

TDS Simulator GUI: User guide

Nicoletta Marrani^a, Tim Hageman^{a,*}, Emilio Martínez-Pañeda^{a,*}

^a*Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK*

Abstract

This document presents a user guide for **TDS Simulator** (Python version), a tool for modelling and interpreting thermal desorption spectroscopy (TDS) experiments. This graphical user interface (GUI) offers two functionalities: (1) simulating TDS curves using the Oriani and McNabb-Foster hydrogen transport models, for arbitrary choices of parameters and hydrogen trap characteristics and (2) interpreting experimental TDS data via a novel neural network machine learning (ML) approach. This guide explains how to use the tool for both functionalities, including details on formatting experimental data, configuring parameters and interpreting results for the ML-based data fitting. The documentation also covers system requirements, installation procedures and includes two example cases to demonstrate each functionality.

If using this tool, please cite: Marrani, N., Hageman, T., Martínez-Pañeda, E., 2025. *A neural network machine-learning approach for characterising hydrogen trapping parameters from TDS experiments*. International Journal of Hydrogen Energy 167, 150874.

Contents

1	Introduction	2
2	Installation and setup	2
2.1	Requirements	2
2.2	Launching the GUI	3
3	GUI components and functionality	3
3.1	Interface layout	3
3.1.1	Simulation tab	4
3.1.2	Parameters tab	4
3.1.3	Hydrogen Traps tab	5
3.1.4	ML Data Fitting tab	6
3.2	GUI outputs and visualisation	8
3.3	Settings and controls	9

*Corresponding authors
Email addresses: tim.hageman@eng.ox.ac.uk (Tim Hageman), emilio.martinez-paneda@eng.ox.ac.uk (Emilio Martínez-Pañeda)

4 Sample results: TDS simulation	9
5 Sample results: ML data fitting	9
References	12

1. Introduction

Understanding hydrogen-material interactions, such as diffusion and trapping, provides insight into the susceptibility of a material to hydrogen embrittlement. Thermal desorption spectroscopy (TDS) is a widely used bulk experimental technique for quantifying the key trapping characteristics of metallic alloys, specifically the trapping capacity, including trap binding energy and density, of different microstructural features (e.g., dislocations, grain boundaries and precipitates). Interpreting the output of TDS experiments, i.e., extracting the number of trap sites and the binding energy and density associated with each trap from the TDS spectrum, is far from straightforward. To overcome the limitations of current TDS analysis techniques, we developed a machine learning-based approach, comprising a multi-neural network (NN) machine learning (ML) model trained exclusively on synthetic data, to predict the trapping parameters directly from experimental data.

Here, we present a guide to the graphical user interface (GUI) for **TDS Simulator** (Python version), a tool for modelling and interpreting TDS experiments. The tool offers two primary functionalities:

- **TDS Simulation:** This feature allows users to simulate TDS curves based on arbitrary parameters and hydrogen trap characteristics using the Oriani and McNabb-Foster hydrogen transport models.
- **ML data fitting:** This functionality enables the interpretation of experimental TDS data using the aforementioned novel multi-NN ML approach.

This guide explains how to use the tool’s two functionalities. For the ML data fitting functionality, it provides detailed instructions on how to format experimental data, configure parameters and interpret results. The documentation also includes essential background information on system requirements and installation procedures, along with two example cases, one for each functionality.

2. Installation and setup

2.1. Requirements

In addition to standard Python libraries, the following external software libraries are also required:

- TensorFlow (recommended version 2.19.0)
- Keras (recommended version 3.10.0)

Note: Python 3.11.0 is recommended.

2.2. Launching the GUI

Before launching the GUI, a virtual environment must be created and the required libraries installed.

Using VS Code:

1. Open the VS Code terminal (Ctrl+‘)
2. Run the following commands:

```
# Create virtual environment:
```

```
python -m venv .venv
```

```
# Activate virtual environment:
```

```
.venv\Scripts\activate      # Windows
```

```
# or
```

```
source .venv/bin/activate   # Mac/Linux
```

```
# Install required libraries:
```

```
pip install numpy pandas tensorflow scikit-learn matplotlib openpyxl seaborn
```

Once the environment is set up, launch the GUI by either:

- Running the `Launch_GUI.py` file directly in VS Code, or
- Executing the following command in the VS Code terminal: `python Launch_GUI.py`

3. GUI components and functionality

Upon launching the GUI, the main screen appears as depicted in Fig. 1. This screen is divided into two distinct panels: a left panel housing the tabbed interface for defining all inputs (Section 3.1) and a right panel serving as the main display for visualisation tools and controls (Sections 3.2 and 3.3).

3.1. Interface layout

As shown in Fig. 1, the GUI includes four main tabs, aimed at: (i) setting up the simulation characteristics (**Simulation** tab), (ii) defining the material, test and numerical parameters (**Parameters** tab), (iii) describing the trap characteristics (**Hydrogen Traps** tab) and (iv) fitting experimental data employing the ML-based approach presented in the associated manuscript (**ML Data Fitting** tab). The specifics of each tab are described in detail in the following sections.



Figure 1: GUI upon launch, highlighting key components. The default configuration loads the "Novak_200" case, which contains all parameters as published in the associated manuscript.

3.1.1. Simulation tab

The **Simulation** tab, depicted in Fig. 2(a), allows the user to define the hydrogen transport theory and configure the output characteristics. For the hydrogen trapping model, the GUI supports three frameworks: no traps (lattice only), McNabb-Foster and Oriani.

The graphical options for the output analysis allow users to customise both the vertical and horizontal axes. For the horizontal axis, the user can choose between temperature (T) in K or $^{\circ}\text{C}$ and time (t) in seconds. The vertical axis can represent either the hydrogen flux (J) or the hydrogen desorption rate (ΔC). For each of these output quantities, the user can select from a list of common units. The available units for flux are $\text{mol}/(\text{m}^2\text{s})$, $\text{mol}/(\text{cm}^2\text{s})$ and wppm m/s, while for desorption rate they are $\text{mol}/(\text{m}^3\text{s})$, $\text{mol}/(\text{cm}^3\text{s})$ and wppm/s.

3.1.2. Parameters tab

All inputs for the TDS experiment are defined in the **Parameters** tab. These are organised into three distinct categories: test, numerical and material inputs.

The test and material inputs are standard for TDS analysis. The numerical inputs, however, require careful consideration. These include the number of temperature evaluations, which directly corresponds to

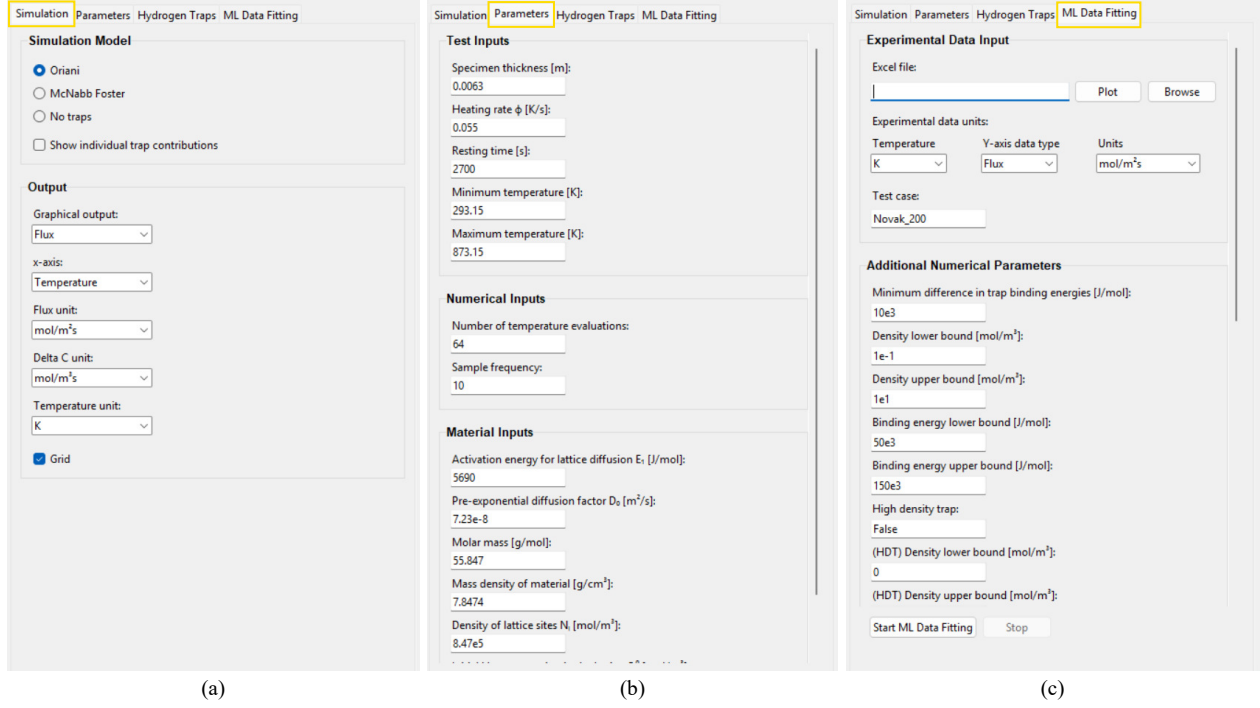


Figure 2: Layout and features of GUI tabs: (a) **Simulation** tab, (b) **Parameters** tab and (c) **ML Data Fitting** tab. The **Hydrogen Traps** tab is described in greater detail in Fig. 3, given the dependency of its contents on the selected transport theory. The interface displays default values for the "Novak_200" reference case.

the data points shown in the simulation output. This parameter also represents the feature inputs for the ML models and should be chosen carefully if conducting ML data fitting. A default value of 64 is provided.

A secondary numerical parameter, the sample frequency, is also required. This value determines the time increments between each temperature evaluation. This parameter allows the user to control the sampling rate while maintaining a consistent number of output data points. This is especially important for models with high trapping rates, as it allows the use of smaller time increments while keeping the total number of output data points consistent.

Note: Use the scroll bar provided to ensure all available inputs are set correctly.

3.1.3. *Hydrogen Traps* tab

The Hydrogen Traps tab is where the user defines the trapping characteristics of the system. First, the user specifies the number of traps, up to a maximum of six, using the drop-down menu in the top left of the tab. Next, the user must define the characteristics for each trap type. The relevant information required depends on the hydrogen trap model selected. For the Oriani model, the required inputs are the trap density N_T in sites/m³ and the trap binding energy ΔH in J/mol, as depicted in Fig. 3(a). For the McNabb-Foster model, the user must provide the trap density N_T , trapping E_t and de-trapping E_d energies, as shown in Fig. 3(b). By default, E_t is set equal to the lattice activation energy E_L and is fixed, while the user can vary E_d accordingly. An additional parameter, the vibration frequency or trapping rate ν , must also be provided.

This value is constant for all trap types and defaults to the Debye frequency, $\nu = 10^{13}$ Hz.

Finally, for the no-traps (lattice only) option, no inputs are required, and the tab remains blank (Fig. 3(c)).

Figure 3 displays three screenshots of the 'Hydrogen Traps' tab in a simulation software interface, showing different configurations for hydrogen transport models.

(a) **Oriani model:** The 'Number of traps' is set to 6. The interface shows six traps, each with input fields for 'Binding energy ΔH [J/mol]' and 'Density of trapping sites N_t [sites/m³]'. The values for the six traps are: Trap 1 (25000, 1.5e25), Trap 2 (30000, 1.5e25), Trap 3 (35000, 1.5e25), Trap 4 (40000, 1.5e25), Trap 5 (45000, 1.5e25), and Trap 6 (50000, 1.5e25).

(b) **McNabb-Foster model:** The 'Number of traps' is set to 6. The interface shows six traps, each with input fields for 'Act. E. for trapping E_t [J/mol]', 'Act. E. for detrapping E_d [J/mol]', and 'Density of trapping sites N_t [sites/m³]'. The values for the six traps are: Trap 1 (5690, 25000, 1.5e25), Trap 2 (5690, 30000, 1.5e25), Trap 3 (5690, 35000, 1.5e25), Trap 4 (5690, 40000, 1.5e25), Trap 5 (5690, 45000, 1.5e25), and Trap 6 (5690, 50000, 1.5e25). A red box highlights the 'Vibration frequency [Hz]' input field, which is set to 1e13.

(c) **No traps (lattice only):** The 'Number of traps' is set to 6. The interface shows a message: 'Simulation will run without hydrogen traps.'

Figure 3: Contents of **Hydrogen Traps** tab when different hydrogen transport models are selected: (a) Oriani, (b) McNabb-Foster and (c) no traps (lattice only).

3.1.4. *ML Data Fitting tab*

The ML Data Fitting tab (see Fig. 2(c)) is where the required parameters for ML model building and training are entered, and where the experimental data fitting is conducted. This tab is organised into three main sections: "Experimental Data Input," "Additional Numerical Parameters" and "ML Model Training Parameters." A detailed description of each input and the procedure to conduct the ML data fitting is provided below.

I. Experimental data and test case definition

To begin, users must select an experimental data file using the **Browse** button. The GUI requires users to specify the axes data types and units for this experimental data. All experimental data must be provided in Excel format (.xlsx or .xls).

After a file is selected, the system automatically processes and formats the data for ML model compatibility. The **Plot** button can then be used to display this post-processed data, allowing for a preliminary verification of the temperature range, flux magnitude and overall data quality without running a full analysis.

The test case, associated with the parameter inputs, must also be defined. The GUI's flexible parameter management system accommodates both predefined and custom test cases through the **Test case** input. If

the provided name matches a predefined case in the `ExpDataParameters` class, the system automatically loads all associated material properties, test parameters and numerical settings. The predefined cases correspond to the test cases of the manuscript (e.g., "Novak_200"). For new entries, the system automatically saves the GUI input values as a new parameter set for future use under the name provided by the user in the `Test case` input. The default configuration loads the "Novak_200" case with all parameters as published in the manuscript.

II. Additional numerical parameters

Next, the parameters required for the training data generation must be specified. These include:

- **Minimum difference in trap binding energies (J/mol):** Binding energy resolution between traps
- **Density bounds (mol/m³):** Range of trap densities considered in the analysis
- **Energy bounds (J/mol):** Range of binding energies considered in the analysis

Additionally, five parameters are available to specify a high-density, low-energy trap, a feature utilised in case studies 2 and 3 of the manuscript. When this trap is present, the High-density low-energy trap parameter should be set to "True," and the corresponding density and energy bounds must be defined. For the "Novak_200" test case, this feature is not required, so the High-density low-energy trap parameter is set to "False," and all associated parameters are set to "0."

III. ML model training parameters

Finally, the following parameters, required to configure the generation and training of the ML models, need to be defined:

- **CPU cores:** Number of processor cores allocated for parallel computation
- **Training data points:** Size of the synthetic dataset used for model training
- **Verification data points:** Size of the synthetic dataset used for model validation
- **Hyperparameter set:** Predefined optimised hyperparameters. If the user wishes to change these, they must do so in the `Model_Parameters.py` file.
- **Maximum traps:** Upper limit on the number of trap types the model can identify
- **Traps/Concentrations:** If "Random", traps are randomly generated. If a specific value is provided, data is generated with that value, and models are trained for that specific configuration
- **Regenerate data:** Forces regeneration of training datasets when set to "True"
- **Regenerate training:** Forces retraining of ML models when set to "True"

Note: The system will utilise existing training data and models for specified test cases unless regeneration is explicitly requested.

IV. Running ML data fitting

The **Start ML Data Fitting** button executes the complete workflow, including data generation, model training, validation, experimental data preprocessing, prediction and plotting. A **Stop** button is also provided, if the user wishes to halt the current analysis (requires restarting from the beginning).

The progress indicator (right centre in Fig. 1) provides real-time feedback on analysis stages. These are reported as: (i) Initialising material and model parameters, (ii) Setting up training parameters, (iii) Generating data and training models, (iv) Running model verification, (v) Processing experimental data and (vi) Making predictions and generating plots.

Notes: Some operations, particularly ML model training, may continue briefly after a stop request due to the underlying computational framework. This behaviour is normal and does not indicate a system error. Experimental data must be loaded before pressing the "Start ML Data Fitting" button.

3.2. GUI outputs and visualisation

The GUI provides two types of outputs to the user, in the form of (i) plots and (ii) a text output panel. The contents vary based on whether you are running TDS simulations or conducting ML data fitting.

The plot section gives you a visual representation of the TDS simulation or ML data fitting results.

- **TDS simulation:** After running a TDS simulation, the plot will show a blue line representing the computed desorption curve. If "Show individual trap contributions" was selected in the **Simulation** tab, the plot will also display additional dashed coloured lines showing the contribution of each individual trap. The plot units will automatically match the format selected in the "Output" section of the **Simulation** tab.
- **ML data fitting results:** When performing an ML data fit, the plot will display the uploaded experimental data as black markers with the model's fitted prediction overlaid as a red line. Again, if "Show individual trap contributions" is enabled, the model's predicted trap contributions will also appear as dashed lines.

The text output panel provides detailed written feedback and results throughout the analysis.

- **TDS simulation:** During a simulation, the output panel displays confirmation of the input parameters and trap settings. It also provides status messages to let you know when the simulation is complete. If the system detects any issues, error messages and warnings will appear here.
- **ML data fitting results:** As the ML model trains, this panel shows the progress of each analysis stage and provides real-time timing updates. Once the process is complete, the panel reports the identified

trap parameters, including the number of traps, their binding energies (J/mol) and their densities (mol/m³ and sites/m³).

3.3. Settings and controls

The top-right corner of the GUI contains the controls for simulation execution, data plotting and clearing of results. Specifically, these include:

- **Run Simulation:** Executes the TDS simulation based on the current parameters from all tabs except for the **ML Data Fitting** tab. The controls for conducting ML data fitting are located within the **ML Data Fitting** tab.
- **Clear Plot:** Removes all graphical data from the plot area while preserving input parameters, allowing fresh visualisation without losing configuration settings.
- **Clear All:** Resets the entire interface by clearing all input fields, selections and plot area.
- **Reset Default:** Restores all parameters to default values.
- **Export Data:** Saves current plot data to CSV format with appropriate headers.

4. Sample results: TDS simulation

Following the completion of a TDS simulation, the GUI appears as depicted in Fig. 4, showing the final analysis results.

5. Sample results: ML data fitting

The default GUI configuration is set to run the ML data fitting for the experimental data from Novak et al. [1], corresponding to a high-strength AISI tempered martensitic steel ($\phi = 200^\circ\text{C/h}$). This is defined as the **Test case**: "Novak_200".

Optimised hyperparameters (as described in the manuscript) are automatically selected by setting the **ParameterSet** variable to "optimised" within the **ML Model Training Parameters** input frame. The expected output from this analysis is displayed in Fig. 5.

Upon completion of the analysis, the text output panel should display the entire analysis sequence, including test case identification, input parameter confirmation, material and trap model specifications, analysis execution and final results, as follows:

```
--- Running ML Fitting ---  
ML Test case: Novak_200  
--- ML Fitting Inputs ---  
Experimental Data File: "File_path"
```

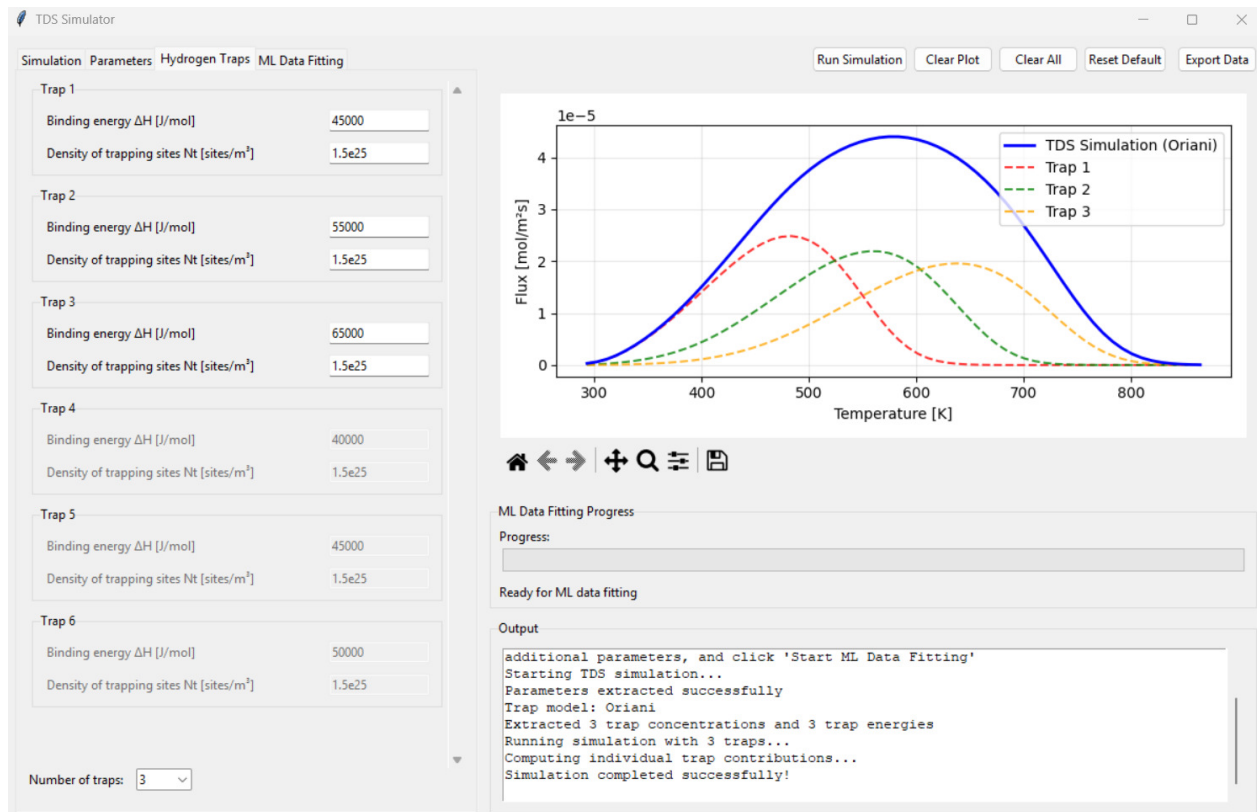


Figure 4: Example of the GUI output after running a TDS simulation. The plot shows both the computed TDS curve and individual trap contributions.

Material parameters:

NL: 8.47×10^5
 E_Diff: 5690.0
 D0: 7.23×10^{-8}
 C0: 0.06
 TrapRate: 1.00×10^{13}
 MolMass: 55.847
 MassDensity: 7.8474

Test parameters:

tRest: 2700.0
 HeatingRate: 0.055
 Thickness: 0.0063
 TMax: 873.15
 TMin: 293.15

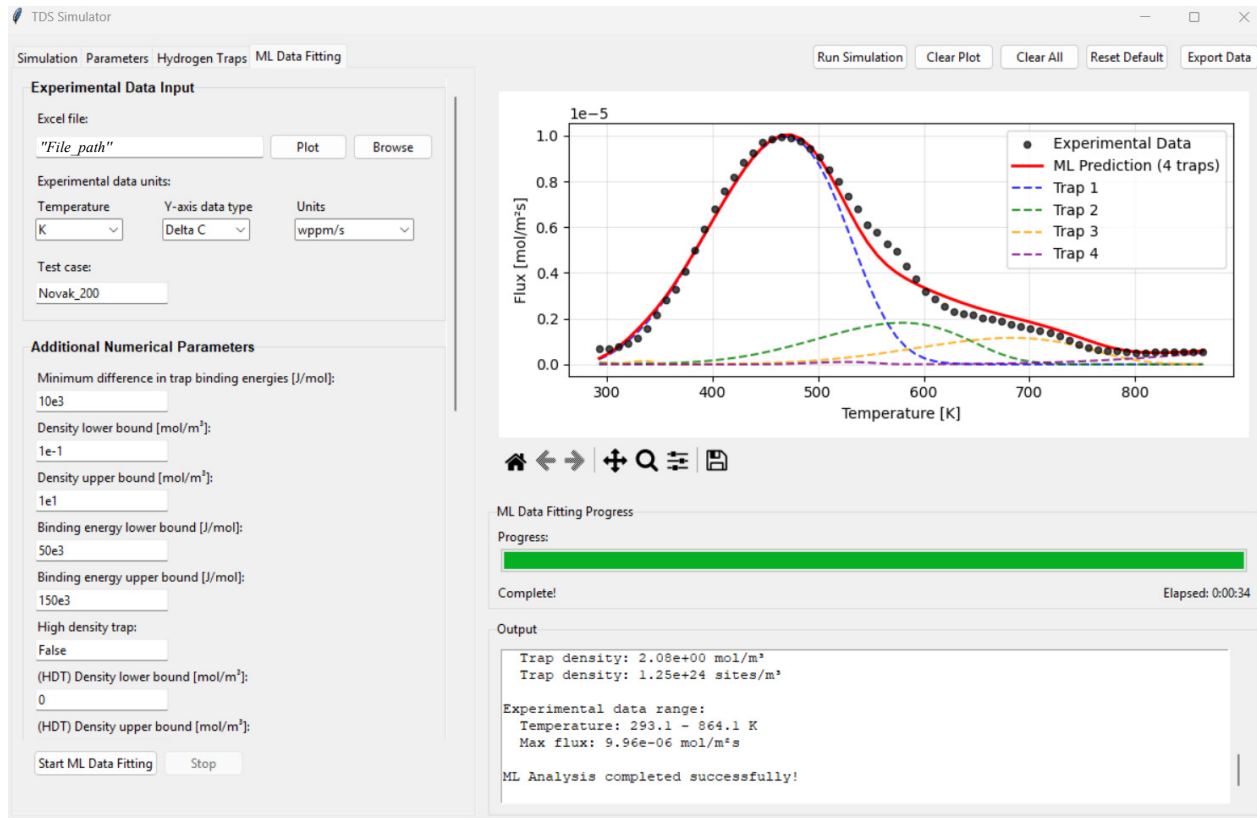


Figure 5: Example of the GUI output after analysis completion. The top right shows the plot comparing experimental data with model predictions and individual trap contributions. The bottom right displays the quantitative results, including the number of traps, binding energies and trap densities. This example shows the "Novak_200" test case [1] as presented in the manuscript.

ML Training parameters:

Traps: Random

Concentrations: Random

MaxTraps: 4

ParameterSet: optimised

Regenerate_Data: False

Regenerate_Training: False

Trap model: McNabb

ML parameters collected! Starting fitting...

Initialising parameters...

Generating data and training models...

Running model verification...

Processing experimental data...

Making predictions...

ML Analysis completed! Generating plots...

=== ML PREDICTION RESULTS ===

Number of predicted traps: 4

Trap 1:

De-trapping energy: 58290.01 J/mol

Trap density: 9.26e+00 mol/m³

Trap density: 5.58e+24 sites/m³

Trap 2:

De-trapping energy: 80340.16 J/mol

Trap density: 1.86e+00 mol/m³

Trap density: 1.12e+24 sites/m³

Trap 3:

De-trapping energy: 97620.30 J/mol

Trap density: 1.35e+00 mol/m³

Trap density: 8,13e+23 sites/m³

Trap 4:

De-trapping energy: 150451.38 J/mol

Trap density: 2.08e+00 mol/m³

Trap density: 1,25e+24 sites/m³

Experimental data range:

Temperature: 293.1 - 864.1 K

Max flux: 9.96e-06 mol/m²s

ML Analysis completed successfully!

References

- [1] P. Novak, R. Yuan, B.P. Somerday, P. Sofronis, and R.O. Ritchie. A statistical, physical-based, micro-mechanical model of hydrogen-induced intergranular fracture in steel. *Journal of the Mechanics and Physics of Solids*, 58(2):206–226, February 2010. Publisher: Elsevier BV.