# Purpose of Document

## Author and version:

Jim Smith (UWE), [james.smith@uwe.ac.uk](mailto:james.smith@uwe.ac.uk) . V0.1, June 2024.

Reviewed by: Amy Tilbrook 2024

V0.2 many contributors Feb 2025

V0.3 Updated Jim Smith June 2025

**The purpose of this version** is to provoke a discussion about the appropriate accessible wording - for both researchers and TRE staff - of the final version.

## Intended Audience and document aim

Researchers and TRE staff involved in the development of projects that may use machine learning to create models trained on confidential data.

This document is intended to inform discussion between Trusted Research Environments (TREs) and researchers *prior to research starting in the TRE* – for example, during the data access application process.

The intention is to make it easier for trained models to be taken out of a TRE by providing a basis for agreement on how the risk of ‘privacy leakage’ from a trained Machine Learning model will be assessed, and what the researcher will provide so that assessment can take place.

**It should be** noted that running attacks for risk assessment is only one element of the process of discussion/agreement of how ‘privacy by design’ can be embedded. Wider discussion should include consideration and selection of the types of data and ML algorithm to be used, and of appropriate strategies for risk mitigation

## Notation

We use the terms:

* **TRE** to refer to a Trusted Research Environment[[1]](#footnote-2) - however named (e.g. the NHS SDEs)
* **SDC** to refer to the process of output Statistical Disclosure Control, also known as output-checking.
* **Deep Learning** as a subset of **Machine Learning**, to refer to classes of algorithms within the field of Artificial Intelligence, used to create predictive models, through a process of training on a dataset.
* ‘**model’** to refer to the trained model, and associated files, that the researcher wishes to output from the TRE.
* ‘**training data’**, ‘**validation data**’ and ‘**test data**’ to refer to the three separate partitions of the available data. Typically:
  + Training data is used for *learning*- updating the parameters within a model.
  + Validation data is used for *model* selection - to compare models but not update weights.
  + Test data is used at the end of the process to *estimate the accuracy* of the selected model on unseen data.
* ‘**privacy leakage**’ to refer to the results of various ‘attacks’ on a model that may reveal (parts of) the confidential data it was trained on (sec 3.2).
* **sacro-ml** to refer to the toolkit produced in the [GRAIMATTER](https://dareuk.org.uk/how-we-work/previous-activities/dare-uk-phase-1-sprint-exemplar-projects/graimatter-guidelines-and-resources-for-artificial-intelligence-model-access-from-trusted-research-environments/), [SACRO](https://dareuk.org.uk/how-we-work/previous-activities/dare-uk-phase-1-driver-projects/sacro-semi-automated-checking-of-research-outputs/), and subsequent projects (e.g. [TREvolution](https://dareuk.org.uk/how-we-work/ongoing-activities/trevolution/#:~:text=TREvolution%2C%20funded%20by%20UK%20Research,data%20infrastructures%20—%20secure%20environments%20where)) as implemented in the python package [sacroml](https://github.com/AI-SDC/SACRO-ML).
* Text in blue font is intended to provide illustrative examples.

## Assumptions:

* Research will take place in an environment in which the python language is available and the package sacro-ml has been installed by the TRE staff, or can be installed by the researcher.
* Models will be created and trained using a mixture of modules from scikit-learn, xgboost, and/or one or both of keras/tensorflow and pytorch. (This list of supported packages is expected to expand in due course).
* The model to be output must output numerical predictions. For example, the probability that a given record comes from a given class A (a number between 0 and 1), rather than the text label “classA”.
* Researchers are familiar with standard Statistical Disclosure Control (SDC) concepts, even if they may not recognize them by name. These include methods such as k-anonymity (e.g., data treatments like suppression or rounding), threshold rules (ensuring minimum cell counts), and class disclosure controls.
* All parties (researchers, TRE staff, and project approval boards) are aware no model can ever be guaranteed to be completely immune to attacks.

Therefore, the level of confidence in a model’s security depends on how many different tests for vulnerability it successfully passes.

* Researchers *may* make use of tools (such as sacroml) to assess the vulnerability of their models, and use that information to adapt their pre-processing and training workflows accordingly,
  + However, there will be a single stage where they will formally request release of the final trained model and provide evidence required to perform the vulnerability testing.
  + The evidence, and any additional/ subsequent tests done by either the researcher or the TRE staff will be retained by the TRE provider.
  + Release of models may still be denied for other reasons.
* Researchers understand that:
  + TREs may have different Information Governance Requirements and operational practice.
  + **Therefore, working with the TRE staff from the outset will result in a far greater chance of having models appropriately risk assessed and the output decision made quickly.**
  + TRE staff may need to see their code. Whether this requires a Non-Disclosure Agreement is out of the scope of this document.

# Summary of recommendations

|  |  |  |
| --- | --- | --- |
| What is needed? | Why is it needed? | Comments/Details |
| Researcher provides details of their planned training and testing process at project approval time. | If a researcher plans to use cross-validation[[2]](#footnote-3) to estimate accuracy, then *all* the data given to them should be treated as ‘training’ data.  In that case the TRE **must** keep some data back from researchers in order to run certain attacks. |  |
| Researcher provides details of preprocessing applied to ‘raw’ data before it is input to the model.  Note that deciding the most effective pre-processing is a routine part of the Machine Learning workflow conducted *inside* the TRE.  Note that the sacro-ml package is currently being refined to make the process of specifying preprocessing as simple as possible | 1. So attribute inference attacks can be run 2. If cross validation is used, or just to strengthen certain attacks, TREs may keep some data back from researchers.   Hence, they must be able to apply the preprocessing to any withheld data, so they can present it to the model.   1. Because in certain cases TRES may wish to be able to see all of the researcher’s code. It is good practice for the ‘pre-processing’ code to be defined in ‘functions’, separate from the code used to train the model. Code that is separated into functions/modules is easier to scrutinise and understand. .   For example (1): if the user has standardised a variable to the range [0,1] using a `min-max scaler’, then the extreme values in the training data can be reverse-engineered. Whether this is an issue will depend on the data.  For example (2) if the user has (incorrectly) applied scaling to the data *before splitting it into training and test sets,* then the preprocessing also contains information about the test set.   1. **Because it may be possible to argue that pre-processing renders the dataset sufficiently anonymised that the model can safely be released** | In the form of:   * a single file of python code containing a method (preferably called ***preprocess()*** ) which takes in data in the ‘raw’ format provided and outputs it in the form presented to the model. * This might include ‘normalising’ variables, standardising image sizes, etc. * a mapping where appropriate. For example, if a ‘raw’ feature that takes one of *n* distinct values has been transformed via ‘one-hot-encoding’ into *n* new complementary binary features, it is useful to know which these are (and that by inference they must sum to 1)   Supporting contextual documentation may be appropriate to explain to TREs how the preprocessing has been conducted, variable names chosen, etc. |
| Researcher provides sufficient details to exactly replicate the training / test splits. | Membership and attribute inference attacks quantify the risk that an external attacker could reliably infer:   * *whether* a record was in the training set; and * *missing values* from a training record.   Quantifying these risks requires knowledge of **exactly** which records were used to train the model.    The assessment process can be improved via knowledge of exactly which records were used to test the trained model. | This needs to be in machine-actionable format - as either separate files/directories or as two lists of filenames.  Ideally researchers would provide both the `raw’ and preprocessed data as files to be ingested by sacro-ml.  If ‘raw’ format data is not available, it may not be possible to run attribute inference attacks.  If train/test data is only provided in ‘raw’ format then it **must** be possible to run the code to preprocess that data.  **Note this preprocessing may in future be automated, but currently requires manual input from TRE staff** |
| Researcher provides sufficient details (filepaths etc.) to load the model from file | Most attacks require the ability to load the stored file and access it. |  |
| Researcher runs a script (part of the sacroml toolkit) to provide all those details | Capturing the information needed to run attacks in a standardised format enables:   * storing the information that might be useful for a model-use register * decoupling model training from model risk assessment. That enables these processes to happen in separate ‘virtual areas’ of the TRE if desirable | This does not stop researchers running attacks themselves. |

# Appendix A: Background: What risks does SACRO\_ML assess and how?

This section is provided for background information only.

It is not mandatory to understanding the above.

## Summary

The sacro-ml toolkit provides support for automatically running a variety of tests to assess different form of attacks and how likely it is an attacker could find out confidential information.

The tool recreates the preprocessing of datasets, loads the model and parameters, and performs tests on 3 types of attacks based on the worst-case scenario (described below).

* some types of attacks require the full pre-processing to be available,
* others can be done with the preprocessed data that is fed into the model,
* and the others can be done using only the probabilities the model outputs different records
* however these last are the weakest type and do not provide much assurance of the safety of the model, especially in representative data

Below we briefly describe these tests, and what data needs to be made available to the risk assessment process.

## Membership and Attribute inference attacks

The [GRAIMATTER Green paper](https://doi.org/10.5281/zenodo.7089491) describes:

* Membership Inference as “*the risk that an attacker … can create systems that identify whether a given data point was part of the data used to train the released mode*l”
* Attribute Inference as “*the risk that an attacker, given partial information about a person, can retrieve values for missing attributes in a way that gives them more information than they could derive just from descriptions of the overall distribution of values in the datasets*”

## Worst-Case Scenarios for estimating the upper bound on risks.

The attacks implemented in sacro-ml are deliberately set up to ‘future-proof’ the risk assessment, by removing elements of the uncertainty relating to the way data is sampled.

The GRAIMATTER report and others have pointed out that typically attackers will be focussed on the ‘extreme’ cases where they can assert with confidence that a person’s data was (or wasn’t) used to train a model.

* This has implications for the choice of risk metrics.

Sacro-ml currently reports a range of metrics. The intention is for the developers and stake-holders to co-design the most informative presentation of these results.

* This also has implications for the attack ‘set-up’: in particular for attribute inference, the simulated attack should be allowed to say, ‘*don’t know’*, rather than forcing it to make prediction*s*. This has a dramatic effect on the accuracy of the predictions it does make.

Thus, sacro-ml estimates an upper-bound of the risk through a ‘worst-case’ scenario, by posing the question:

*How accurate are the predictions that an attacker can make given*

* *perfect knowledge of what is in the training data or not,*
* *not requiring an in/out prediction to be made for every record*

Currently, sacro-ml implements a number of different attacks based on the model’s

* output probabilities: the premise being that a model will be more confident about records it has seen during training[[3]](#footnote-4).

In some cases, these may be provided in a file. Generally it is more robust (i.e. relies less on trust and has less scope for human error) for the model and data to be loaded and create these at ‘attack-time’

* ‘losses’: the premise being that the chance of a model’s prediction being incorrect for a given record *may be* different if the record was used for training[[4]](#footnote-5).

These attacks absolutely require being able to load model and data.

The intention is that this list will be continuously updated as the field evolves.

## Implications for risk assessment

1. **Given only the model’s output probabilities for train/test datasets, sacro-ml can only run probability-based membership inference attacks**.

However, since these attacks have been questioned in the literature, they are more useful as an early warning’ system

* + possibly avoiding computational expense if the attacks are ‘successful’
  + but only providing limited assurance if the attacks ‘fail’.

1. **All other attacks need to know which records were used for training the model**.
2. **All but the weakest attacks require that sacro-ml can load the model**, query its parameters, and use it to make predictions.
3. **Membership inference attacks use ‘pre-processed’ data**.
   * The toolkit can ingest training and test data in both ‘raw’ forms (as provided by the TRE) and ‘pre-processed’ (as presented to the model).
   * If only the former is available, then the pre-processing code must be made available in a format that can be used by the toolkit.
4. Attribute inference attacks need to know how the data was pre-processed.

For example, whether a categorical variable with N levels has been ‘one-hot-encoded’ into N binary variables. If this is not available, attribute inference attacks cannot be performed.

## ‘Structural’ Attacks

These attacks implement concepts from the SDC of traditional outputs such as ‘residual degrees of freedom’, ‘k-anonymity’ and ‘class disclosure’.

### Implications for risk assessment

* The model must be provided in a format that can be loaded by the toolkit and have its hyper-parameters queried.
* Some of these structural measures need to know, for each training record, the model’s output probabilities for each possible label (class). Either
  + This information could be provided in a file (if the TRE is content),
  + or the training data must be provided in preprocessed form so it can be input to the loaded model,
  + or the training data could be provided in ‘raw’ form – in which case the preprocessing code must also be made available for use.

1. see [UK TRE glossary](https://glossary.uktre.org/en/latest/#term-trusted-research-environment--tre-) for a working definition [↑](#footnote-ref-2)
2. An approach to estimating the accuracy on unseen data that averages over repeated train-test splits. Typically, the final model is then trained using the whole dataset. [↑](#footnote-ref-3)
3. As these are computationally cheap, sacro-ml runs these attacks. However, recent research suggests they are weaker for ‘representative’ training data, since they do not take into account the difficulty of making a correct prediction, which is typically greater for ‘edge-cases’. [↑](#footnote-ref-4)
4. At the time of writing these – such as the Likelihood Ratio Attack (LIRA)are ‘State of the Art’. [↑](#footnote-ref-5)