INCM FINAL PROJECT PRESENTATION

Ashutosh Rudrabhatla
UG3 CSD
(2022111036)

AGENDA

Aim of the Project

Previous Work

Implementation

Results

Challenges Faced

Conclusion and Future Works

AIM OF THE PROJECT

Izhikevich Neuron Model

- Getting a clear understanding of the various aspects of the Izhikevich Neuron Model.
- Successfully simulating various types of firing using the Neuron model to get a better understanding of how it works.

Classification

- Getting an understanding of how to handle and convert datasets into current input signals.
- Comparing and contrasting performance of (traditional) models used for classification with the Izhikevich Neuron Model.

PREVIOUS WORK

Izhikevich Neuron Model and its Application in Pattern Recognition

- Roberto Vazquez, 2012.
- This is the paper being replicated.
- Uses the Izhikevich Neuron Model for classification and compares it to some traditional methods.

Simple Model of Spiking Neurons

- Eugene M. Izhikevich, 2003.
- This is the original paper that introduced the model.
- Gives physical significance as well as simulation procedures for the Izhikevich neuron.

IMPLEMENTATION

Neuron Model Simulation

Parsing and understanding the data

Conversion of data into current input signals

Creating functions to train and test the neuron model

Comparing accuracy with traditional classification methods

Visualization of results

NEURON SIMULATION

System of Equations:

$$v' = 0.04v^2 + 5v + 140 - u + I$$
$$u' = a(bv - u)$$
$$\text{If } v = 30mV \text{ Then } v = c \text{ and } u = u + d$$

Significance of variables/parameters:

- **u**: represents the membrane recovery variable.
- **v**: represents the membrane potential of the neuron.
- **I**: External current input.
- **a**: represents time scale of **u**.
- **b**: represents the sensitivity of **u** to the fluctuations of **v**.
- **c**: represents the after-spike reset value of **v**.
- **d**: represents the after-spike reset value of **u**.

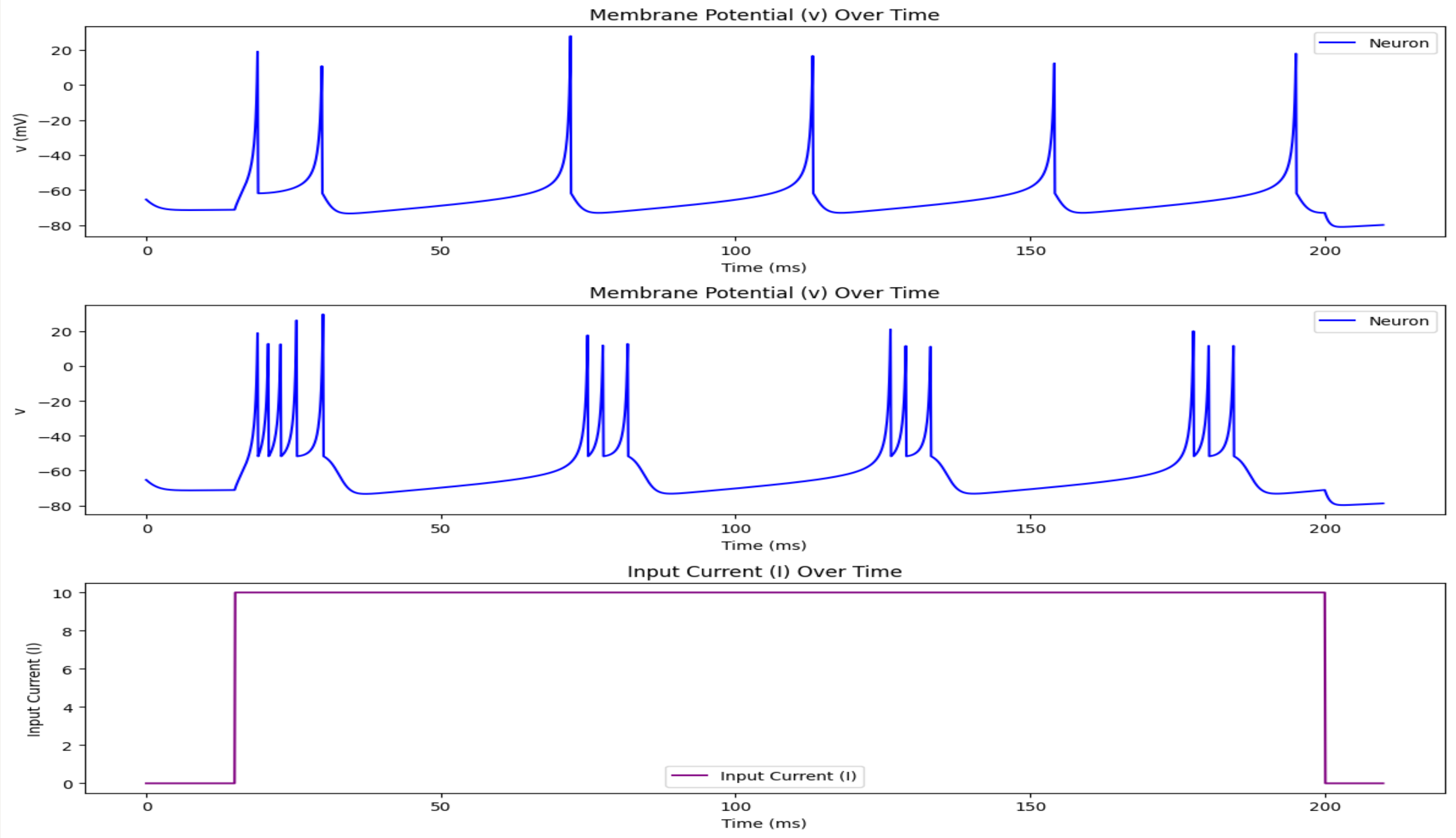
Note - Property of params:

- For Regular Spiking:
  - $c, d = -65mV, 8$
- For Intrinsic Bursting:
  - $c, d = -55mV, 4$

NEURON SIMULATION

NEURON SIMULATION PERFORMED

- Upper Graph: Regular firing
- Lower Graph: Intrinsic Bursting



DATASETS USED

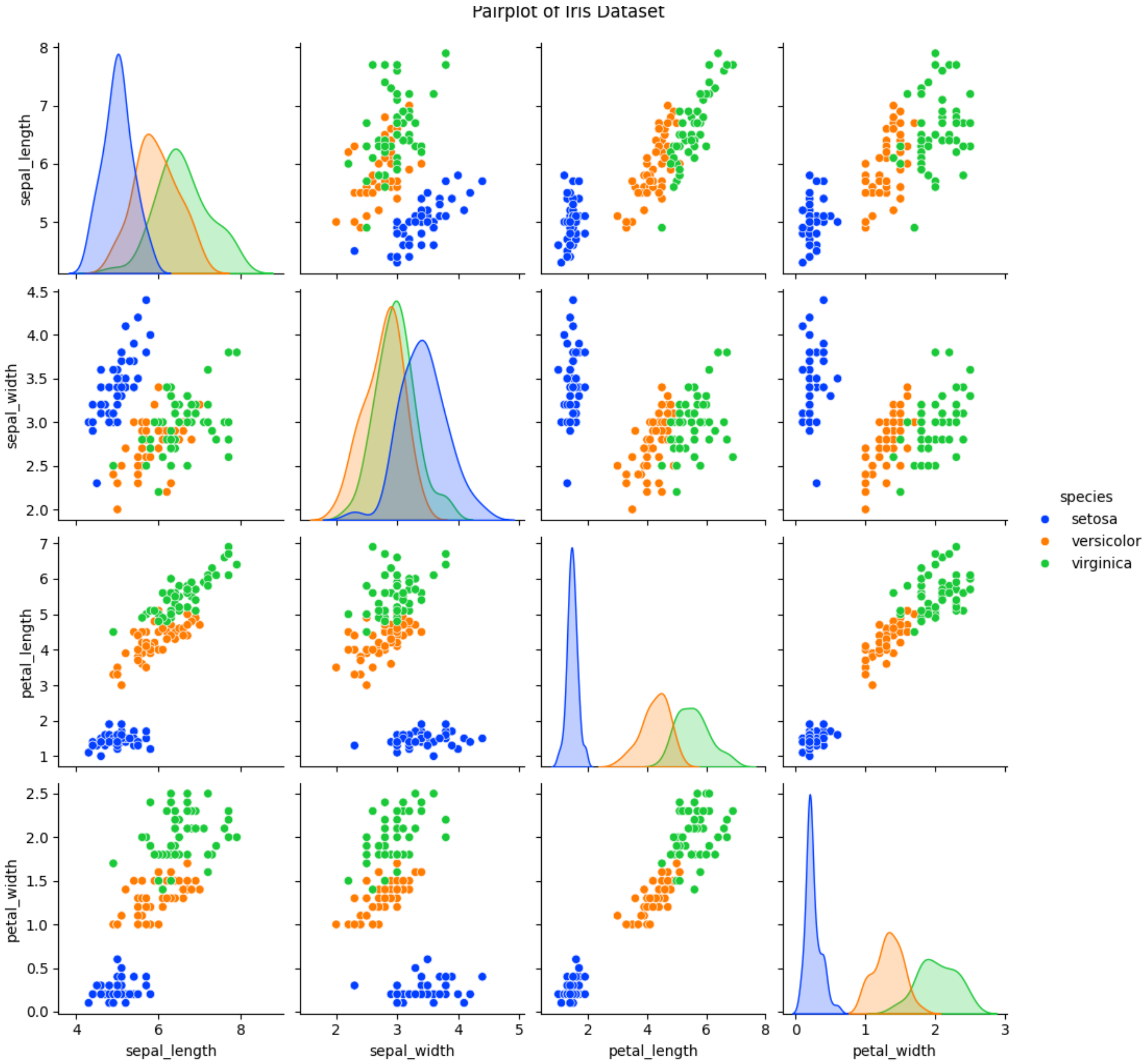
Iris Flower Dataset

- 3 different types of flowers, along with Sepal length, Sepal width, Petal length, Petal width.
- 150 total datapoints.
- Contains only float values

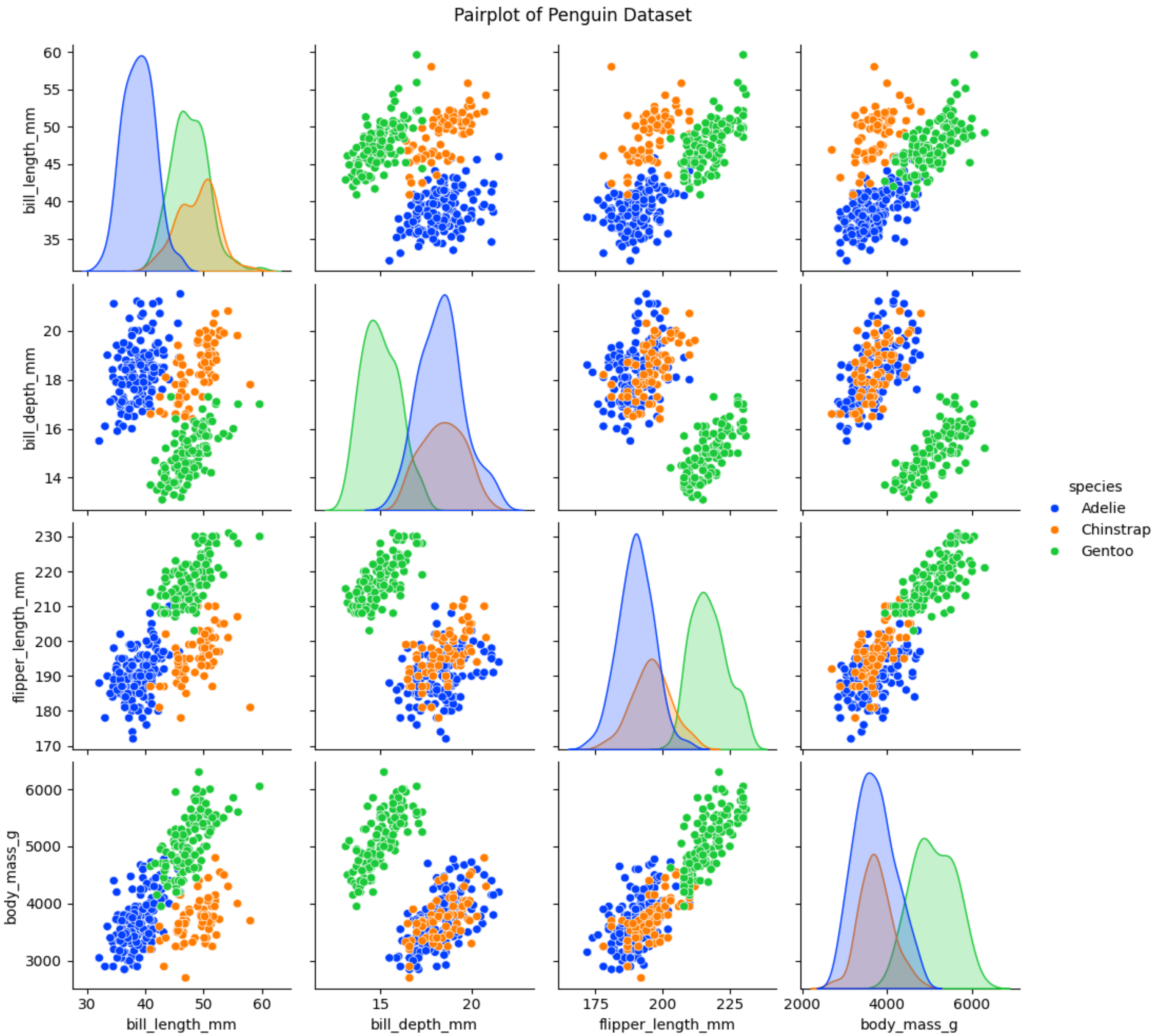
Penguins Dataset

- 3 types of penguins, which islands they are found in, bill length, bill depth, flipper length, weight and sex.
- 344 total datapoints.
- Contains text and NaN values along with float values.

SOME VISUALIZATIONS (IRIS DATASET)



SOME VISUALIZATIONS (PENGUINS DATASET)



Preprocessing

- Categorical variable encoded using SciPy modules encoder.
- Missing values replaced with means
- Numeric features scaled appropriately
- Non-float variables ignored

Conversion

- Current input signal broken into chunks, signifying each column
- Rescaling is done and current output is constant step current for each chunk.

CONVERSION INTO CURRENT INPUT

TRAINING AND TESTING THE MODEL

Training

- The parameters a, b, c and d are optimized.
- The bounds for each are as follows:
- a: (0.001, 0.2), b: (0.1, 1), c: (-80, -30), d: (0, 10)
- The bounds are determined through trial and error.
- Optimization function used: **differential evolution.** (since this is what the paper references)
- The feature of the neuron which works as extraction feature for classification is **spiking rate.**

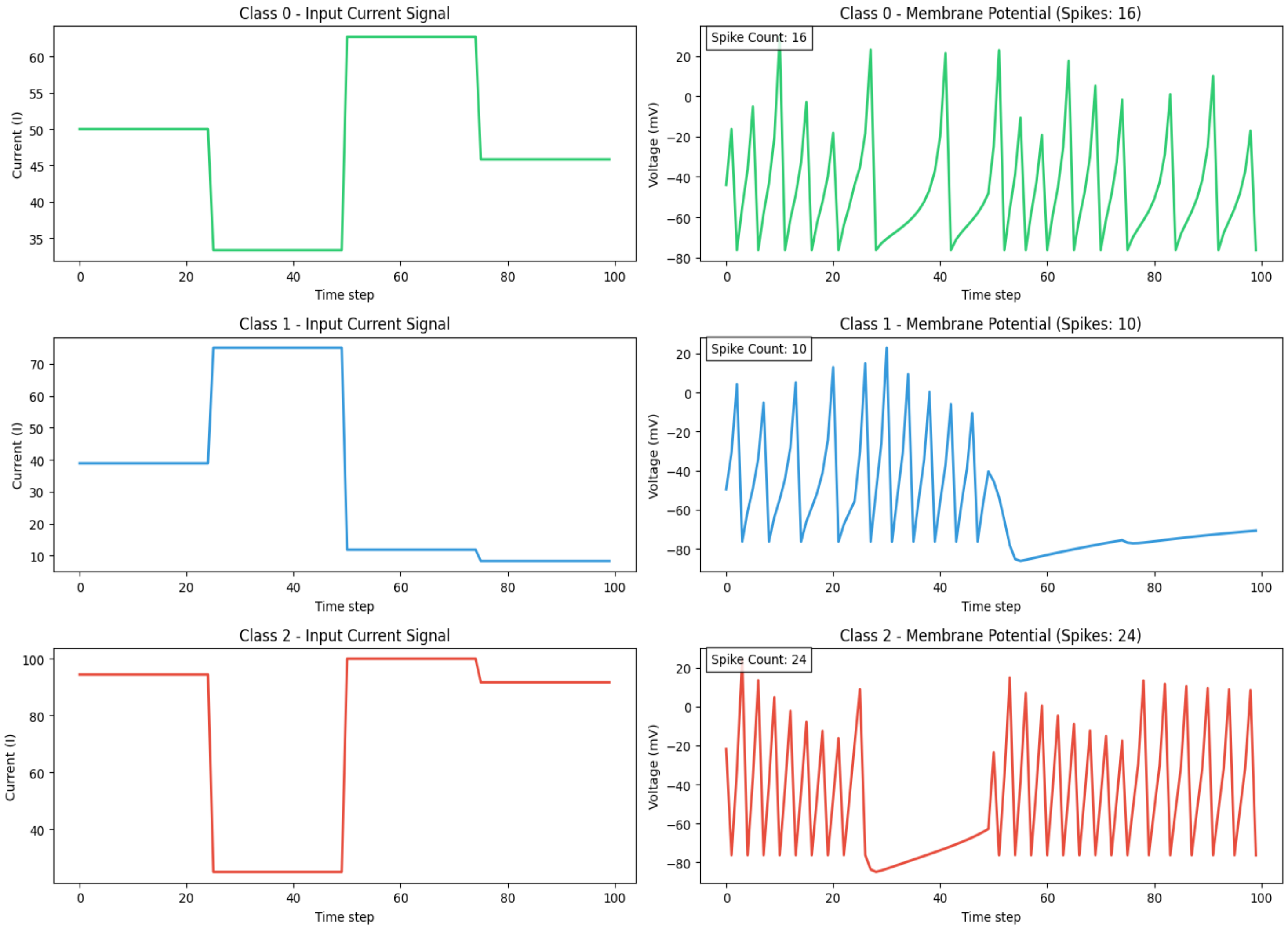
TRAINING AND TESTING THE MODEL

Testing

- Testing is done using an 80:20 split of train vs test data.
- Five-fold cross-validation strategy used (as mentioned in the paper) to calculate accuracy.
- optimal a, b, c, d are reported, along with accuracy.

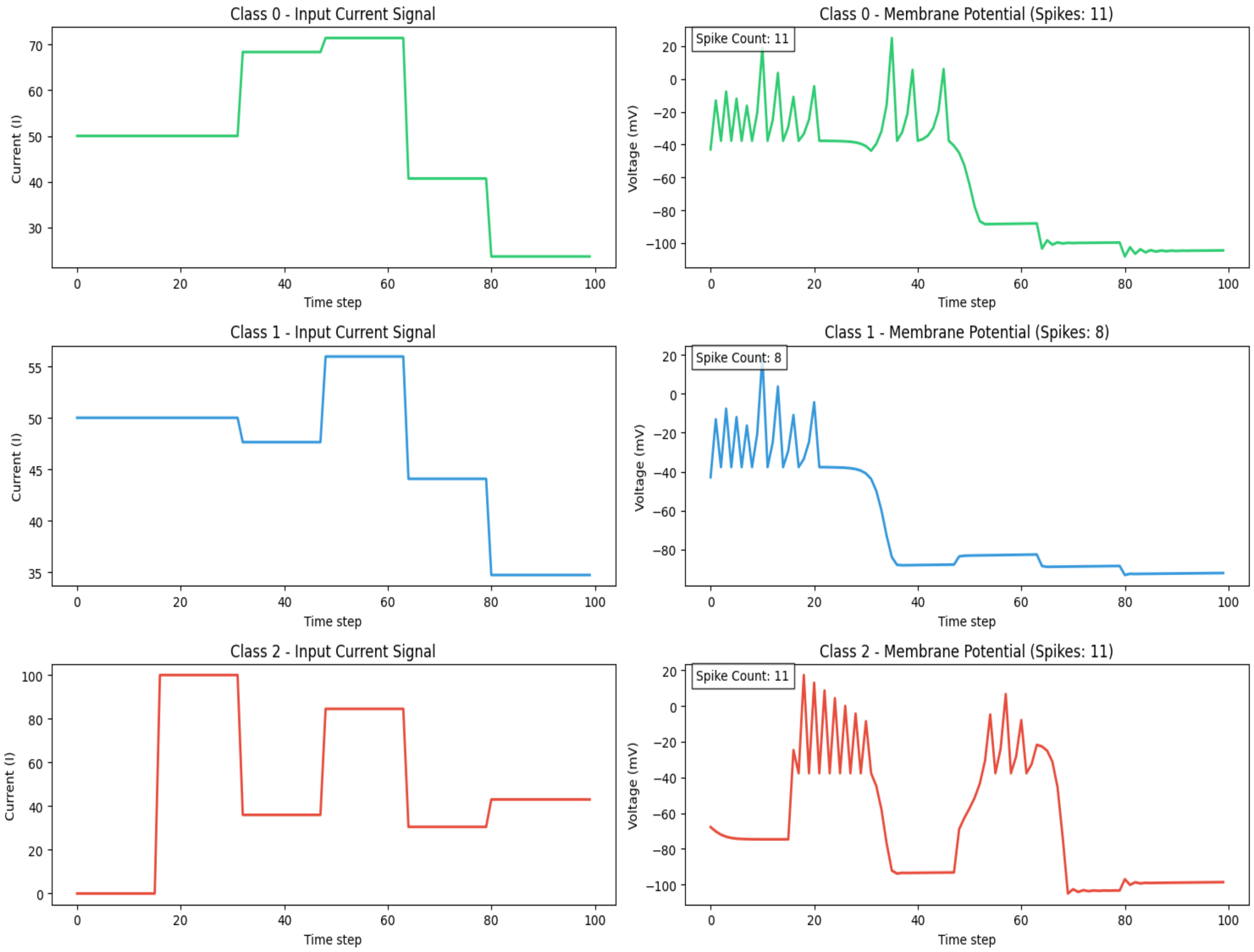
VISUALIZING THE RESULTS

Neuron Spiking patterns for each class/category of **Iris dataset**, along with the current input signals generated.



VISUALIZING THE RESULTS

Neuron Spiking patterns for each class/category of **Penguins dataset**, along with the current input signal generated



COMPARING ACCURACIES

Iris Dataset

- Logistic Regression Accuracy: 1.00
- k-NN Accuracy: 1.00
- SVM Accuracy: 0.97
- Decision Tree Accuracy: 1.00
- Random Forest Accuracy: 1.00
- **IZ model accuracy:** 0.90

Penguins Dataset

- Logistic Regression Accuracy: 0.97
- k-NN Accuracy: 0.68
- SVM Accuracy: 0.99
- Decision Tree Accuracy: 0.97
- Random Forest Accuracy: 0.96
- **IZ model accuracy:** 0.80

DISCUSSION

- This project serves as a proof of concept to how single-neuron models can also be used for classification problems.
- However, with advancement in specialized algorithms for classification, these models might not be able to compare.
- It is still worth noting, however, that the power of these models when used as activation functions for neurons of neural networks still remains untested.

CHALLENGES FACED

- **No resources available for reliable conversion of data into current input signals.**
- **Finding optimal bounds, iterations etc. for the optimization function was tedious.**
- **The original study is not well-explained.**

POSSIBLE IMPROVEMENTS

- **Using features other than firing rate for classification.**
- **Testing out other optimization functions, different bounds etc.**
- **Testing out multi-neuron models: Using the IZ neuron spiking function as activation function for neural network.**
- **Trying out the single-neuron model for other ML problems.**

THANK YOU

Ashutosh Rudrabhatla

Roll no: 2022111036

ashutosh.rudrabhatla@research.iiit.ac.in

github link