

ECE 209AS Fall 2020 Bake-off 2 Mid Term Demo, 11/23/20

---

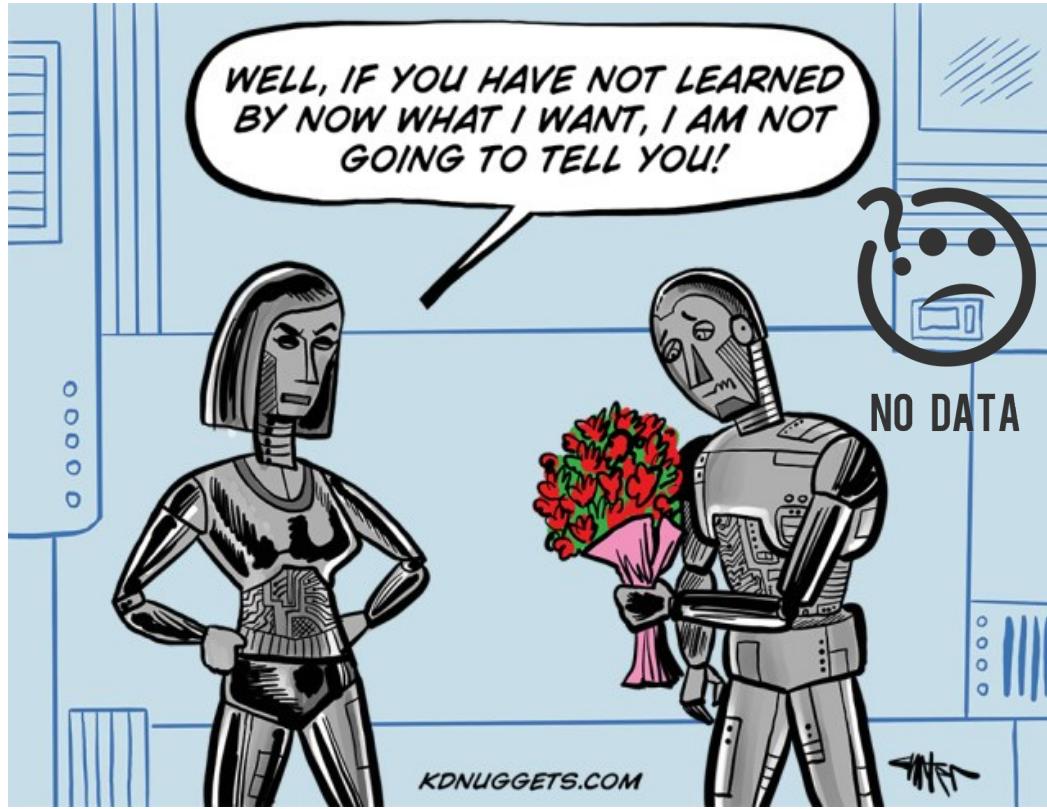
**GUI-GAN:** Towards an interactive graphical framework for privacy-preserving artificial data synthesis and imputation using generative adversarial networks.

---

**Viacheslav Inderiakin** and **Swapnil Sayan Saha**  
Dept. of ECE, UCLA

# Problem Statement

Hurdles in deploying AI-enabled interactive systems:



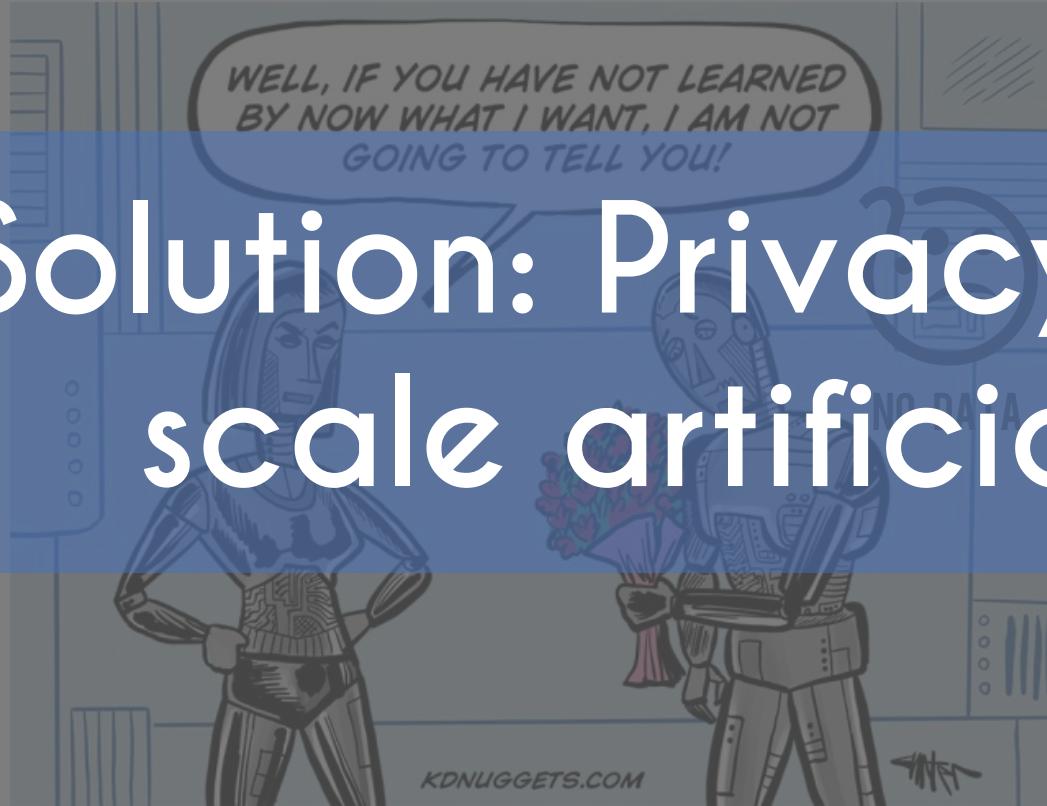
Lack of Data



Data Privacy

# Problem Statement

Hurdles in deploying AI-enabled interactive systems:



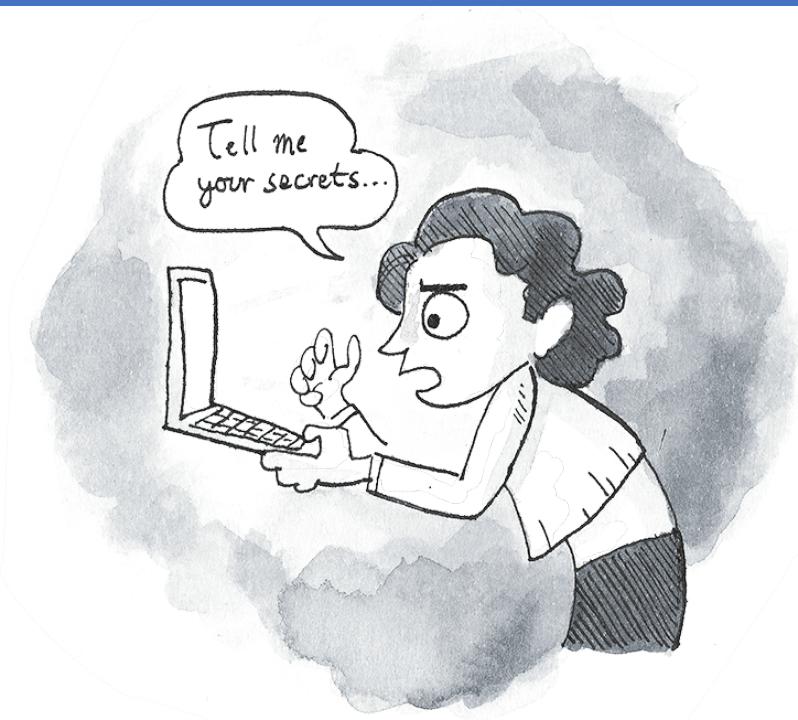
Solution: Privacy preserving large-scale artificial data synthesis

Lack of Data



Data Privacy

# Existing Methods



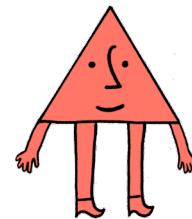
Lack of interactivity  
and control

Domain and  
application-specific

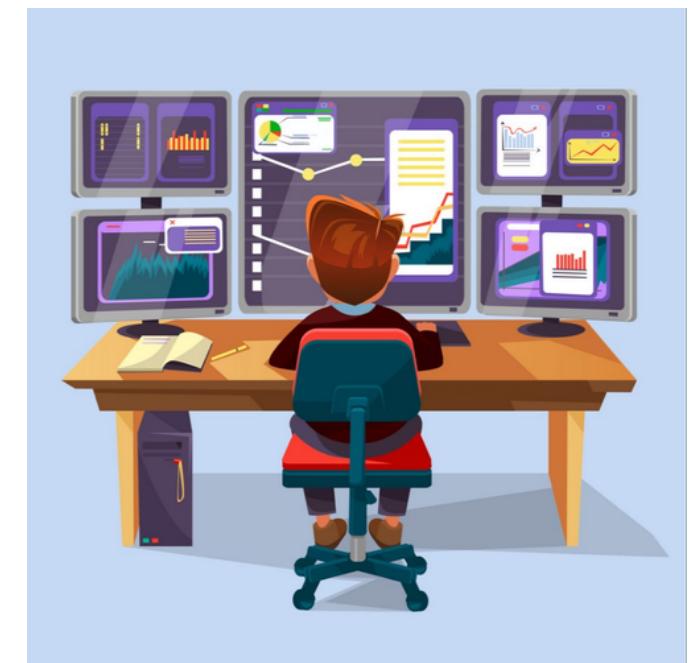
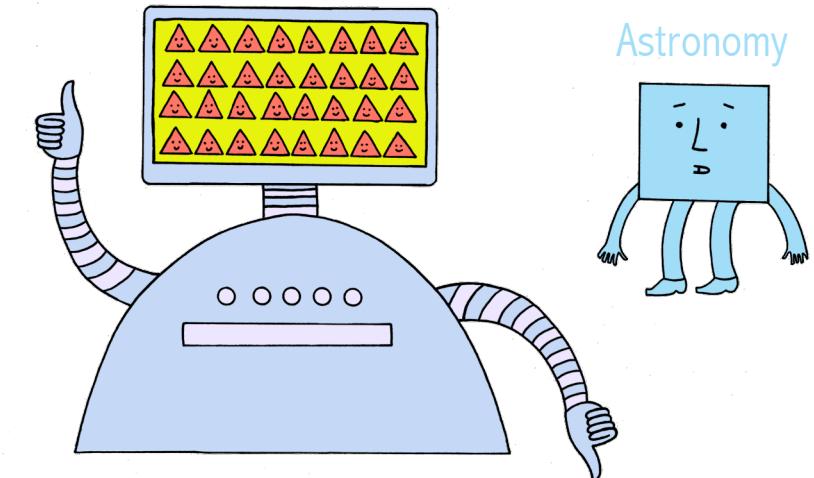
Challenges in deploying  
existing artificial data  
synthesizers

Computational and  
domain expertise required

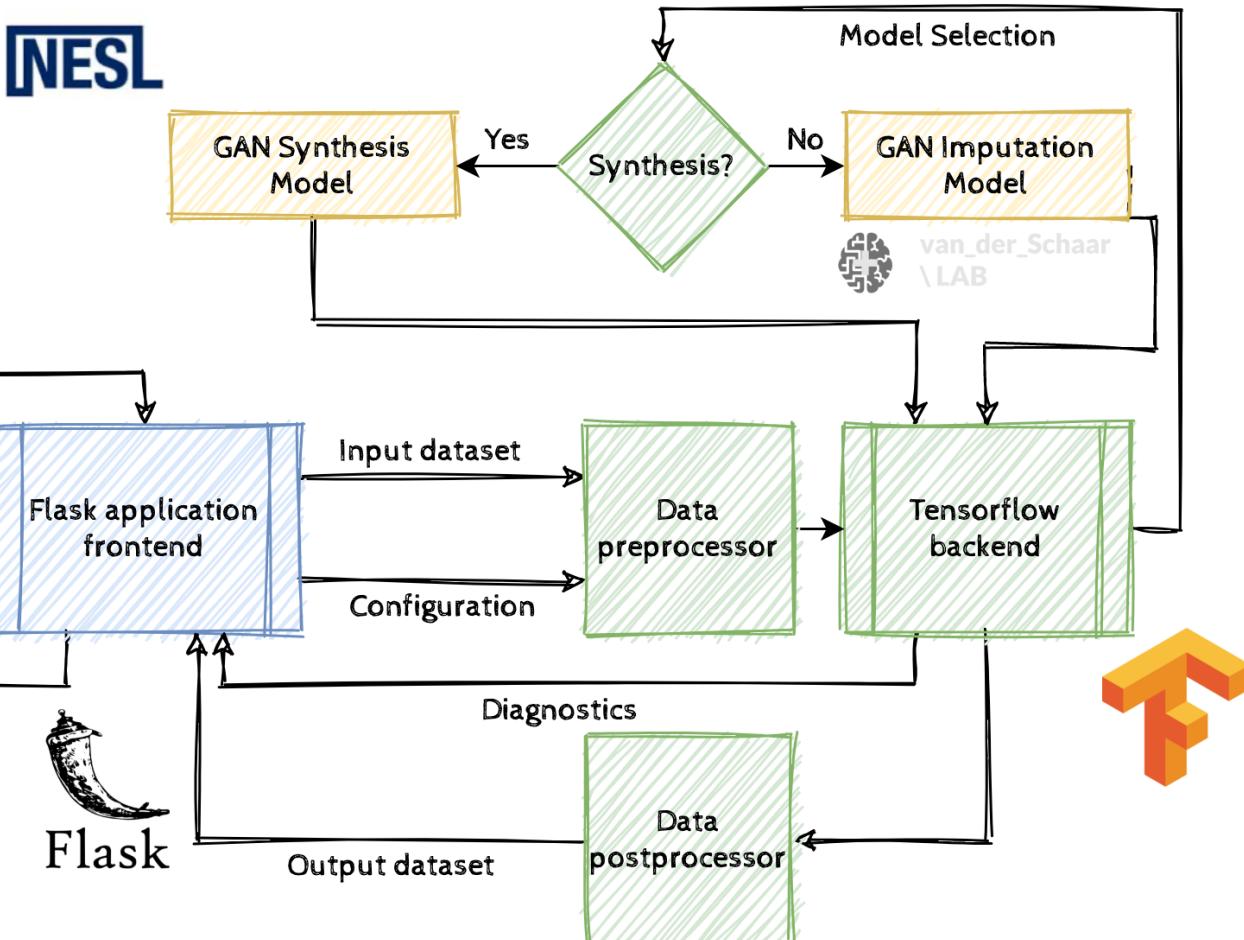
Medicine



Astronomy



# Proposed Solution



**GUI-GAN: A highly generalizable and AI-enabled synthetic time-series processing framework**

Generates privacy preserving annotated multi-class datasets from small time-series datasets in any domain

Collaborative and controllable: allows user to make graph-based and symbolic corrections to groups of generated datasets

Aimed for non-experts: assumes zero machine-learning and coding expertise; runs on any generic computing device

# Storyboard (Synthesis)

127.0.0.1:5000

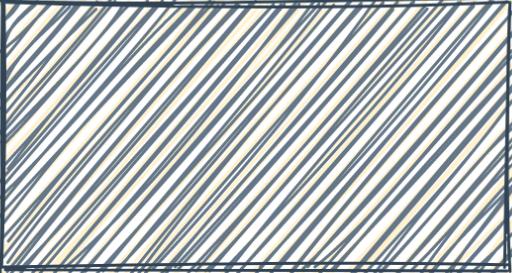
### Generation

Choose file   Load dataset  
Generator hyperparams.  
Generate   Save

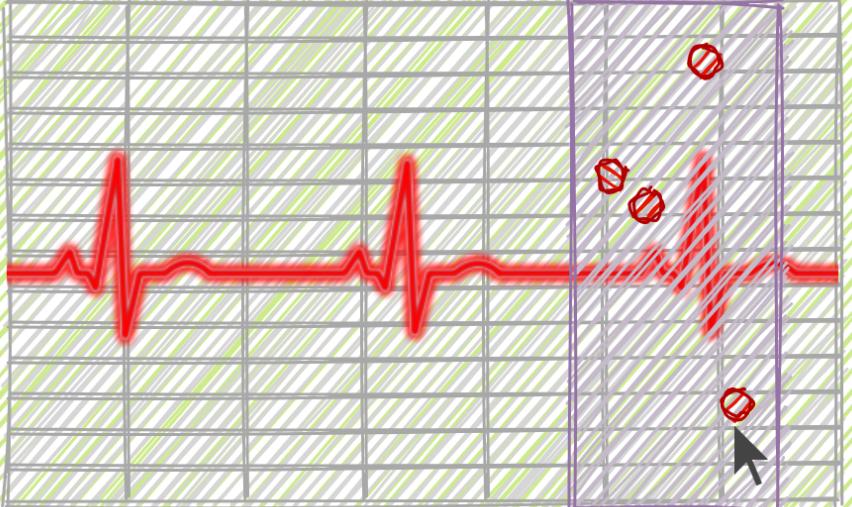
### Correction

Range   Ref. Points  
Corrector hyperparams.  
Impute   Save

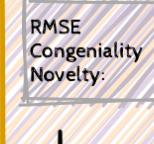
### Execution Log



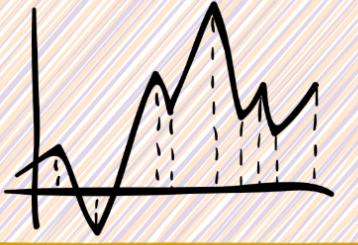
«« Original   Sample no.   Synthesized   »»



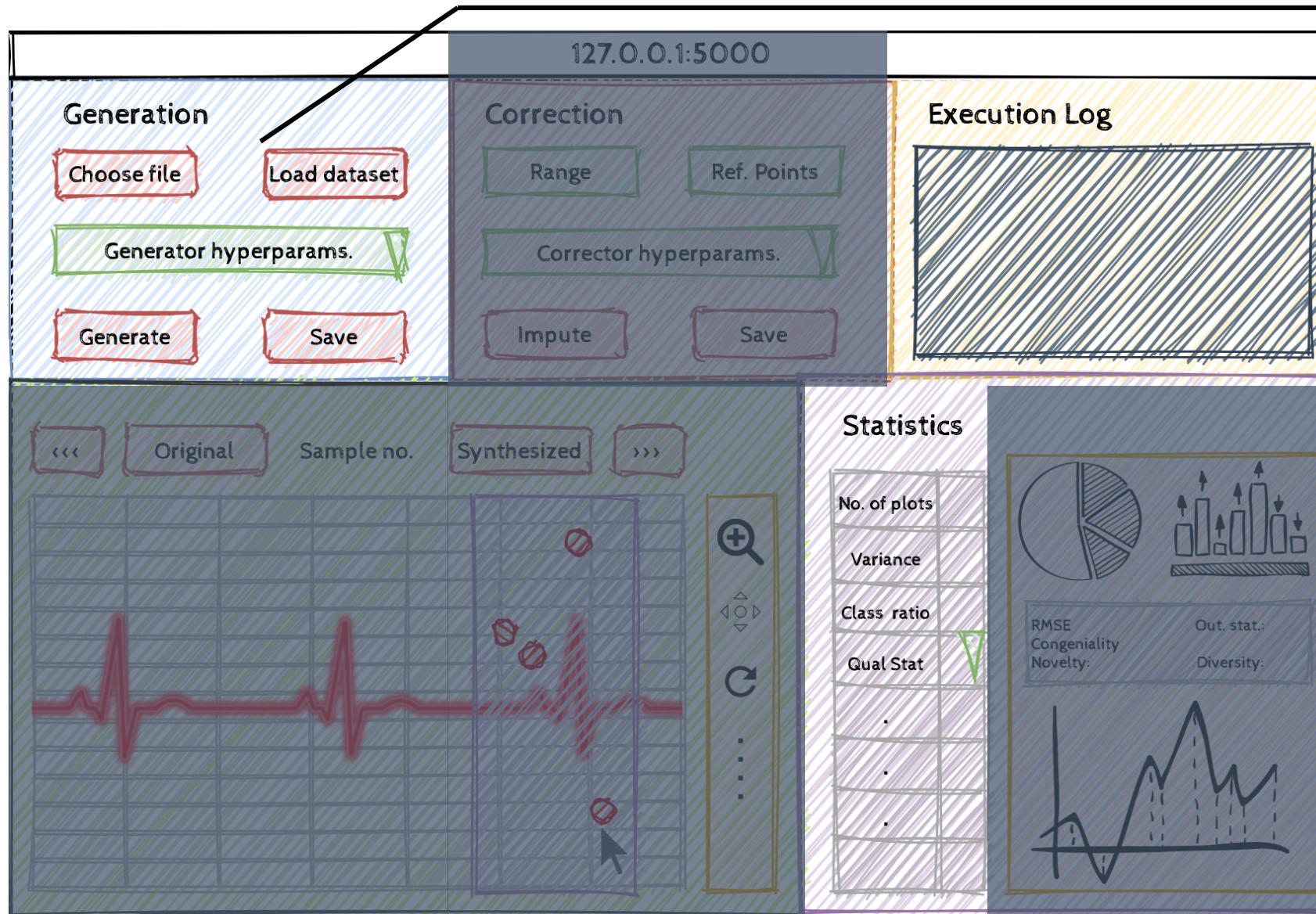
### Statistics

No. of plots	
Variance	
Class ratio	
Qual Stat	
.	
.	
.	

RMSE   Out. stat.:  
Congeniality   Diversity:  
Novelty:



# Storyboard (Synthesis)



Choose file to specify directory of small dataset as zip file

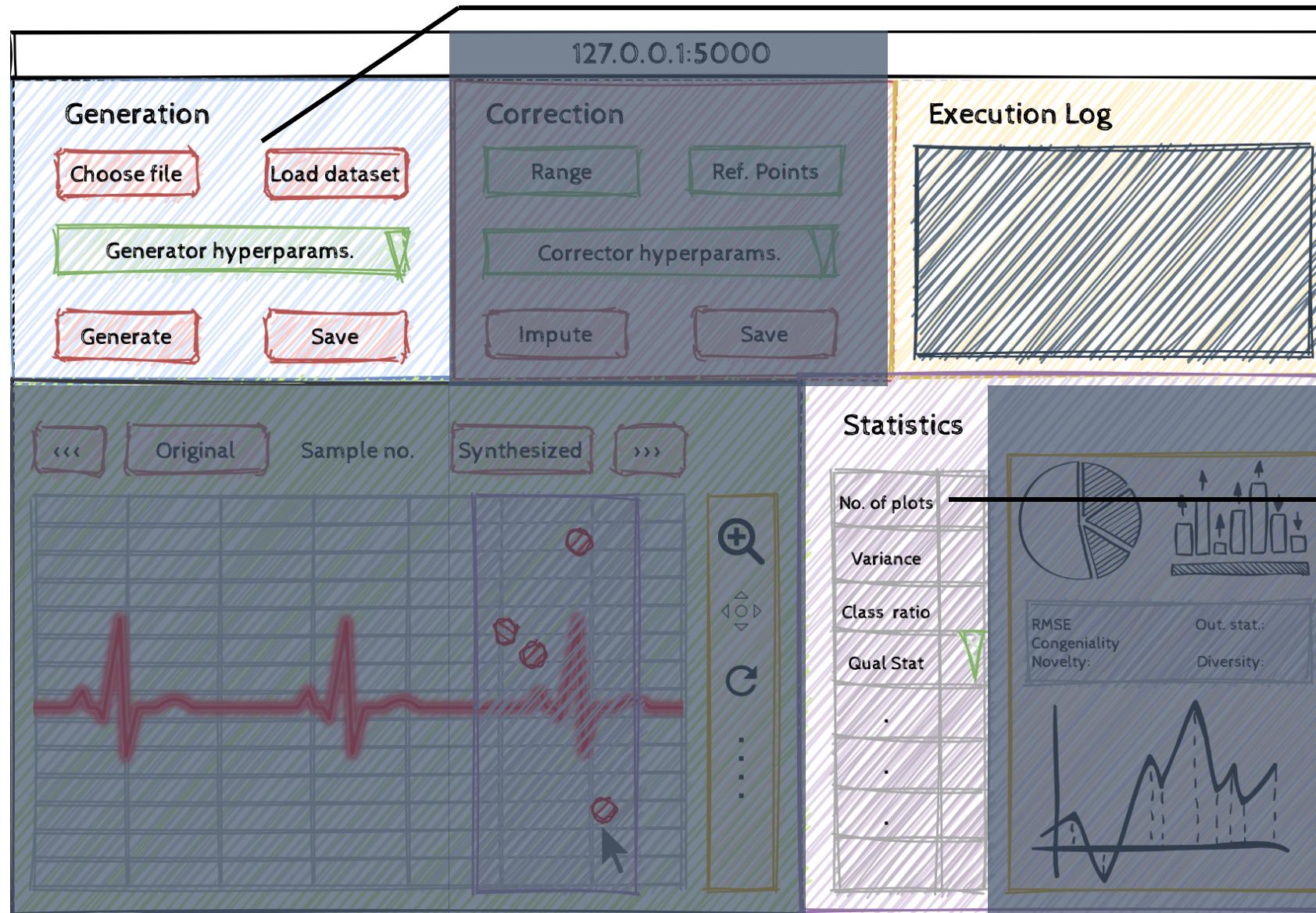
Click load dataset to load data into memory

Specify generator hyperparameters (listed in non-technical language with explanation)

Input coveted dataset statistics symbolically in the table

Click generate and wait for model to train and synthesize with training log shown in execution

# Storyboard (Synthesis)



Choose file to specify directory of small dataset as zip file

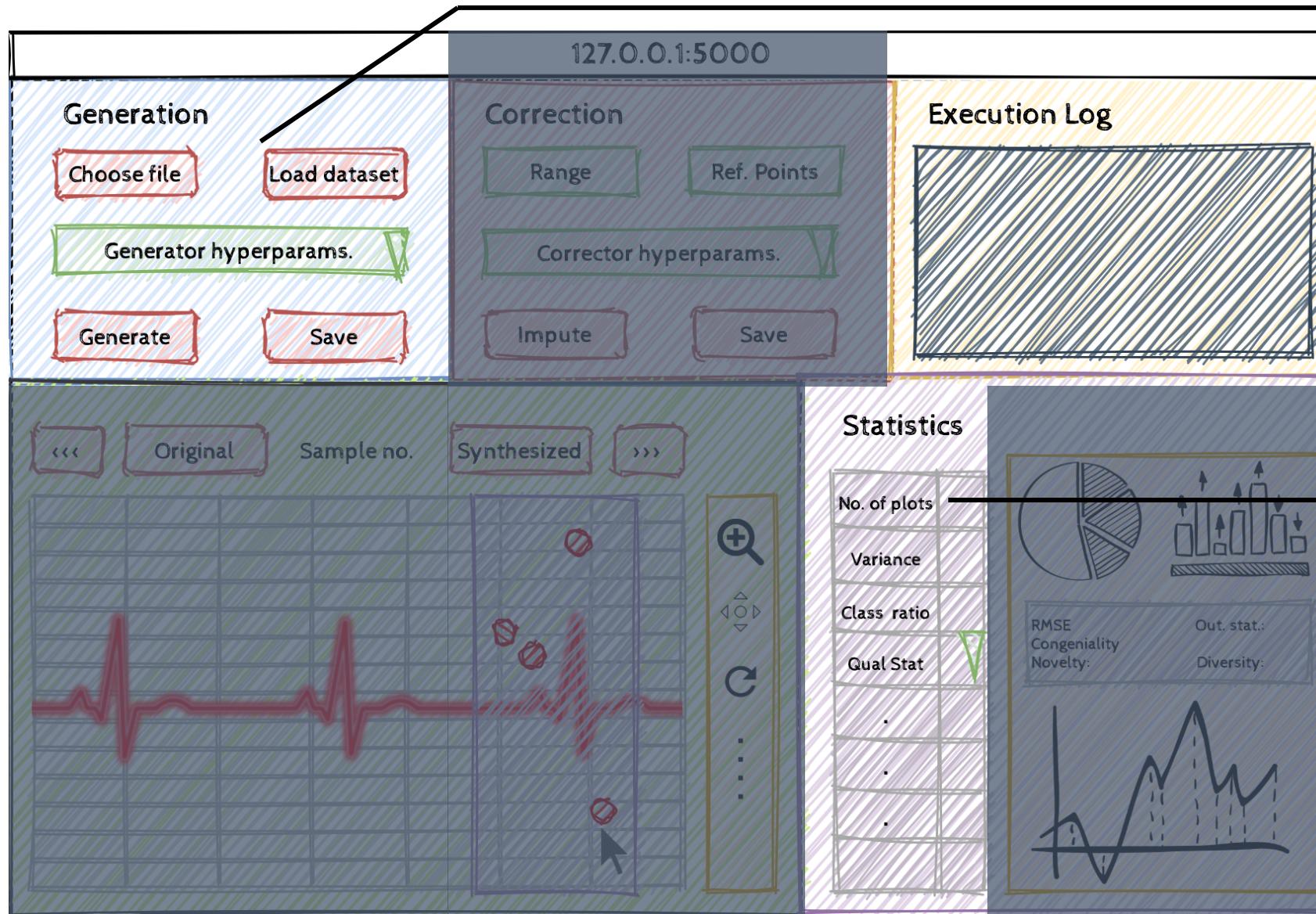
Click load dataset to load data into memory

Specify generator hyperparameters (listed in non-technical language with explanation)

Input coveted dataset statistics symbolically in the table

Click generate and wait for model to train and synthesize with training log shown in execution

# Storyboard (Synthesis)



Choose file to specify directory of small dataset as zip file

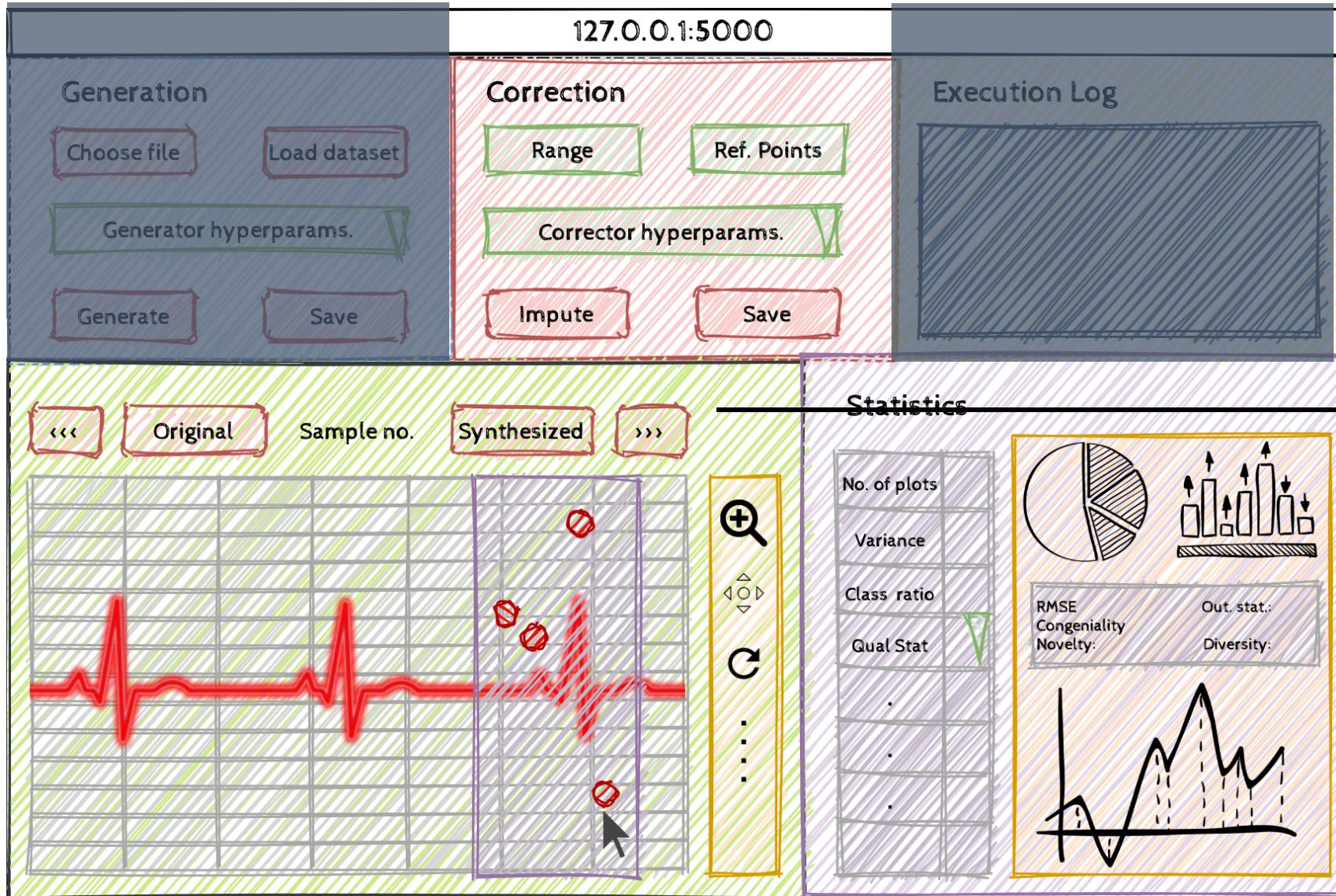
Click load dataset to load data into memory

Specify generator hyperparameters (listed in non-technical language with explanation)

Input coveted dataset statistics symbolically in the table

Click generate and wait for model to train and synthesize with training log shown in execution

# Storyboard (Correction)



Check plots in the graphical window (use arrow keys to move between plots), check generated dataset statistical plots in statistics window

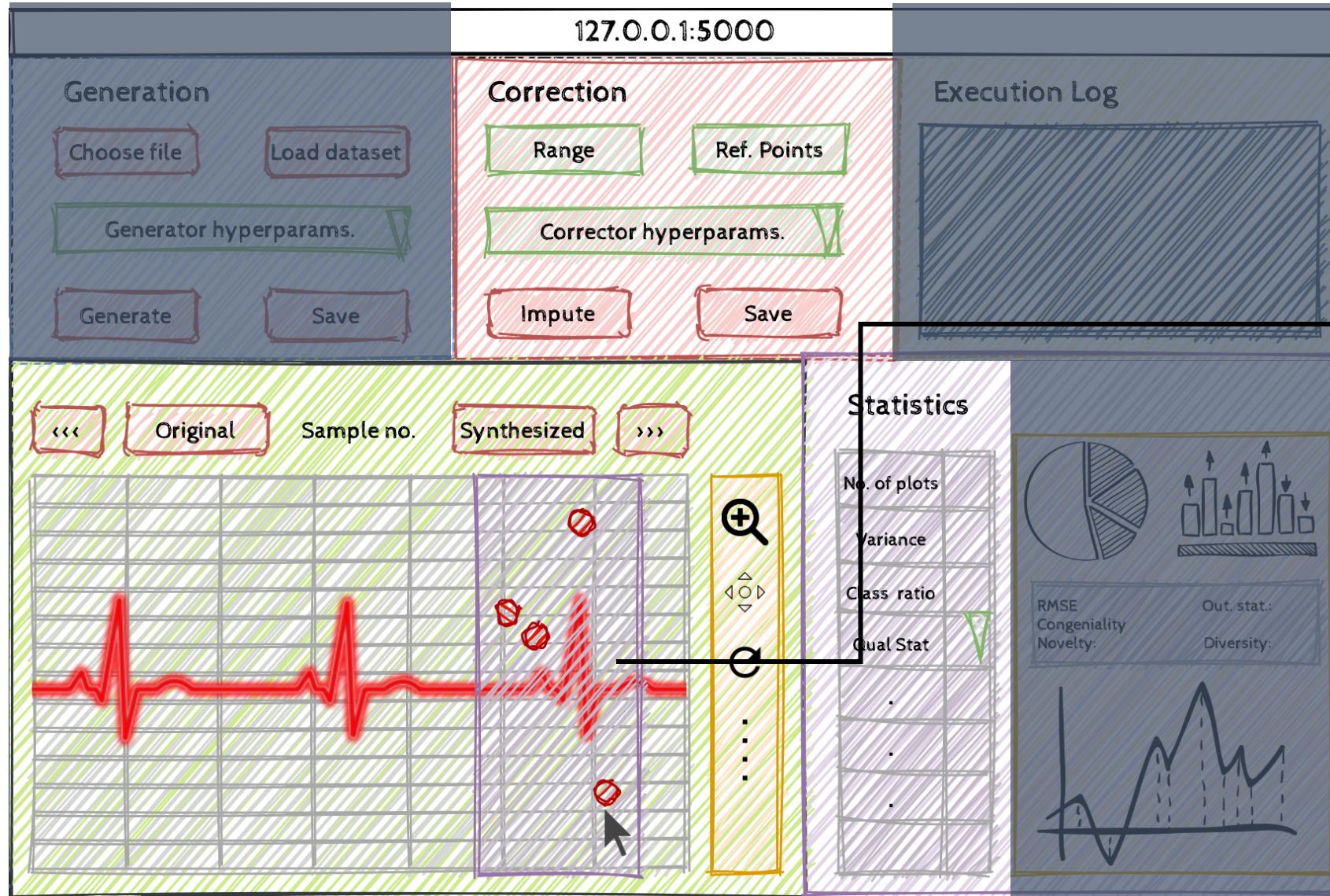
Specify correction by drawing bounding box and clicking guide points

Input imputation parameters

Input coveted dataset statistics in the table

Click **impute** and wait for model to train and synthesize with training log shown in execution

# Storyboard (Correction)



Check plots in the graphical window (use arrow keys to move between plots), check generated dataset statistical plots in statistics window

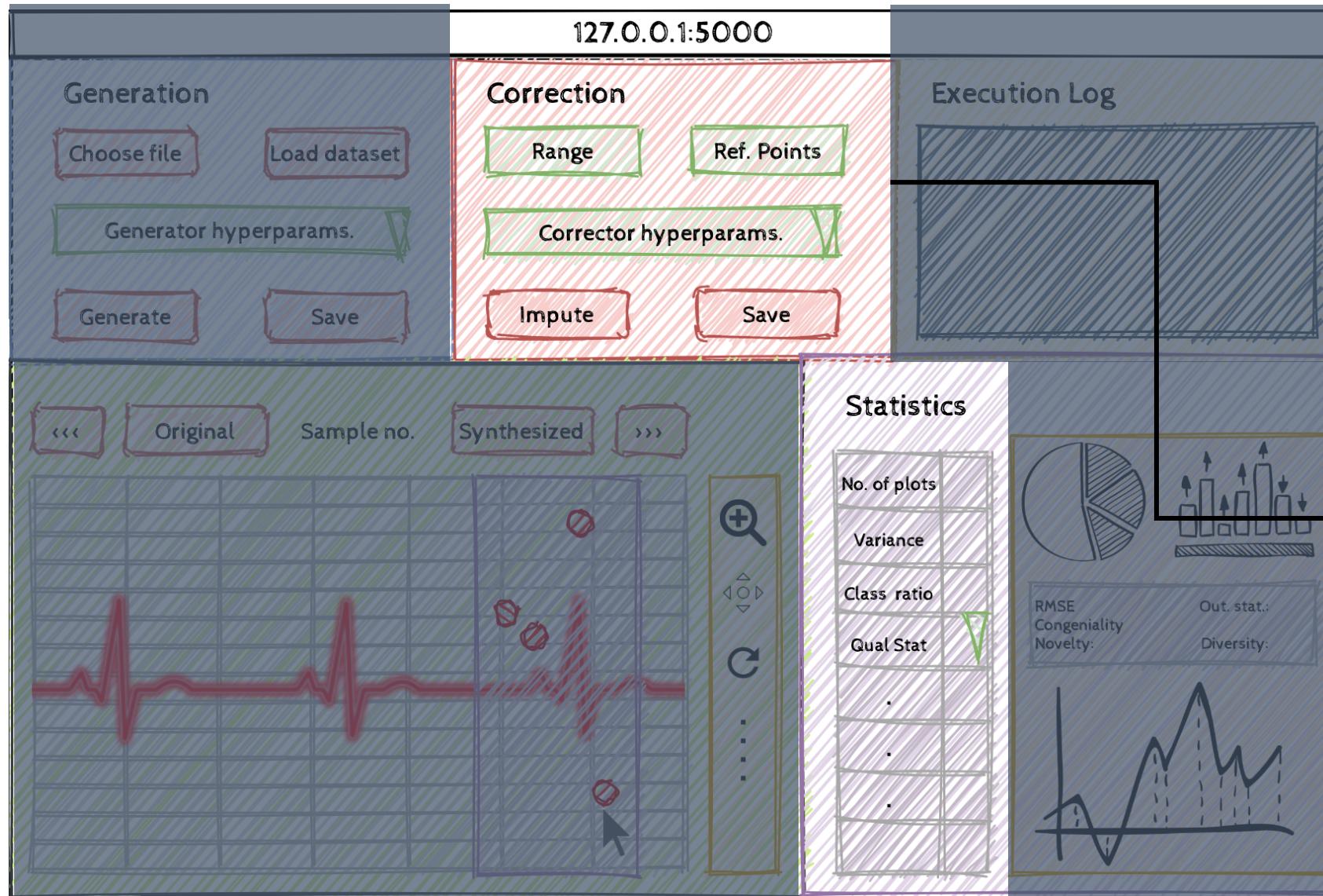
Specify correction by drawing bounding box and clicking guide points

Input imputation parameters

Input coveted dataset statistics in the table

Click **impute** and wait for model to train and synthesize with training log shown in execution

# Storyboard (Correction)



Check plots in the graphical window (use arrow keys to move between plots), check generated dataset statistical plots in statistics window

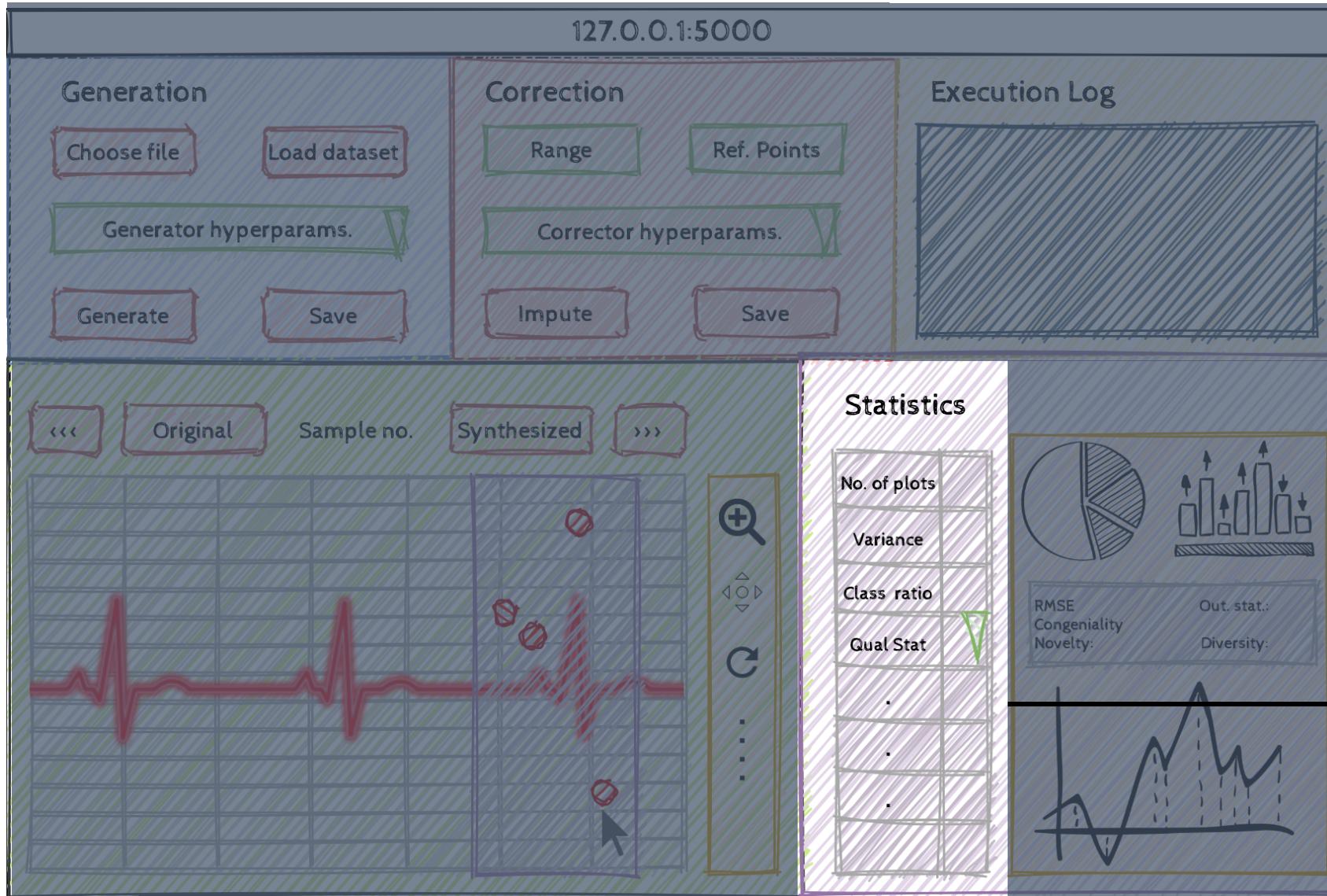
Specify correction by drawing bounding box and clicking guide points

Input imputation parameters

Input coveted dataset statistics in the table

Click **impute** and wait for model to train and synthesize with training log shown in execution

# Storyboard (Correction)



Check plots in the graphical window (use arrow keys to move between plots), check generated dataset statistical plots in statistics window

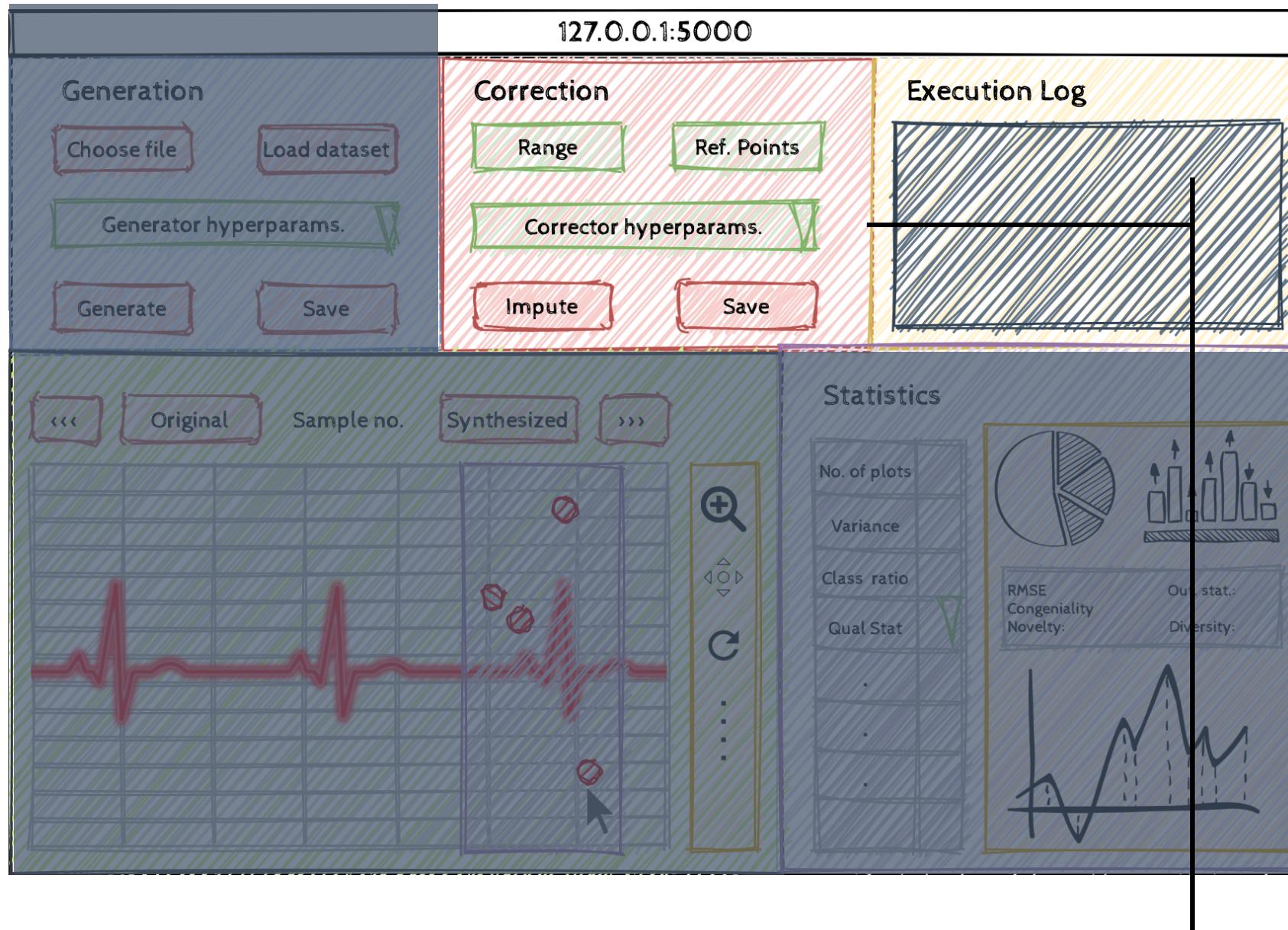
Specify correction by drawing bounding box and clicking guide points

Input imputation parameters

Input coveted dataset statistics in the table

Click **impute** and wait for model to train and synthesize with training log shown in execution

# Storyboard (Correction)



Check plots in the graphical window (use arrow keys to move between plots), check generated dataset statistical plots in statistics window

Specify correction by drawing bounding box and clicking guide points

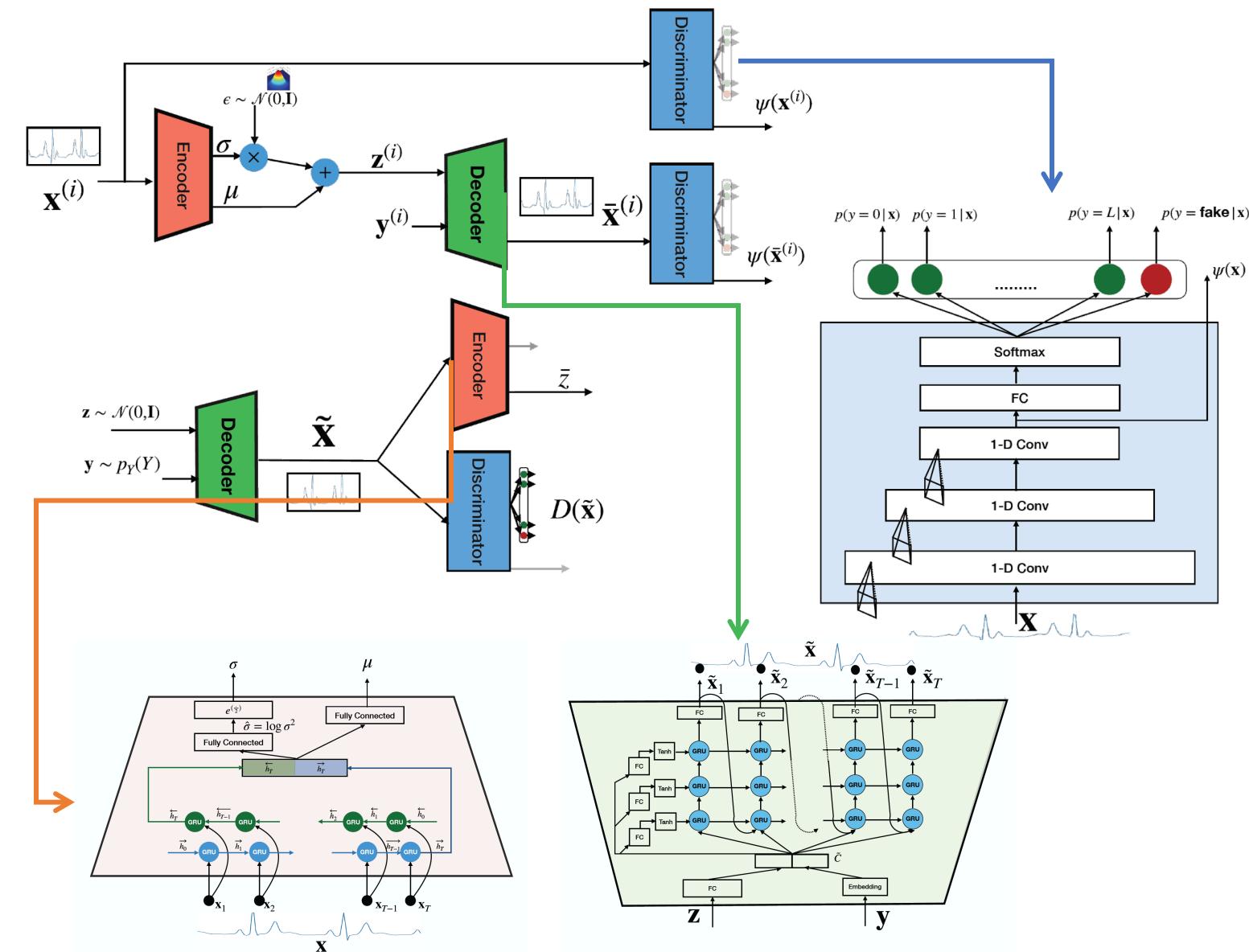
Input imputation parameters

Input coveted dataset statistics in the table

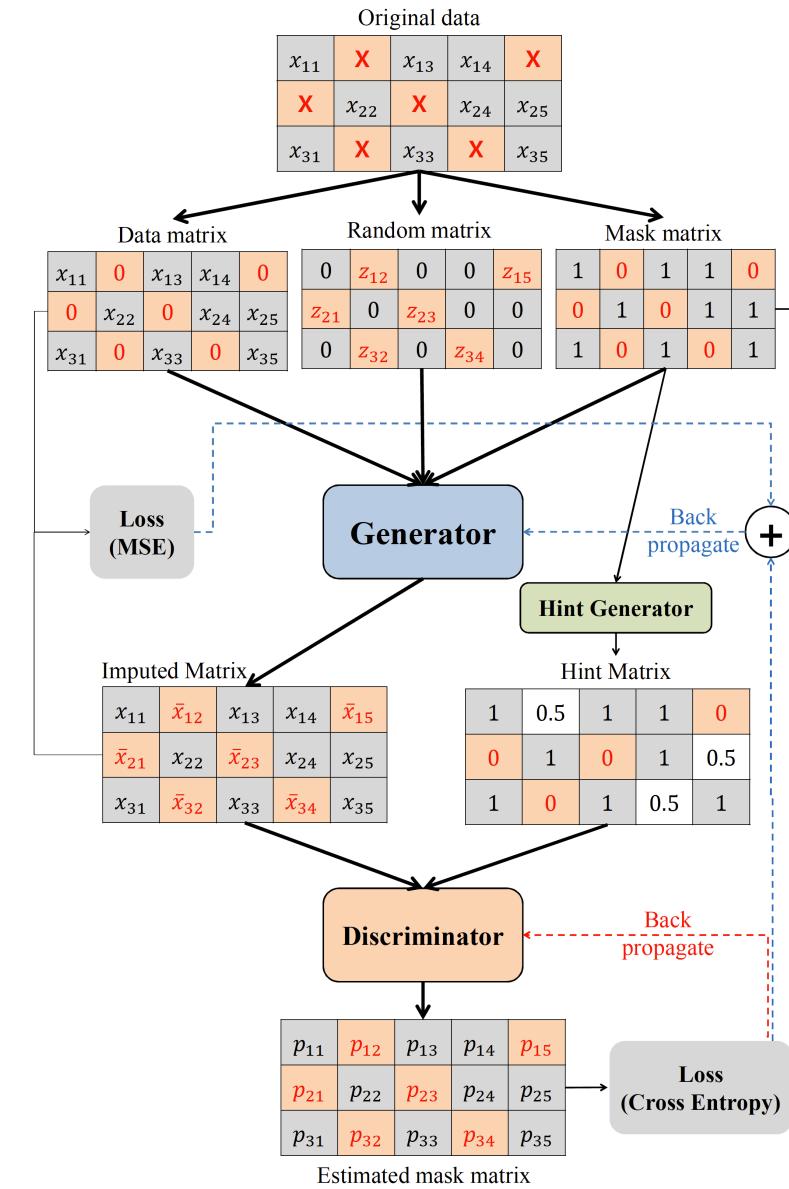
Click **impute** and wait for model to train and synthesize with training log shown in execution

# GAN-framework

## Dataset synthesizer (PhysioGAN)



## Dataset corrector (GAIN)

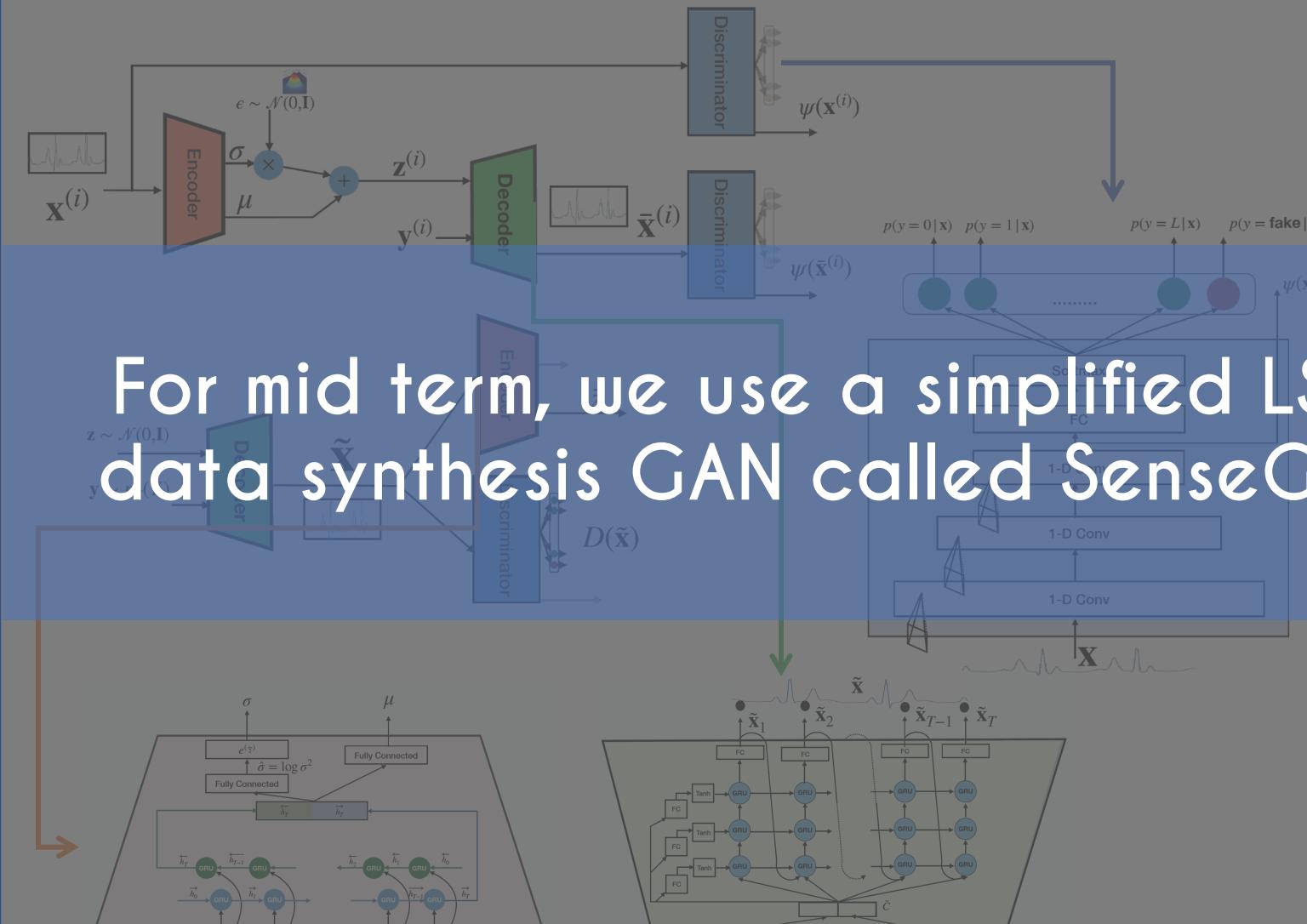


[1]. Alzantot, Moustafa Farid Taha Mohammed. "Secure and Private Machine Learning for Smart Devices". Ph.D. Diss. UCLA, 2019.

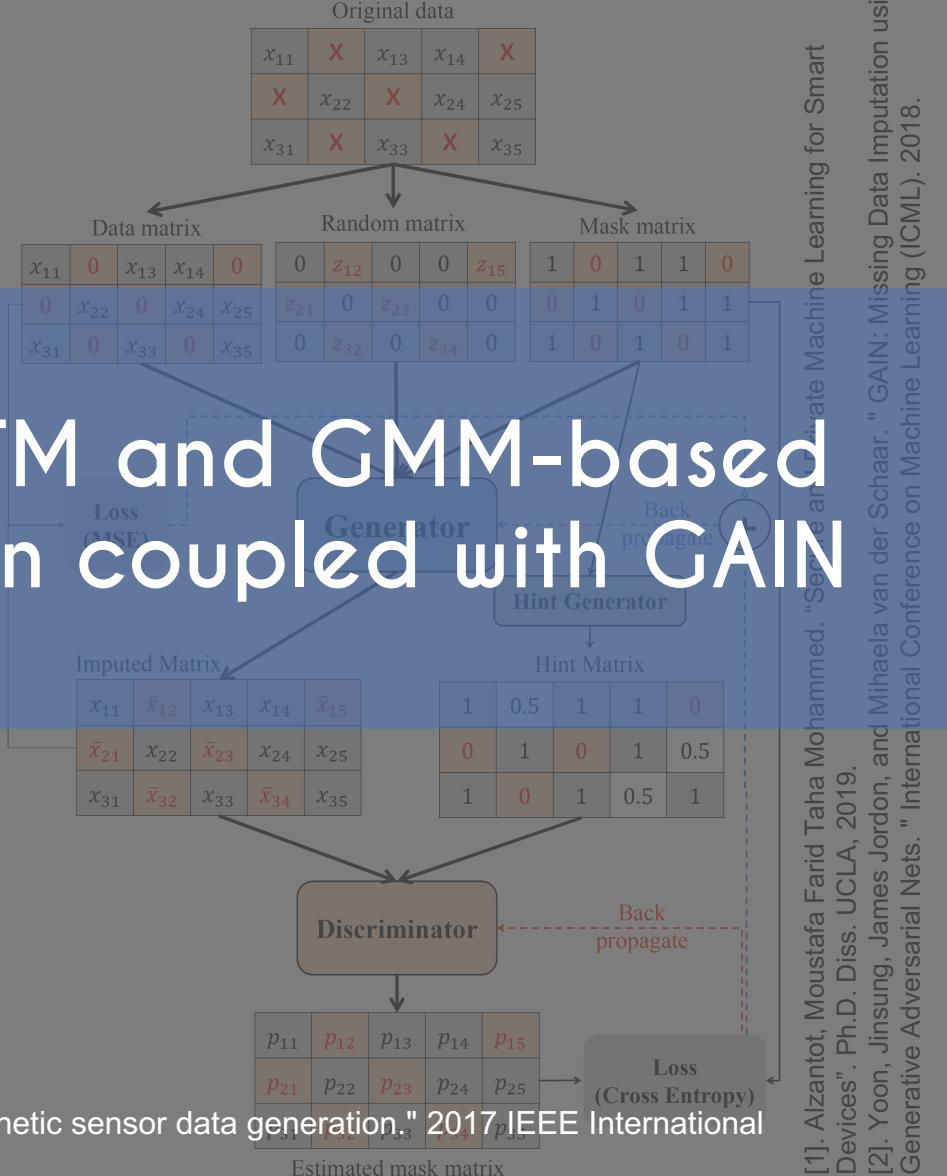
[2]. Yoon, Jinsung, James Jordon, and Mihaela van der Schaar. "GAIN: Missing Data Imputation using Generative Adversarial Nets." International Conference on Machine Learning (ICML). 2018.

# GAN-framework

## Dataset synthesizer (PhysioGAN)



## Dataset corrector (GAIN)



# Demo

Chrome File Edit View History Bookmarks People Tab Window Help 0 KB/s 22 100% Sun Nov 22 8:20 PM

Home 127.0.0.1:5000 Apps Favorites Other Bookmarks

## Interactive Data Generator

<https://youtu.be/vt53jW436O8>

Choose File No file chosen load\_file

Range 0-0

Epochs 10 Samples 5 generate

Reference Points impute

Execution log

save

<<< original Sample 0 synthesized >>>

original data

x

y

Temp.

Original data

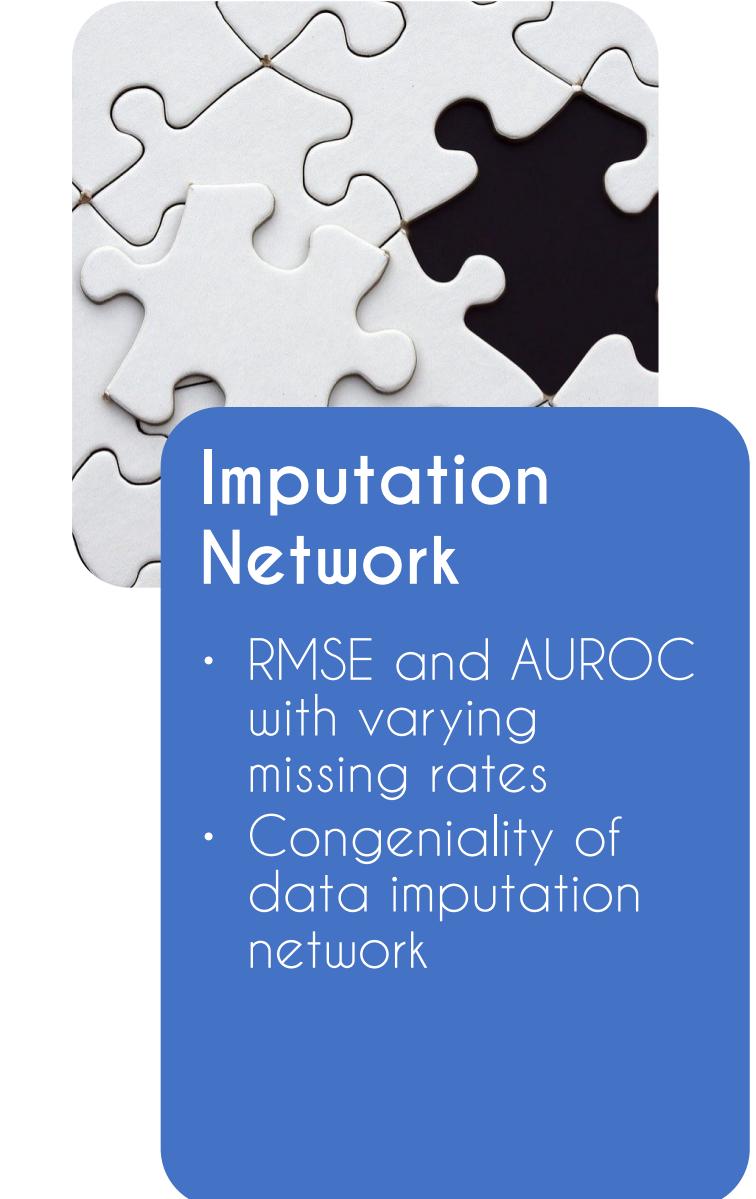
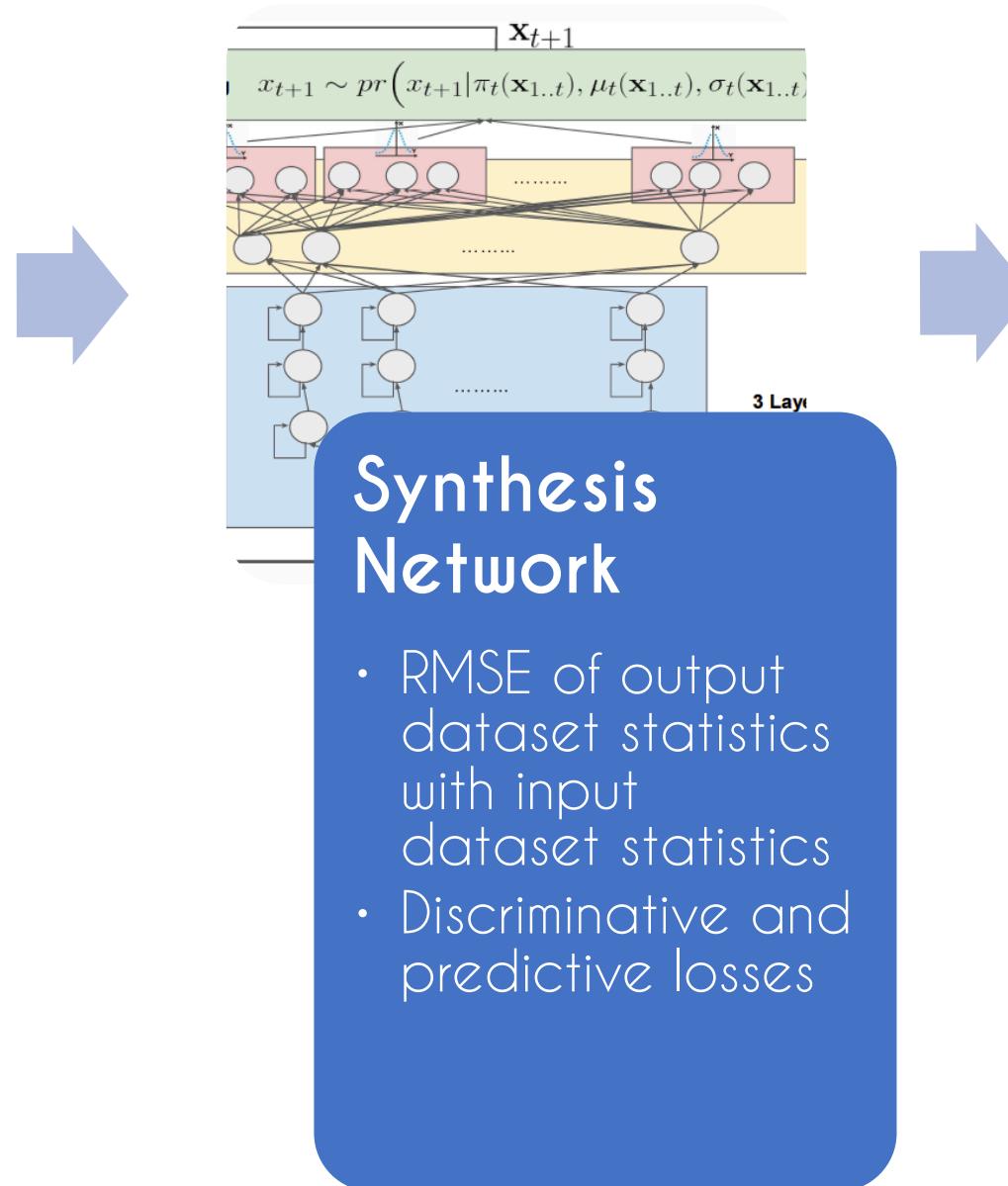
Sample 0

Synthesized

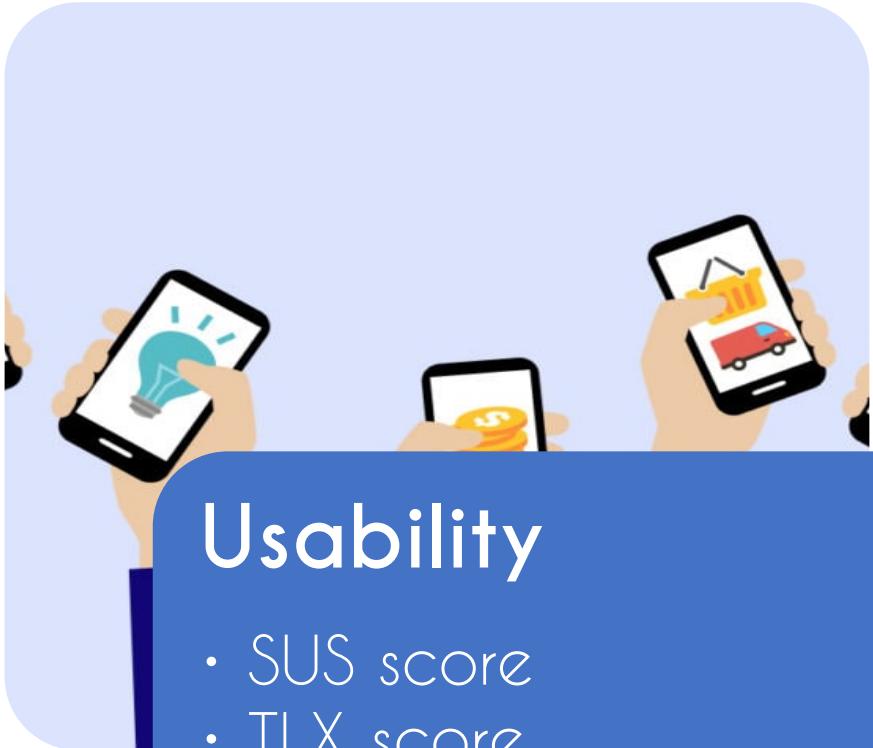
<<< original >>>

Nov 1 22 27,410 EAGLE MATLAB Word OneNote Mail Phone MATLAB Python R QuickTime

# Evaluation Metrics

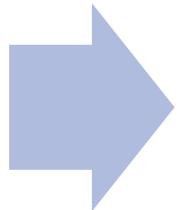


# User Satisfaction Metrics



## Usability

- SUS score
- TLX score
- General feedback and suggestions



## Controllability and collaboration

- Novelty and diversity of dataset w.r.t. user parameters
- Utility with tertiary model w.r.t. user parameters
- General feedback

# Project Plans

- Frontend and backend base has been setup and complete:
  - Overall UI architecture and interaction elements finalized and deployed
  - Final imputation network up and running
  - Simplified version of synthesis network up and running for single-class dataset synthesis
  - Example applications finalized: ECG and human motion data synthesis
- To-do-list:
  - Change SenseGen to PhysioGAN
  - Alter UI to include statistical user parameter/condition entry elements, multi-class synthesis and explainability elements:
    - Novelty, diversity, privacy, synthesis and imputation metrics displayed in non-technical and graphical manner
  - Report evaluation metrics and conduct user-study to report user satisfaction metrics

# THANK YOU

[https://github.com/swapnilsayansaha/ECE209AS\\_Fall2020/tree/main/Bake\\_off\\_2-GUI\\_GAN](https://github.com/swapnilsayansaha/ECE209AS_Fall2020/tree/main/Bake_off_2-GUI_GAN)

**Contributions:**

- Viacheslav Inderiakin: Application frontend design and integration of all the parts.
- Swapnil Sayan Saha: Tensorflow backend design and general project directions.

# Discussion

- Novelty, diversity and utility (w.r.t. user input parameters and corrections) are three numerical controllability and collaboration parameters that we have proposed to ensure “*a user can freely control the generative process and obtains results that match their expectation*”. Do you have any other metrics in mind that can do the same?
- The framework currently deals with data processing but not data collection. Can you visualize a framework that will guide non-experts in any domain starting from data collection until model deployment?
- How can this framework be expanded for images? (keep in mind the framework must generalize for any application like the one proposed)