AML A1 Milestone 2

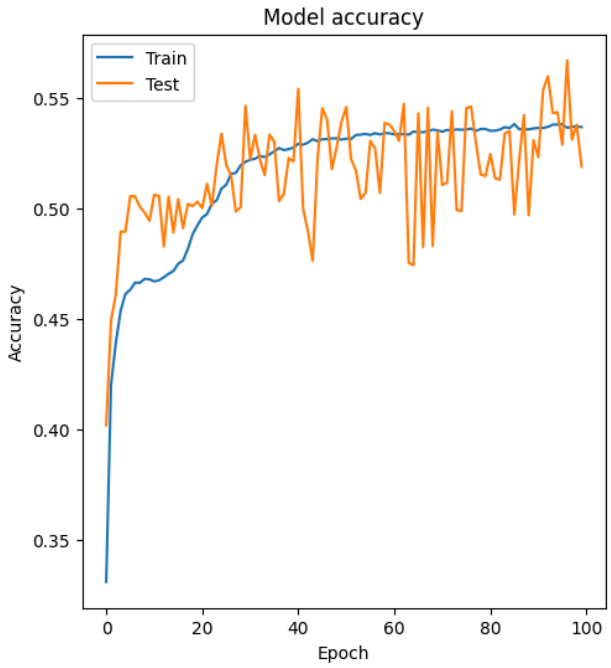
Group 1

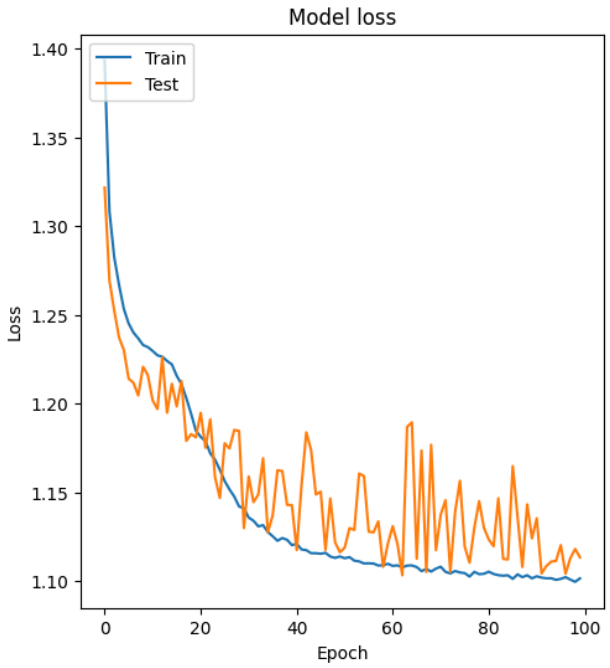
Aaditya Gupta & Harjeet Singh Yadav

Predicting target_10 of High noise data with LSTM

- Sorted the data frame wrt [‘row_num’]
- To create sequential data, created a window of size 10 with the label of that of the 10th datapoint.
- Trained an LSTM Model over this dataset.
  - ```
    model = Sequential([
        LSTM(100, input_shape=(n_steps, n_features), return_sequences=True),
        Dropout(0.2),
        LSTM(50),
        Dropout(0.2),
        Dense(n_classes, activation='softmax')
    ])
    ```
-

Data type	Accuracy	Loss
Training	**0.5370**	**1.1017**
Validation	**0.5191**	**1.1135**




    ```

Predicting target_10 of High noise data with TabPFN:

- Created an ensemble of TabPFN classifiers
- Since they only run on small subsets of datasets, made predictions using the ensemble trained on random subsets of datasets and took the majority voting as the actual predictions.
- The test set was taken of size 3000, and from the remaining data, random chunks of 3000 were used to train 81 TabPFN Classifiers.
- Accuracy on test set: **0.6123333333333333**

Predicting 'era' using TabPFN

- Predicting 'era' using TabPFN involves a constraint: it can only handle up to 1000 training examples and 10 classes. To circumvent this limitation, an **ensemble learning approach** was employed to manage the 12 classes present in the dataset.

- The strategy involved random selection of 1000 samples from the training dataset, focusing on only 10 classes at a time. These selected samples were then utilized for training the TabPFN model. Subsequently, predictions were made on the test set.

- This process was repeated multiple times to cover all classes effectively. Finally, a majority voting mechanism was implemented to determine the final predicted label.

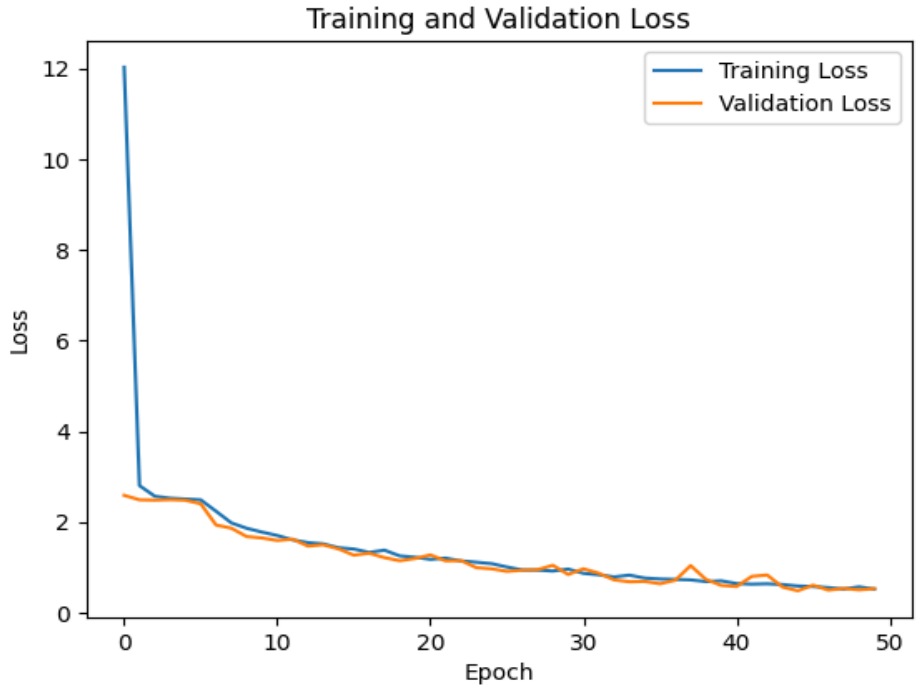
- **Accuracies**
    - Zero Noise:- 91.15 %
    - Low Noise:- 67.9 %
    - High Noise:- 45.84 %

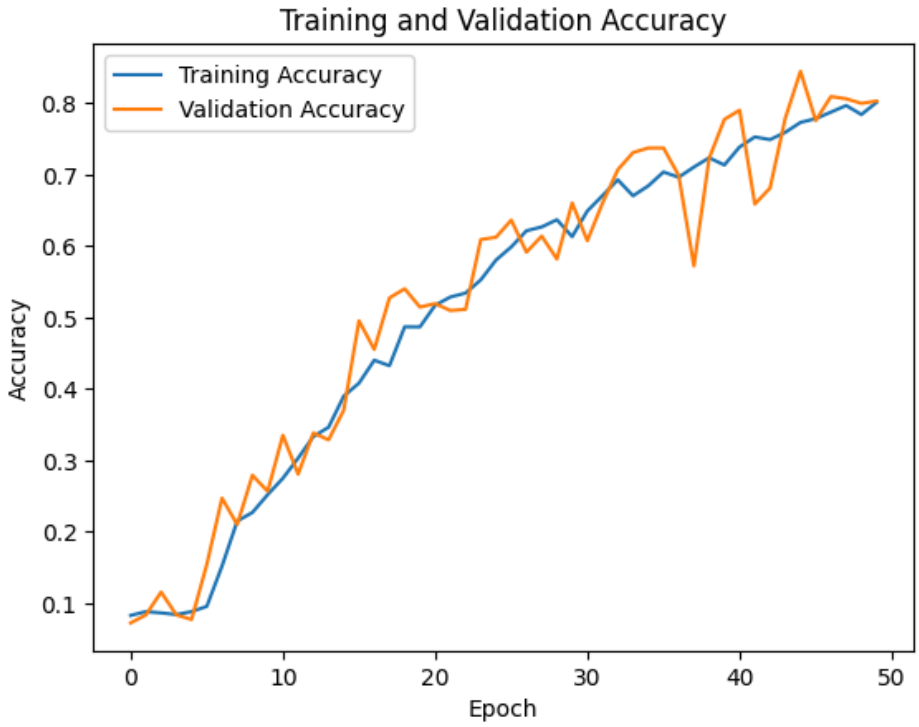
Predicating 'era' using sequence model:- RNN

- implemented a Recurrent Neural Network (RNN) model using TensorFlow and Keras for a classification task on a dataset representing the properties of a sinusoidal curve.

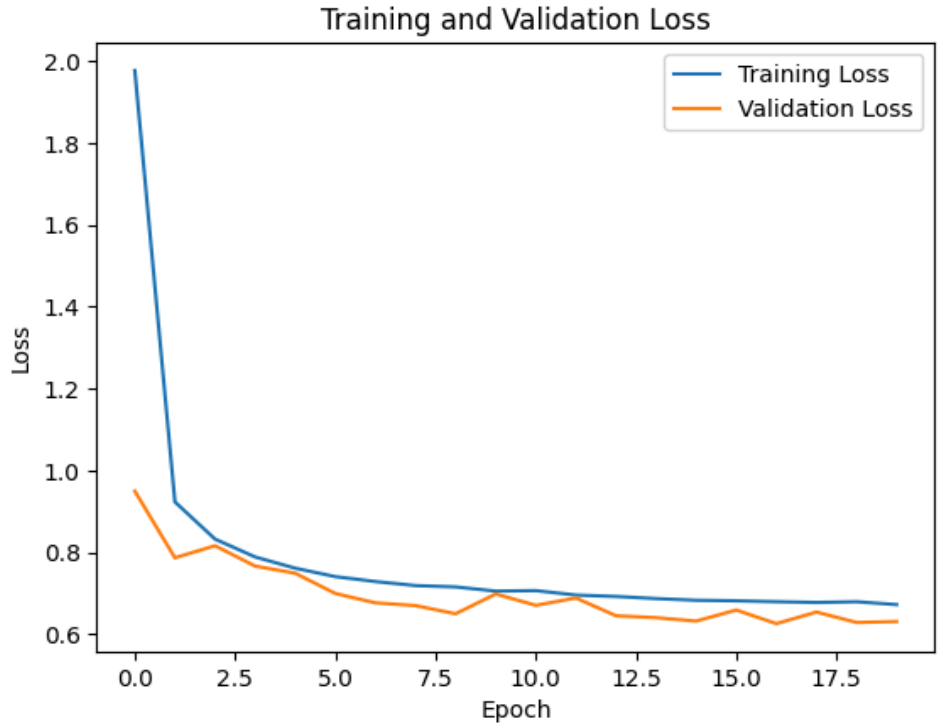
- The dataset consists of sinusoidal curve properties represented as features. The model architecture includes three SimpleRNN layers with dropout regularization for improved generalization.
- The output layer uses softmax activation for multi-class classification. The input shape for the model is (26, 1), indicating 26 features per time step. The model is trained for 10 epochs with a batch size of 32.
- Training and validation loss and accuracy curves are plotted to visualize model performance. Finally, the model is evaluated on the test set, and the test loss and accuracy are reported.

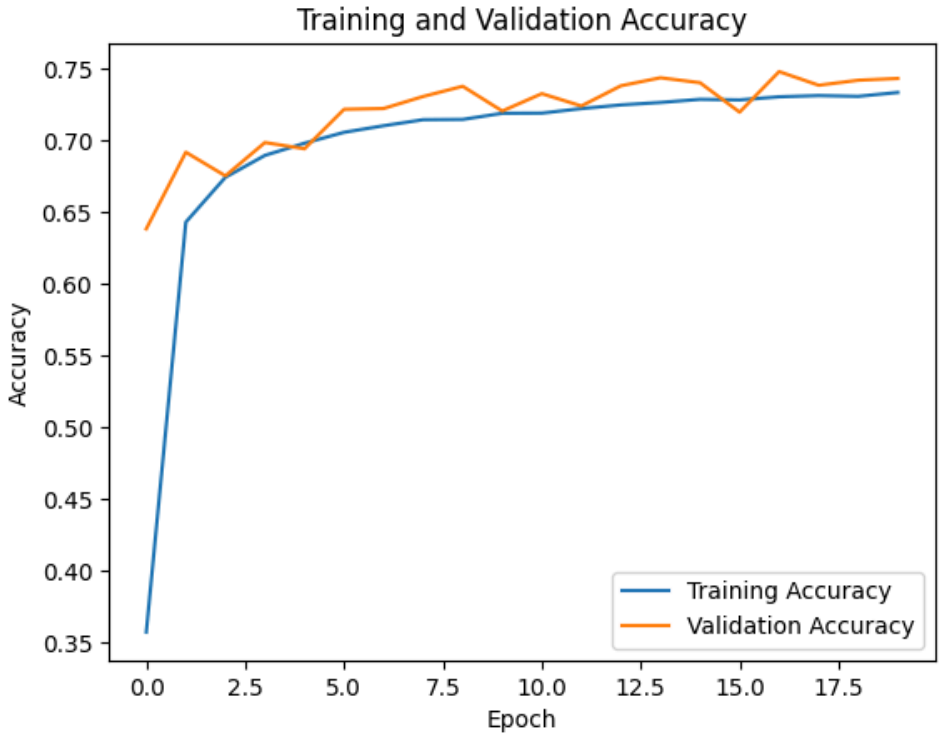
- **Accuracies**
  - Zero Noise:- 79.74 %
  - Low Noise:- 74.2 %
  - High Noise:- 50.01 %
- **Loss & Accuracy plot**
  - **Zero Noise**



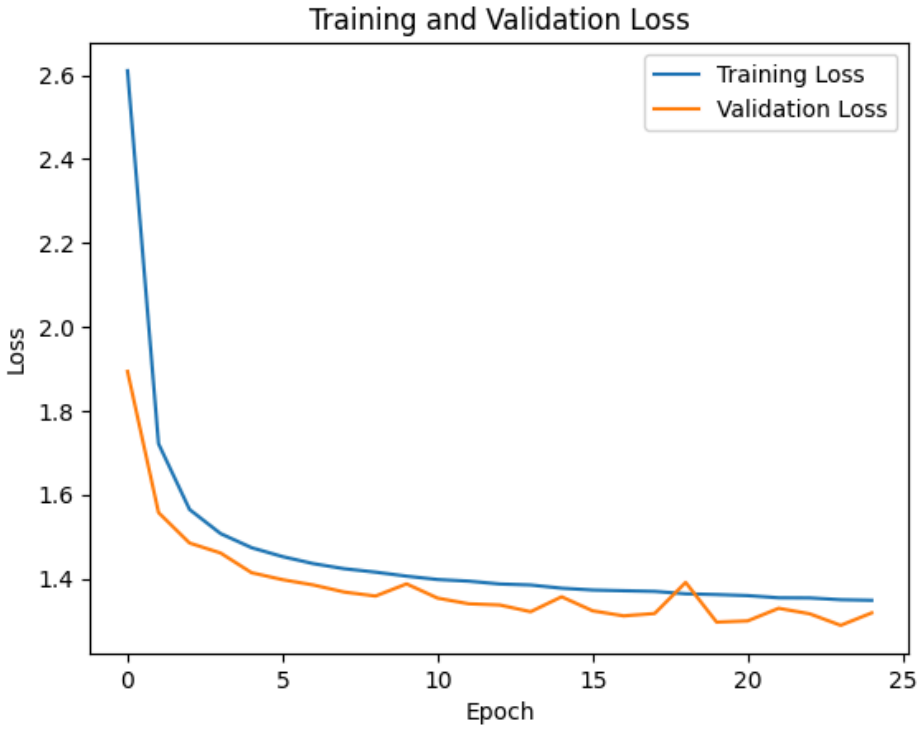


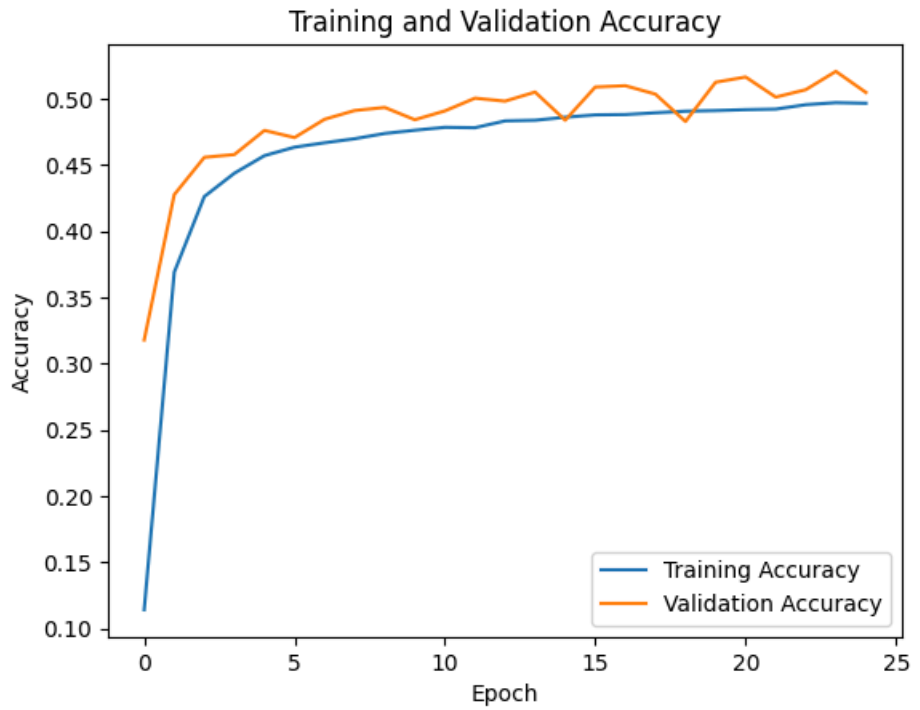
- **Low Noise**


Training and Validation Loss
Training Loss
Validation Loss
Loss
Epoch
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Training and Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epoch
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

- **High Noise**


Training and Validation Loss
Training Loss
Validation Loss
Loss
2.6
2.4
2.2
2.0
1.8
1.6
1.4
0
5
10
15
20
25
Epoch


Training and Validation Accuracy
Accuracy
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
Training Accuracy
Validation Accuracy
0
5
10
15
20
25
Epoch