**AAAI 2025 Submission #XXXX Code and Data Instruction**

This is the code and data instruction for AAAI 2025 submission #XXXX titled

*Logic-based Quantitative Predictive Monitoring and Control for Safe Human-Machine Interaction Under Uncertainty*

This instruction is for reproducing all the experiment results step by step. The code and results of each step can be found in the corresponding subfolder. There is no need to change the hyper parameters in the code if you want to reproduce the results, as those are used in the experiments part (Section 5) of the paper.

* To generate results reported in Section 5.1 of the paper, please follow Steps 1-4.
* To generate results reported in Section 5.2 of the paper, please follow Step 5 after conducting Steps 1-4.
* To generate results reported in Section 5.3 of the paper, please follow Steps 6-7 after conducting Steps 1-5.
* All the code used for statistical tests (Step 8) and figure drawing (Step 9) are also included.

**Medical Case Study**

1. **Patient trace generation – Generate simulated patient traces using the simglucose simulator for model training**
   1. Install the original version of simglucose simulator

The code for original simulator is simglucose-master-original.zip. Set a path for this package as path\_simglucose.

cd '{path\_simglucose}/simglucose-master-original'

python setup.py install

* 1. Use the original version of simglucose simulator to generate patient trace data
     + 1. Code file name:

generate\_patient\_trace\_data\_with\_original\_simglucose\_simulator\_controller.py

* + - 1. Set path for saving patient trace data and simulation results at line 28, 29.
      2. Run simulation with following parameters (can be found at line 30-39):

# Simulation parameters:

# Start date: 2022-06-30

# Input simulation time (hr): 720h

# Random Scnenario

# Input simulation start time (hr): 0:00

# Select random seed for random scenario: 15

# Select the CGM sensor: Dexcom

# Select Random Seed for Sensor Noise: 10

# Select the insulin pump: Insulet

# Select controller: Basal-Bolus Controller

* 1. The final patient trace files used for the paper are in **results/patient\_trace\_data.zip**.

There are a total of 30 patients (10 children, 10 adolescents and 10 adults.)

These traces are used for LSTM model training, testing and validation in the following steps.

1. **LSTM Training – Preprocess the data, choose dropout params with the proposed loss function Lqt** 
   1. Three different models were trained for the three different patient populations (child, adolescent and adult). Select the file name (you can use any of them for testing the code):

lstm\_medical\_data\_with\_dropout\_adolescent.py

lstm\_medical\_data\_with\_dropout\_adult.py

lstm\_medical\_data\_with\_dropout\_child.py

* 1. Put STLU in a folder (code is in stlu\_monitor.zip). Set path for package STLU at line 29 of above code.
  2. Set path for loading patient trace data obtained in Step 1 at line 68.
  3. Set path for saving trained models, at line 72.
  4. Set path for saving mean/std values at line 211.
  5. Set path for saving all\_mean\_value.csv, all\_std\_value.csv at line 212, 213. Those files are for converting standardized data back to original scale.
  6. Set path for saving output metrics results at line 860.
  7. Run the code to see results.
  8. The final trained models for each patient type used in the paper are in **trained model.zip.** The prediction results for each patient type (metric value, overall results, **dropout parameter choice files**) are in corresponding folder. Files with prediction of each segment are not listed because of limited file size, but they will be generated after Step 2. The **all\_mean\_value.csv and all\_std\_value.csv** are also saved. They will be used in following steps.

1. **Choose dropout with baseline loss functions (Lsat, Lacc)**
   1. **Choose dropout type/rate with Lsat**
      * 1. Folder name: select\_dropout\_with\_lossfunc\_LSAT
        2. Code file name (you can use any of them for testing the code):

lstm\_medical\_data\_choose\_dropout\_lsat\_adolescent.py

lstm\_medical\_data\_choose\_dropout\_lsat\_adult.py

lstm\_medical\_data\_choose\_dropout\_lsat\_child.py

* + - 1. Set path for loading STLU package at line 29 (as in Step 2, b)).
      2. Set path for loading all patient trace data obtained in Step 1 at line 68.
      3. Set path for saving output results at line 921.
      4. Set path for loading trained models for each patient type get in Step 2 at line 929, 931, 933.
      5. Set path for loading data files used for standardizing dataset get from Step 2 (all\_std\_value.csv, all\_mean\_value.csv) at line 936, 937.
      6. Run the code to see results.
      7. The final results used in paper are in **results/choose\_dropout\_lsat\_results.zip**. Those files will be used in Step 4 and beyond.
  1. **Choose dropout type/rate with Lacc**

The procedures are the same as for Lsat in a).

* + - 1. Code file name (you can use any of them for testing the code):

lstm\_medical\_data\_choose\_dropout\_lacc\_adolescent.py

lstm\_medical\_data\_choose\_dropout\_lacc\_adult.py

lstm\_medical\_data\_choose\_dropout\_lacc\_child.py

* + - 1. Set path for loading STLU package at line 29 (as in Step 2, b)).
      2. Set path for loading all patient trace data get in Step 1 at line 68.
      3. Set path for saving output results at line 916.
      4. Set path for loading trained models for each patient type get in Step 2 at line 924, 926, 928.
      5. Set path for loading data files used for standardizing dataset (the same all\_std\_value.csv, all\_mean\_value.csv as in a)) at line 931, 932.
      6. Run the code to see results.
      7. The final results used in paper is in **results/choose\_dropout\_lacc\_results.zip**. Those files will be used in Step 4 and beyond.

1. **Predict test set with trained LSTM model with chosen dropout type/rate & evaluation**
   1. **Predict test set with dropout params chosen by Lqt**
      * 1. Code file name (you can use any of them for testing the code):

lstm\_medical\_predict\_testing\_set\_lqt\_adolescent.py

lstm\_medical\_predict\_testing\_set\_lqt\_adult.py

lstm\_medical\_predict\_testing\_set\_lqt\_child.py

* + - 1. Set path for loading STLU package at line 29 (as in Step 2, b)).
      2. Set path for loading all patient trace data get in Step 1 at line 68.
      3. Set path for saving output results at line 832.
      4. Set path for loading trained models for each patient type get in Step 2 at line 838, 840, 842.
      5. Set path for loading data files used for standardizing dataset (the same all\_std\_value.csv, all\_mean\_value.csv get in Step 2) at line 847, 848.
      6. Confirm the chosen dropout type and rate at line 844, 845 is consistent with results in **Step 2**.

For example, for **adult** patient, look in to the file named like **dropout\_choice\_result**\_train\_type\_4\_train\_rate\_0.9\_lr\_0.01\_e\_50.csv in **adult results folder** get in Step 2, and find the **smallest loss value** in the form. Then use the corresponding dropout type and rate in line 844, 845 for dropout\_rate\_dict and dropout\_type\_dict. The same for adolescent and child.

* + - 1. Run the code to see results (accuracy, F1, prediction of each sample in testing set).

Metric file name is named like: **metrics\_results**\_one\_side\_b\_1024\_seqlen\_20\_stepback\_10\_e\_1\_lr\_0.01\_f\_8\_conf\_0.95\_dt\_2\_dr\_0.8.csv in each patient type output folder.

Segment file with prediction on each sample in testing set is named like: **segment\_results\_**b\_1024\_seqlen\_20\_stepback\_10\_e\_1\_lr\_0.001\_f\_8\_conf\_0.95\_dt\_2\_dr\_0.9.csv in each patient type output folder.

Overall result file is named like results\_b\_1024\_seqlen\_20\_stepback\_10\_e\_1\_lr\_0.001\_f\_8\_conf\_0.95\_dt\_2\_dr\_0.9.csv in each patient type output folder.

* + - 1. The final segment\_results files used in paper are not included because of limited file size, but will be generated after this step. Other outputs are in the results folder.
  1. **Predict test set with dropout params chosen by Lsat**

The procedure is the same as in a). Except in step 7):

Confirm the chosen dropout type and rate at line 845, 846 is consistent with results in **Step 3, a)**.

For example, for **adult** patient, look in to the file **dropout\_choice\_result**\_train\_type\_4\_train\_rate\_0.9\_lr\_0.001\_e\_1.csv in adult result folder get in Step 3, a). Find the **smallest loss value** in the form. Then use the corresponding dropout type and rate in line 845, 846 for dropout\_rate\_dict and dropout\_type\_dict. The same for adolescent and child.

* 1. **Predict test set with dropout params chosen by Lacc**

The procedure is the same as in a). Except in step 7):

Confirm the chosen dropout type and rate at line 845, 846 is consistent with results in **Step 3, b)**.

For example, for **adult** patient, look in to the file **dropout\_choice\_result**\_train\_type\_4\_train\_rate\_0.9\_lr\_0.001\_e\_1.csv in adult result folder get in Step 3, a). Find the **smallest loss value** in the form. Then use the corresponding dropout type and rate in line 845, 846 for dropout\_rate\_dict and dropout\_type\_dict. The same for adolescent and child.

* 1. **For Table 1 results shown in the paper**

Please look at metric result file named like **metrics\_results\_**one\_side\_b\_1024\_seqlen\_20\_stepback\_10\_e\_1\_lr\_0.001\_f\_8\_conf\_0.95\_dt\_3\_dr\_0.5 for each type of patient in the results folder generated by a), b) and c), respectively.

1. **Pre-alert time evaluation**
   1. Code file name: calculate\_prediction\_metrics\_and\_pre\_alert\_time.py
   2. Set following path:
      * 1. Path for loading STLU at line 28.
        2. Path for loading prediction on testing set for each type of patient at line 311~314. The file loaded should be the **segment\_result files** generated from **Step 4, a), 8)**.
        3. Path for saving results at line 319, 518.
   3. Run the code to see results and save generated data files. Those files will be used in following steps.
   4. The results used in the paper are in folder results.
2. **Closed-loop simulation for 3 types of patients with proposed controller and baseline**
   1. Set path for STLU at line 17 in file sim\_engine.py in simglucose\_modified\simglucose\simulation
   2. Install modified version of simglucose package in simglucose\_modified.zip
   3. Set paths for baseline simulation in file: medical\_case\_pipeline\_new\_controller\_structure\_no\_lstm.py
      * 1. Path for STLU at line 29.
        2. Path for trained lstm models for adult, adolescent and child at line 647, 649, 651. The models are obtained from Step 2.
        3. Path for data files for standardizing dataset (the same all\_std\_value.csv, all\_mean\_value.csv as in Step 2) at line 656, 657.
        4. Path for saving simulation results at line 679.
   4. Run simulations with baseline controller for each type of patient and save the results.

Code file name:

no\_lstm\_adolescent.py

no\_lstm\_adult.py

no\_lstm\_child.py

* 1. Get all meal time and amount data of each patient from baseline results with file:

get\_meal\_time\_amount\_each\_patient.py

* + - 1. Set path for loading baseline simulation trace results at line 34. The files are obtained from d).
      2. Set path for saving result file **all\_meal\_time.csv** and **all\_meal\_amount.csv** at line 38, 40.
      3. Run code to save the 2 files. They will be used in following steps.
  1. Set following paths for proposed controller in medical\_case\_pipeline\_new\_controller\_structure\_eq\_2.py:
     + 1. Path for STLU at line 28
       2. Path for trained lstm models for adult, adolescent and child at line 658, 660, 662. The models are from Step 2.
       3. Path for data files for standardizing dataset (the same all\_std\_value.csv, all\_mean\_value.csv get in Step 2) at line 667, 668.
       4. Path for loading each meal time and meal amount for each patient (all\_meal\_amount.csv, all\_meal\_time.csv get in e)) at line 669, 670.
       5. Path for saving simulation results at line 693.
  2. Make sure the dropout types and rates in dropout\_type\_dict and dropout\_rate\_dict at line 664, 665 are the same chosen parameters as in **Step 4, a), 7)** for each type of patient.
  3. Run simulations with proposed control type for each type of patient and save the results.

Code file name:

batchRun\_adolescent\_1.py

batchRun\_adult\_1.py

batchRun\_child\_1.py

* 1. The results used in the paper are in the results folder:

Folder no\_lstm for baseline simulation results.

Folder adult, adolescent, child for proposed method simulation results.

Those files will be used in following steps.

1. **Closed-loop simulation evaluation**
   1. Code file name: calculate\_medical\_case\_pipeline\_controller\_metrics.py
   2. Set path for loading simulation results of baseline and proposed method at line 364. This folder should contain 4 sub-folders: no\_lstm, adult\_patient, child\_patient, adolescent\_patient. The first is baseline results, the other 3 are proposed controller results. They are obtained from Step 5.
   3. Set path for saving results at line 371, 430, 518.
   4. Run the code to get the results. The results contain the metric values in Table 3 in the paper.
   5. The results used in the paper are in folder results.
2. **Statistical test** 
   1. Code file name: statistical\_test.py
   2. For pre-alert time:
      * 1. Set path at line 18 to load the result of pre-alert time from Step 7, c). The files used in this step are named like real\_cgm\_trace\_with\_hazard\_lable\_{patient\_type}\_{control\_type}.csv.
        2. Set path at line 19, 61 to save statistical test results of pre-alert time.
   3. For Time in Range (TIR) and hazards number:
      * 1. Set path at line 92 to load result files of simulations obtained from Step 6, d). The files are named like controller\_metrics\_result\_{patient\_type}.csv and medical\_metrics\_results\_{patient\_type}.csv.
        2. Set path at line 93 to save statistical test results of Time-in-range and hazards number.
   4. Run the code to see results.
3. **Figure drawing**
   1. Code file name: figure\_drawing.py
   2. Draw Fig.5: compare CGM of baseline and proposed method for 3 patients.
      * 1. Set path at line 30, 42, 43, 48, 49, 53, 54 to load the closed-loop simulation traces of baseline and proposed method.
        2. Set path at line 31 to save the figure.
   3. Draw Fig.6: compare number of hazards.
      * 1. Set path at line 174~178 to load the controller metrics results for each type of patient obtained from Step 6, d), and save the figures.
   4. Draw Fig.4~8: compare loss function values with different dropout types and rates for each type of patient
      * 1. Set path at line 214~218 to load the results of each type of patient from Step 2. Example file name: dropout\_choice\_result\_train\_type\_4\_train\_rate\_0.9\_lr\_0.001\_e\_1.csv
   5. Run the code to see the figures.

**Driving Case Study**

1. **Vehicle trace generation – Generate simulated vehicle traces using the SafeBench simulator for model training**
   1. Install the original version of SafeBench with following link:

https://github.com/trust-ai/SafeBench

* 1. Replace the folder ‘safebench’ and ‘script’ with the provide folder in SafeBench\_1
  2. Generate vehicle trace data
     + 1. Run following code at safebench\predictive\_monitor\_trace:

run\_multiple\_times.sh

Set i in range [1, 15], this is the seed for generating traces;

Set --if\_new\_controller to 0;

Change the file path in the script.

* 1. The final vehicle trace files used for the paper are in **results/vehicle\_trace\_data.zip**.

There are 2 examples for each behavior type.

These traces are used for LSTM model training, testing and validation in the following steps.

1. **LSTM Training – Preprocess the data, choose dropout params with the proposed loss function Lqt** 
   1. 2 different models were trained for the 2 different behavior types (cautious, aggressive). Select the file name (you can use any of them for testing the code):

lstm\_driving\_data\_with\_dropout\_Lqt\_behavior\_0.py

lstm\_driving\_data\_with\_dropout\_Lqt\_behavior\_2.py

* 1. Put STLU in a folder (code is in stlu\_monitor.zip). Set path for package STLU.
  2. Set following path in the file:

path for loadingtrace data obtained in Step 1,

path for saving trained models,

path for saving mean/std values,

path for saving all\_mean\_value.csv, all\_std\_value.csv. Those files are for converting standardized data back to original scale.

path for saving output metrics results.

* 1. Run the code to see results.
  2. The final trained models for each behavior type used in the paper are in **trained model.zip.** The prediction results for each behavior type (metric value, overall results, **dropout parameter choice files**) are in corresponding folder. Files with prediction of each segment are not listed because of limited file size, but they will be generated after Step 2. The **all\_mean\_value.csv and all\_std\_value.csv** are also saved. They will be used in following steps.

1. **Choose dropout with baseline loss functions (Lsat, Lacc)**
   1. **Choose dropout type/rate with Lsat**
      * 1. Folder name: select\_dropout\_with\_lossfunc\_LSAT
        2. Code file name (you can use any of them for testing the code):

lstm\_driving\_data\_choose\_dropout\_lsat\_behavior\_0.py

lstm\_driving\_data\_choose\_dropout\_lsat\_behavior\_2.py

* + - 1. Set following path before running the code:

Path for loading STLU packag.

Path for loading all vehicle trace data obtained in Step 1.

Path for saving output results.

Path for loading trained models for each behavior type get in Step 2.

Path for loading data files used for standardizing dataset get from Step 2 (all\_std\_value.csv, all\_mean\_value.csv).

* + - 1. Run the code to see results.
      2. The final results used in paper are in **select\_dropout\_with\_lossfunc\_LSAT**. Those files will be used in Step 4 and beyond.
  1. **Choose dropout type/rate with Lacc**

The procedures are the same as for Lsat in a).

* + - 1. Code file name (you can use any of them for testing the code):

lstm\_driving\_data\_choose\_dropout\_lacc\_behavior\_0.py

lstm\_driving\_data\_choose\_dropout\_lacc\_behavior\_2.py

* + - 1. Set path before running the code.
      2. Run the code to see results.
      3. The final results used in paper is in **select\_dropout\_with\_lossfunc\_LACC**. Those files will be used in Step 4 and beyond.

1. **Predict test set with trained LSTM model with chosen dropout type/rate & evaluation**
   1. **Predict test set with dropout params chosen by Lqt**
      * 1. Code file name (you can use any of them for testing the code):

lstm\_driving\_data\_predict\_testing\_set\_Lqt\_behavior\_0.py

lstm\_driving\_data\_predict\_testing\_set\_Lqt\_behavior\_2.py

* + - 1. Set following path before run the code:

path for loading STLU package.

path for loading all vehicle trace data get in Step 1.

path for saving output results.

path for loading trained models for each behavior type get in Step 2.

path for loading data files used for standardizing dataset (the same all\_std\_value.csv, all\_mean\_value.csv get in Step 2).

* + - 1. Confirm the chosen dropout type and rate at line 844, 845 is consistent with results in **Step 2**.
      2. Run the code to see results (accuracy, F1, prediction of each sample in testing set).
      3. The final segment\_results files used in paper are not included because of limited file size, but will be generated after this step. Other outputs are in the results folder.
  1. **Predict test set with dropout params chosen by Lsat**

The procedure is the same as in a). Except in step 7):

Confirm the chosen dropout type and rate is consistent with results in **Step 3, a)**.

* 1. **Predict test set with dropout params chosen by Lacc**

The procedure is the same as in a). Except in step 7):

Confirm the chosen dropout type and rate is consistent with results in **Step 3, b)**.

* 1. **For Table 1 results shown in the paper**

Please look at metric result file named like **metrics\_results\_one\_side\_b\_1024\_seqlen\_50\_stepback\_30\_e\_1\_lr\_0.001\_f\_17\_conf\_0.95\_dt\_4\_dr\_0.6.csv** for each type of behavior in the results folder generated by a), b) and c), respectively.

1. **Pre-alert time evaluation**
   1. Code file name: calculate\_prediction\_metrics\_and\_pre\_alert\_time.py
   2. Set following path:
      * 1. Path for loading STLU.
        2. Path for loading prediction on testing set for each behavior type. The file loaded should be the **segment\_result files** generated from **Step 4, a), 8)**.
        3. Path for saving results.
   3. Run the code to see results and save generated data files. Those files will be used in following steps.
   4. The results used in the paper are in folder results.
2. **Closed-loop simulation for 2 behavior types with proposed controller and baseline**
   1. Run following code at safebench\predictive\_monitor\_trace:

run\_multiple\_times.sh

Set i in range [0, 3], this is the seed for different trials;

Set --if\_new\_controller 1 --control\_type lstm\_with\_monitor to run proposed method.

Set --if\_new\_controller 1 --control\_type no\_lstm to run baseline method.

The results used in the paper are in the results folder:

Folder no\_lstm for baseline simulation results.

Folder lstm\_with\_monitor for proposed method simulation results.

Those files will be used in following steps.

1. **Closed-loop simulation evaluation**
   1. Code file name: metric\_calcualtion\_close\_loop\_driving\_case.py
   2. Set path for loading simulation results of baseline and proposed metho.They are obtained from Step 5.
   3. Set path for saving results.
   4. Run the code to get the results. The results contain the metric values in Table 3 in the paper.
   5. The results used in the paper are in the same folder.
2. **Figure drawing**
   1. Code file name: figure\_drawing\_driving\_case.py
   2. Set path to files obtained in steps above. They have a similar name as in the examples in the code.
   3. Run the code to see the figures.