

**Replication code for
“An Explainable Attention Network for Fraud Detection
in Claims Management”**

Helmut Farbmacher, Leander Löw, Martin Spindler
Journal of Econometrics, doi: [10.1016/j.jeconom.2020.05.021](https://doi.org/10.1016/j.jeconom.2020.05.021)

For inquiries and questions, please get in touch with:

Helmut Farbmacher, farbmacher@mpisoc.mpg.de or farbmacher@econ.lmu.de

You may find potential updates here:

<https://github.com/farbmacher/claims-management-joe>

Replication of models

Please note that the original data is confidential and cannot be shared. To illustrate our code, we generated some artificial claims. This data is not intended to replicate the results in our paper.

0. Prerequisites:

Install Anaconda on our machine, for more details see

<https://www.anaconda.com/products/individual>

(we used Anaconda3-2019.10-MacOSX, conda 4.9.2 (macOS) and conda 4.8.3 (Windows))

I. Build only our deep learning model: (no administrator rights are necessary)

1. Create a conda environment using `environment.yml`.
Copy `environment.yml` to your home folder. Type in terminal (macOS) or Anaconda Prompt (Windows):
 - (a) `conda env create -f environment.yml`
 - (b) `conda activate claims-management-joe`
 - (c) If you have a CUDA GPU on your machine, you can activate it by typing
`pip install torch==1.7.1+cu110 -f https://download.pytorch.org/whl/torch_stable.html`
You can change the number of workers with `CORES_TO_USE` in `constants_imports.py`
 - (d) `spyder`
2. Change working directory in Spyder to the ‘Code JoE’ folder
3. To prepare data, run `loading_data.py`
4. To fit the deep learning model, run `training_functions.py`
Note: `training_functions.py` performs a hyperparameter search. All fitted DL models are stored in the subfolders of `./data_modified/tensorboard_files_6`
5. Use `store_dl_pred.py` to find the best model and export the deep learning predictions

II. Build the baseline (boosted trees) model and our DL model: (administrator rights are necessary)

1. Create a conda environment using `environment.yml`.
Copy `environment.yml` to your home folder. Type in terminal:
 - (a) `conda env create -f environment.yml`
 - (b) `conda activate claims-management-joe`
 - (c) To install `xgboost`, **switch to administrator user** first and type in terminal
 - i. `/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`
 - ii. `brew install libomp`
 - (d) Now, you can **switch back to non-administrator user** and type in terminal:
 - (e) `conda activate claims-management-joe`
 - (f) `pip3 install xgboost`
 - (g) `spyder`

For more information about installing `xgboost` and `brew`, see
<https://xgboost.readthedocs.io/en/latest/build.html>
<https://brew.sh>
2. Change working directory in Spyder to the 'Code JoE' folder
3. To prepare data, run `loading_data.py`
4. To fit the boosted trees model, run `baseline_training_functions.py`
5. To fit the deep learning model, run `training_functions.py`
Note: `training_functions.py` performs a hyperparameter search. All fitted DL models are stored in the subfolders of `./data_modified/tensorboard_files_6`
6. Use `store_dl_pred.py` to find the best model and export the deep learning predictions

III. Generate main graphs:

1. Build our deep learning and the competitor model (see II.)
2. Graphs have been generated in Stata, see
 - (a) 0 - `load predictions.do`
 - (b) 1 - `graph Top Frauds.do`
 - (c) 2 - `graph Pi.do`

Note: “environment.yml” sets up a conda environment (named claims-management-joe) with the following content:

python==3.7; spyder==3.3.6; torch==1.7.1; tensorboard==2.1.0; scikit-optimize==0.7.1; pandas==0.25.1; tqdm==4.36.1; scikit-learn==0.21.3; and
the package `warmup-scheduler` (v0.3.2), via

```
pip install git+https://github.com/ildoonet/pytorch-gradual-warmup-lr.git
```

or use

```
pip install https://github.com/ildoonet/pytorch-gradual-warmup-lr/zipball/master
```

If anything goes wrong during the setup, you can delete the environment with
`conda remove --name claims-management-joe --all`