TDS ML-based analysis 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 the TDS ML-based analysis GUI, a tool that facilitates the use of the developed neural network machine learning approach for interpreting TDS experiments. The GUI provides an intuitive interface with real-time progress tracking and result visualisation. Users can work with built-in test cases or create custom configurations, with all results being automatically saved for future reference. This guide explains how to use the tool, format experimental data, configure parameters and interpret results. The documentation also includes background information on system requirements and installation procedures.

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

*Corresponding authors

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

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) designed to facilitate the use of the Python implementation of the proposed ML approach. The guide outlines the installation and setup steps, GUI components (e.g., input panel, output panel, progress bar and control buttons) and their functions, instructions for formatting experimental data, interpreting results and accessing saved files. An example test case is provided to demonstrate the expected analysis results.

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

Prior to 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 key features

Upon launching the GUI, the interface shown in Fig. 1 appears. The left side contains input parameters (outlined in Section 3.1), whilst the right side displays outputs (discussed in Section 3.2).

The interface includes a central scroll bar for viewing all input parameters, analysis control buttons (bottom left) for managing workflow execution, progress tracking frames for monitoring completion status (central right) and experimental data input capabilities with optional plotting for verification (top left).

The analysis control buttons include:

- `Run Analysis`, which executes the complete workflow including data generation, model training, validation, experimental data preprocessing, prediction, and plotting;
- `Stop Analysis`, which halts the current analysis (requires restarting from the beginning);
- `Clear`, which removes all results and input values; and
- `Reset to Defaults`, which returns input values to the "Novak_200" test case settings.

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 "Run Analysis" button.

3.1. Inputs

An overview of all required inputs is provided in Fig. 2. These are organised into six categories: (i) experimental data input, (ii) material properties, (iii) trap model parameters, (iv) test parameters, (v) numerical parameters and (vi) ML model training parameters.

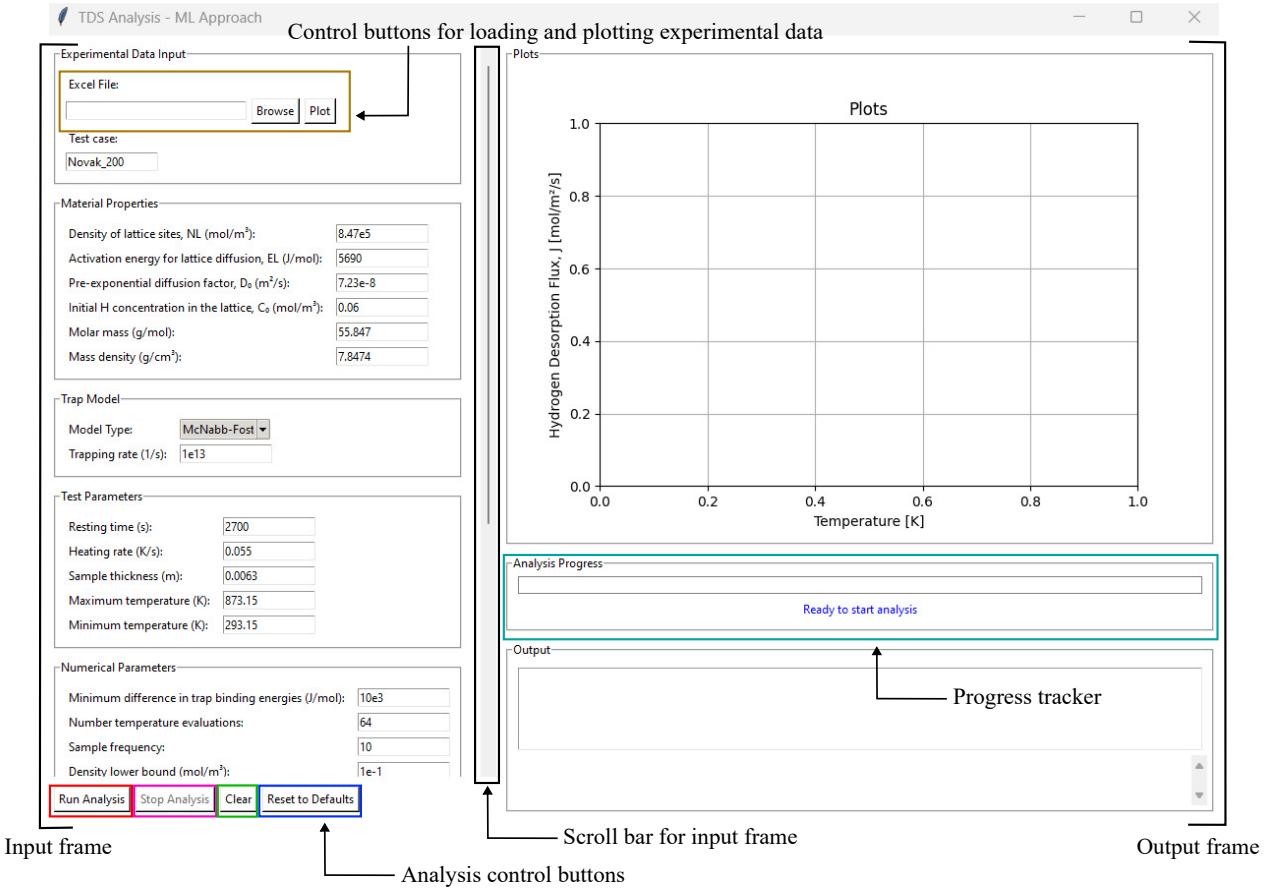

Figure 1: GUI interface upon launch, highlighting key components. The default configuration loads the "Novak_200" case, which contains all parameters as published in the associated manuscript.

3.1.1. Experimental data and test case definition

Data loading and verification:

The `Browse` button can be used to select the experimental data file. The system automatically processes the data upon selection. The `Plot` button displays experimental data for verification without running the full analysis, allowing confirmation of temperature range, flux magnitude, and data quality before complete modelling.

Data format requirements:

Experimental data must be provided in Excel format (.xlsx or .xls) with temperature values in Kelvin and hydrogen desorption flux in wppm/s (weight parts per million per second). The processing module automatically handles unit conversions and formatting for simulation compatibility.

Dataset management system:

The GUI operates with a flexible parameter management system that accommodates both predefined and custom test cases through the Test Case Name (`ExpName`) field. If the `ExpName` matches a predefined case in the `ExpDataParameters` class (such as "Novak_200"), the system automatically loads all associated material

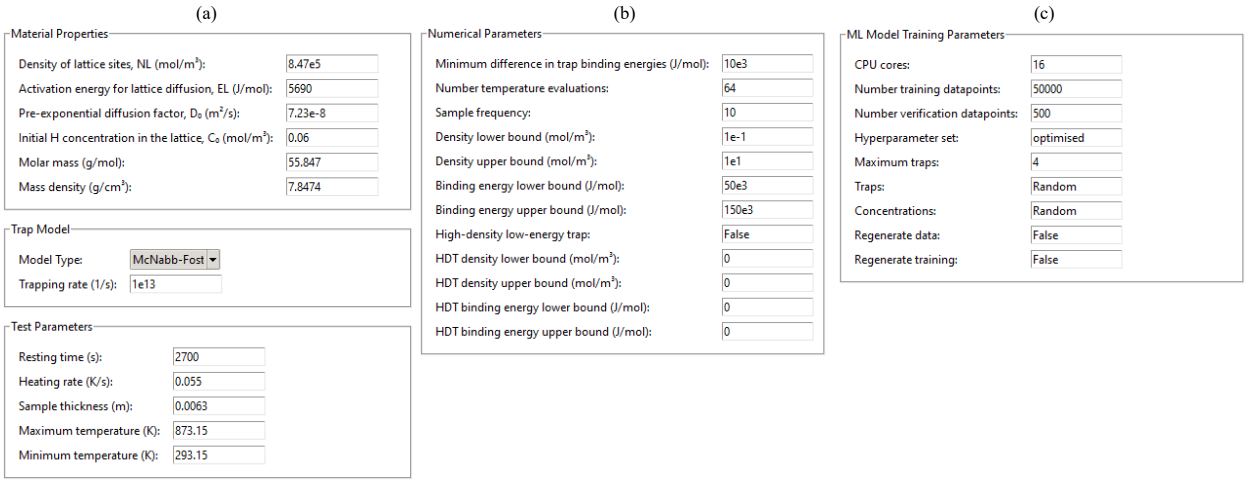

Figure 2: GUI parameter input sections: (a) material properties, trap model and test parameters, (b) numerical parameters and (c) machine learning configuration settings. The interface displays default values for the "Novak_200" reference case.

properties, test parameters and numerical settings. For unique `ExpName` entries, the system creates a new parameter set using the GUI input values and automatically saves them for future use.

The default configuration loads the "Novak_200" case with all parameters as published in the associated manuscript.

3.1.2. Material properties

The following material properties must be specified:

- `Density of lattice sites,` N_{L} `(mol/m`^{3}`)`
- `Activation energy for lattice diffusion,` E_{L} `(J/mol)`
- `Pre-exponential diffusion factor,` D_0 `(m`^{2}`/s)`
- `Initial H concentration in the lattice,` C_0 `(mol/m`^{3}`)`
- `Molar mass (g/mol)`
- `Mass density (g/cm`^{3}`)`

3.1.3. Trap model selection

Two trapping models are supported by the framework:

- **McNabb-Foster Model**
- **Oriani Model**

For the McNabb-Foster model, the `Trapping rate (1/s)` parameter, which defines the rate constant for hydrogen capture by traps, must also be specified. This field is disabled when the Oriani model is selected. The default value is set to the Debye frequency (i.e., 10^{13} Hz).

3.1.4. Test parameters

The following test parameters, which define the TDS setup, are required:

- `Resting time (s)`: Duration between charging and heating phases
- `Heating rate (K/s)`: Linear temperature ramp rate
- `Sample thickness (m)`: Physical thickness of the analysed specimen
- `Maximum temperature (K)`: Upper limit of the temperature ramp
- `Minimum temperature (K)`: Starting temperature of the analysis, typically taken as room temperature ($T = 293.15$ K)

3.1.5. Numerical parameters

These parameters are required for the TDS simulation (i.e., the training data generation step):

- `Minimum difference in trap binding energies (J/mol)`: Binding energy resolution between traps
- `Number of temperature evaluations`: Discretisation points for temperature evaluations
- `Sample frequency`: Number of time increments between temperature evaluations
- `Density bounds (mol/m`^{3}`)`: Range of trap densities considered in the analysis
- `Energy bounds (J/mol)`: Range of binding energies considered in the analysis

High-density trap configuration:

Five additional parameters are provided to enable users to specify the presence of a high-density low-energy trap. This feature is utilised in case studies 2 and 3 of the manuscript. When such a 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 accordingly.

For the "Novak_200" test case, this feature is not required. Consequently, the `High-density low-energy trap` parameter is set to "False" and all associated `HDT` parameters are set to "0".

3.1.6. ML model training parameters

Configuration parameters for ML model generation and training:

- `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 utilises existing training data and models for specified test cases unless regeneration is explicitly requested.

3.2. Outputs

The GUI provides three main types of output to the user:

Plots:

The main plotting area displays either the post-processed experimental data (when the `Load` button is selected) or a comprehensive comparison between experimental data and ML model predictions. This comparison includes both the reconstructed spectrum based on ML model predictions and the individual reconstructed spectra showing contributions from each identified trap type.

Text output panel:

The output panel provides detailed quantitative results and real-time progress information throughout the analysis process. Upon completion, it displays the number of identified traps along with their corresponding binding energies and densities. A detailed example of the complete output is provided in Section 4.

Additional saved outputs:

All analysis results are automatically saved to the "Figures/{ExpName}/" folder for future reference. This includes the main comparison plot between predicted and experimental spectra (with individual trap contributions) that appears in the GUI plotting area, as well as additional scatter plots generated during the model validation phase that provide insights into model performance and accuracy.

3.3. Progress monitoring and analysis control

The progress indicator 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.

4. Sample results: "Novak_200" Case Study

Running the GUI in the default setting will conduct the analysis for the experimental data obtained by Novak et al. [1], corresponding to a high-strength AISI tempered martensitic steel ($\phi = 200°C/h$), i.e., test case 1 of the manuscript. This is defined as `ExpName`: "Novak_200". The optimised hyperparameters (as described in the manuscript) have been selected by setting the `ParameterSet` variable in the `ML Model Training Parameters` input frame as "optimised".

Running the analysis should produce the results shown in Fig. 3.

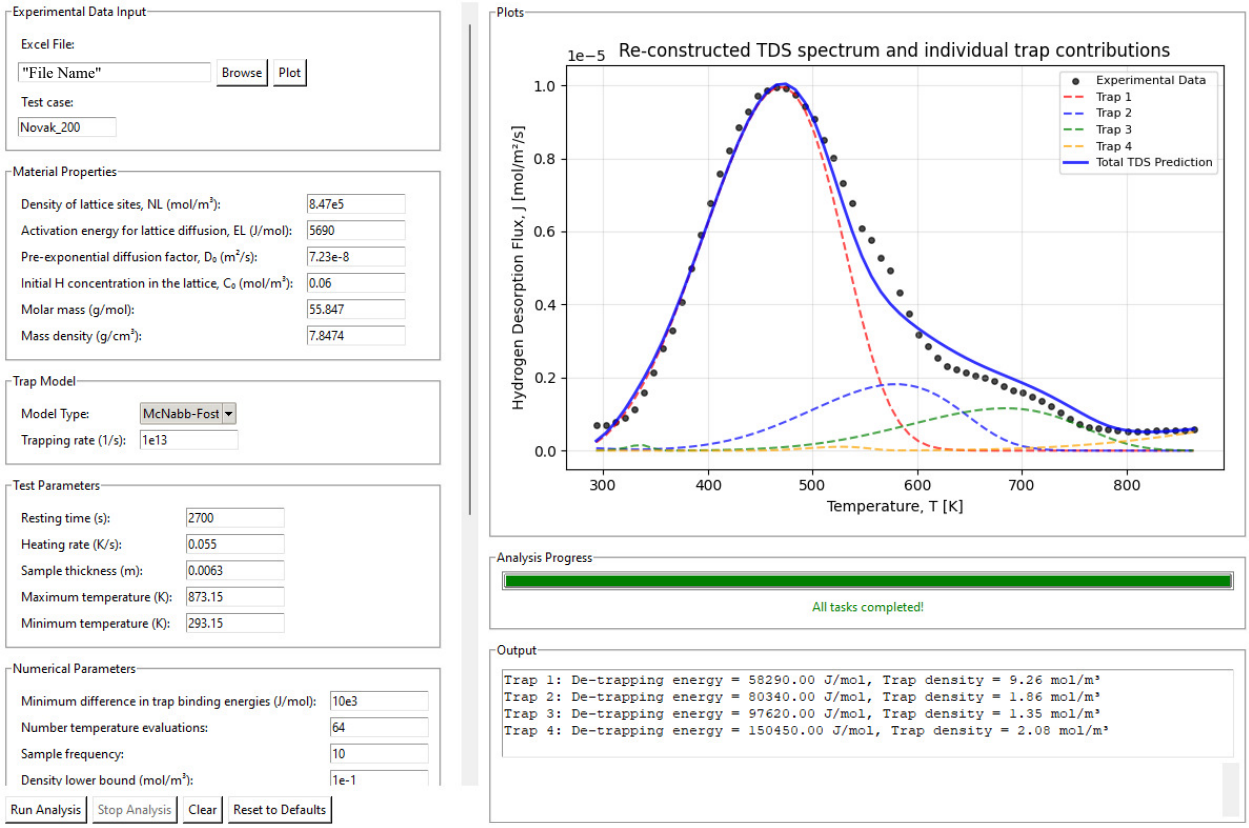

Figure 3: Example of the GUI output panel after analysis completion. The top right shows the plot comparing experimental data with model predictions and individual trap contributions. The bottom right displays the quantitative results, including the number of traps, binding energies and trap densities. This example shows the "Novak_200" test case [1] as presented in the main manuscript.

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

```
--- Running Analysis ---
Test case: Novak_200
```

```
--- Inputs for Analysis ---
Experimental Data File: Exp_Data_Filename.xlsx

Material parameters:
  NL: 8.47e+05
  E_Diff: 5690.0
  D0: 7.23e-08
  C0: 0.06
  TrapRate: 1.00e+13
  MolMass: 55.847
  MassDensity: 7.8474

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

Numerical parameters:
  dEMin: 1.00e+04
  ntp: 64
  SampleFreq: 10
  NRange: [0.1, 10.0]
  ERange: [50000.0, 150000.0]
  NumTraining: 50000
  NumVerification: 500
  n_cpu_cores: 16

Trap model: McNabb
Parameters collected!
Starting analysis...

Analysis completed successfully!
Plots saved to 'Figures/Novak_200/' folder
```

```
Predicted Traps: 4
Trap 1: De-trapping energy = 58290.01 J/mol, Trap density = 9.26 mol/m³
Trap 2: De-trapping energy = 80340.16 J/mol, Trap density = 1.86 mol/m³
Trap 3: De-trapping energy = 97620.30 J/mol, Trap density = 1.35 mol/m³
Trap 4: De-trapping energy = 150.451.38 J/mol, Trap density = 2.08 mol/m³
```

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.