributing to the trustworthiness in AI. For instance, we illustrated its application in a use case of 2D object detection task for driving assistance and server machine sensors analysis for machine's performances monitoring. As Machine Learning evolves, DebiAI can play a pivotal role in ensuring reliable and interpretable ML outcomes, solidifying its relevance in the field. Compared to our previous version, we simplified data and algorithms insertion as well as installation process with debiai-gui and docker. In DebiAI's outlook, the priorities are to enhance interoperability with the learning process to retrieve and analyze data from each cycle. The concepts of robustness and explainability are also tied to model's quality. Therefore, incorporating these metrics into the process is critical for overall trust. We aim to make it easier for users to integrate and run their own algorithms, be it custom or from a specific library such as DQM and TADkit libraries.

Acknowledgments

This work has been supported by the French Government under the "France 2030" program, as part of the SystemX Technological Research Institute. This work was conducted as part of the Confiance.ai program, which aims to develop innovative solutions for enhancing the reliability and trustworthiness of AI-based industrial systems. The team, especially T. M. & F. A., would like to thank Rémi Boyer, from IRT SystemX, for his feedback while testing DebiAI's different features and tutorials, and Jaime De Oliveira along with his team at Thales for their encouraging support and suggestions along the construction of this project.

References

[1] T. Mansion, R. Braud, A. Amrani, S. Chaouche, F. Adjed, and L. Cantat, "DebiAI: Open-Source Toolkit for Data Analysis, Visualisation and Evaluation in Machine Learning," in *ICAS 2024*, 2024.

[2] F. Sperrle, M. El-Assady, G. Guo, R. Borgo, D. H. Chau, A. Endert, and D. Keim, "A Survey of Human-Centered Evaluations in Human-Centered Machine Learning," *Computer Graphics Forum*, vol. 40, no. 3, pp. 543–568, Jun. 2021, https://doi.org/10.1111/cgf.14329, last accessed on 2024-11-20.

[3] T. Kaluarachchi, A. Reis, and S. Nanayakkara, "A review of Recent Deep Learning Approaches in Human-Centered Machine Larning," *Sensors*, vol. 21, no. 7, p. 2514, 2021.

[4] T. Capel and M. Brereton, "What is human-centered about human-centered AI? a map of the research landscape," in *Proceedings of the 2023 CHI conference on human factors in computing systems*, 2023, pp. 1–23.

[5] D. Sacha, M. Sedlmair, L. Zhang, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim, "What You See is What You Can hange: Human-centered Machine Learning by Interactive Visualization," *Neurocomputing*, vol. 268, pp. 164–175, 2017.

[6] Confiance.ai, "Confiance.ai Program," https:www.confiance.ai, 2024, online; accessed 19-February-2024.

[7] M. Adedjouma, C. Alix, L. Cantat, E. Jenn, J. Mattioli, B. Robert, F. Tschirhart, and J.-L. Voirin, "Engineering Dependable AI Systems," in *17th Annual System of Systems Engineering Conference (SOSE)*. Rochester, United States: IEEE, Jun. 2022, https://hal.science/hal-03700300, last accessed on 2024-11-29.

[8] M. Juliette, L. R. Xavier, B. Bertrand, C. Loic, T. Fabien, R. Boris, G. Rodolphe, and N. Yves, "AI Engineering to Deploy Reliable AI in Industry," *AI4I*, vol. 25, no. 1, 2023.

[9] Taylor, "TAYLOR Network Program," https://tailor-network.eu/, 2024, online; accessed 19-February-2024.

[10] J. Yuan, C. Chen, W. Yang, M. Liu, J. Xia, and S. Liu, "A Survey of Visual Analytics Techniques for Machine Learning," *Computational Visual Media*, vol. 7, no. 1, pp. 3–36, Mar. 2021, https://doi.org/10.1007/s41095-020-0191-7, last accessed on 2024-11-29.

[11] J. Wang, S. Liu, and W. Zhang, "Visual Analytics For Machine Learning: A Data Perspective Survey," *CoRR*, vol. abs/2307.07712, 2023, https://doi.org/10.48550/arXiv.2307.07712, last accessed on 2024-11-29.

[12] F. Hohman, K. Wongsuphasawat, M. B. Kery, and K. Patel, "Understanding and Visualizing Data Iteration in Machine Learning," in *Proceedings of the 2020 CHI conference on human factors in computing systems*, 2020, pp. 1–13.

[13] J. Françoise, B. Caramiaux, and T. Sanchez, "Marcelle: Composing Interactive Machine Learning Workflows and Interfaces," in *The 34th Annual ACM symposium on user interface software and technology*, 2021, pp. 39–53.

[14] S. Kandel, R. Parikh, A. Paepcke, J. M. Hellerstein, and J. Heer, "Profiler: Integrated Statistical Analysis and Visualization for Data Quality Assessment," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2012, pp. 547–554.

[15] M. Chakraborty and A. P. Kundan, "Grafana," in *Monitoring Cloud-Native Applications: Lead Agile Operations Confidently Using Open Source Software*. Springer, 2021, pp. 187–240.

[16] J. Turnbull, *Monitoring with Prometheus*. Turnbull Press, 2018.

[17] K. Ahmad and M. Ansari, "Hands-on influxdb," in *NoSQL*. Chapman and Hall/CRC, 2017, pp. 341–354.

[18] S.-E. Hong, J. Moon, and J. Lee, "Data Acquisition and Visualization for AI/ML-based Radio Resource Management Optimization in the ns-O-RAN Framework," in *2024 Fifteenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2024, pp. 476–478.

[19] E. G. Rani and D. Chetana, "Using GitHub and Grafana Tools: Data Visualization (DATA VIZ) in Big Data," in *Computer Vision and Robotics: Proceedings of CVR 2022*. Springer, 2023, pp. 477–491.

[20] A. Vulpe, C. Dobrin, A. Stefan, and A. Caranica, "AI/ML-based Real-Time Classification of Software Defined Networking TSraffic," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, 2023, pp. 1–7.

[21] M. Khorasani, M. Abdou, and J. H. Fernández, "Web Application Development with Streamlit," *Software Development*, pp. 498–507, 2022.

[22] E. Haedecke, M. Mock, and M. Akila, "ScrutinAI: A Visual Analytics Tool Supporting Semantic Assessments of Object Detection Models," *Computers & Graphics*, vol. 114, pp. 265–275, 2023, https://www.sciencedirect.com/science/article/pii/S009784932300105X, last accessed on 2024-11-20.

[23] J. Zhang, Y. Wang, P. Molino, L. Li, and D. S. Ebert, "Manifold: A Model-Agnostic Framework for Interpretation and Diagnosis of Machine Learning Models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 364–373, 2019.

[24] C. Chen, Y. Guo, F. Tian, S. Liu, W. Yang, Z. Wang, J. Wu, H. Su, H. Pfister, and S. Liu, "A Unified Interactive Model Evaluation for Classification, Object Detection, and Instance Segmentation in Computer Vision," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[25] S. Amershi, M. Chickering, S. Drucker, B. Lee, P. Simard, and J. Suh, "ModelTracker: Redesigning Performance Analysis Tools for Machine Learning," in *Proceedings of the Conference on Human Factors in Computing Systems (CHI 2015)*. ACM - Association for Computing Machinery, Apr. 2015, m/en-us/research/publication/modeltracker-redesigning-performance-analysis-tools-for-machine-learning, last accessed on 2024-11-29.

[26] D. Ren, S. Amershi, B. Lee, J. Suh, and J. D. Williams, "Squares: Supporting Interactive Performance Analysis for Multiclass Classifiers," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 01, pp. 61–70, jan 2017.

[27] C. Wang, W. Shi, J. Zhang, W. Wang, H. Pan, and F. Feng, "Debias Can be Unreliable: Mitigating Bias Issue in Evaluating Debiasing Recommendation," *arXiv preprint arXiv:2409.04810*, 2024.

[28] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricar, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Chennupati, S. Nayak, S. Mansoor, X. Perroton, and P. Perez, "WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving," 2021.

[29] S. Chaouche, Y. Randon, F. Adjed, N. Boudjani, and M. Ibn Khedher, "DQM: Data Quality Metrics for AI Components in the Industry," in *The symposium on AI Trustworthiness and Risk Assessment for Challenged Contexts (ATRACC)*, 2024, pp. 1–8.

[30] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricár, S. Milz, M. Simon, K. Amende *et al.*, "Woodscape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9308–9318.

[31] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context," 2015.

[32] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust Anomaly Detection for Multivariate Time Series Through Stochastic Recurrent Neural Network," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2828–2837.

[33] R. Martin. (2024) Topological Data Analysis for Anomaly Detection. https://confianceai.pages.irt-systemx.fr/ec_5/ec5_as3/tdaad/tech_docs.html, last accessed on 2014-09-15.