Description of Files in order as they should be accessed when training from scratch:

(1) Retweet Folder

- Note: The Google drive link for training and testing data is provided by the authors of the neural Hawkes process at: https://github.com/HMEIatJHU/neurawkes/tree/master/data

- "LSTM_construction.py" : the heart of our project - the implementation of the continuous time LSTM (CTLSTM) using pytorch - this file should always be present in the working directory where training and testing is being carried out.

- "data Loader-Retweet_train.ipynb" - contains code to convert data from the "train.pkl" file, containing training sequences into HDF5 file format.

- "RetweetTrainData.h5" - output of "data Loader-Retweet_train.ipynb" containing serialized training sequences and arrival times in HDF5 format.

- "Training-Retweet.ipynb" - contains code to train the CTLSTM network on the training sequences. First 4000 sequences were used for training.

- Folder "RetweetNets" - contains saved trained networks at every epoch.

- "data Loader-Retweet_test.ipynb" - contains code to load the "test.pkl" file and convert testing sequences into numpy arrays as before for training sequences.

- "RetweetTestData.h5" - the output of "data Loader-Retweet_test.ipynb".

- "Thinning_algo_testing-Retweet_Run1.ipynb" and "Thinning_algo_testing-Retweet_Run2.ipynb" - contain code to predict arrival time and event type on the testing sequences saved in the file above for two separate runs.
    - Since the thinning algorithm is stochastic and arrival times in this dataset span several orders of magnitude, it might take around an hour to predict arrival times and event types for test sequences (the first 1000 were considered for testing).

(2) Stackoverflow Folder

The stackoverflow folder contains files exactly analogous to the ones mentioned above (except "Retweet" in the file names is replaced with "SO"). They have the same sequence of operations and a copy of "LSTM_construction.py" is also provided in the folder.

Important Note:

- Stable accurate networks were obtained for the Retweet dataset at 40 epochs and for the Stackoverflow datasets and 28 epochs of training. Beyond this, instabilities were found to arise.