# CSE572 DataMining Project2 Report

1215173661 Jinyung Hong

## 1. Model

The goal of the project2 is to predict whether patients have a meal or not using glucose level. This is one of kinds of problems which classify multivariate time series (MTS) consisting in assigning each MTS to one of a fixed number of classes. To address the problem, I used a reservoir computing approach. Reservoir computing (RC) is an extension of RNN models whose recurrent part is kept fixed and is either generated randomly, or by means of custom topologies for facilitating the information flow. Despite this simple structural feature, the recurrent part of the model (the reservoir) provides a rich pool of dynamic features which are suitable for solving a large variety of tasks. Indeed, RC models can achieve excellent performance in many fields, including time series forecasting, process modelling, and speech analysis. Since the reservoir is fixed, one needs to train only the readout, which provides an instant mapping between the internal representation of the reservoir and the task-specific output. By referring to [1], the structure of the model used is as follows.

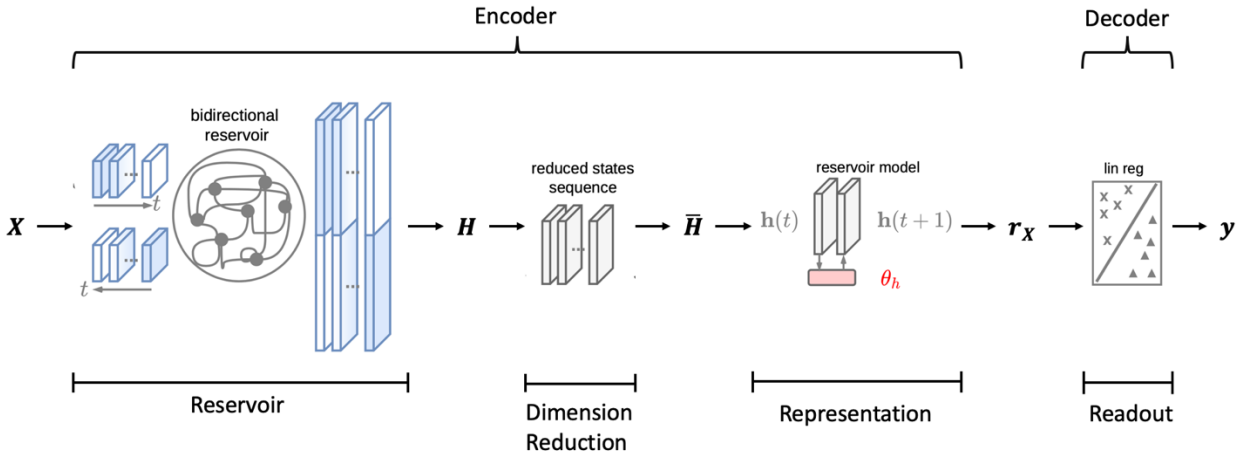



*Figure 1 The structure of the model*

The encoder part is to pre-process the input by applying a form of transformation from the input space to the feature space, where the latter is usually far higher-dimensional than the former. After that, dimension reduction such as PCA is applied to the last reservoir state to improve performance in the inference task. And then, the procedure to generate the reservoir model space representation is executed. Its result is a new feature vector and in the decoder part, a simple linear model is going to be trained using a new feature vector. You can check the structure of the model in detail in [1].

10-fold cross validation is used to choose the best model with the score, which is the average of sum of test accuracy and f1 score of each model.

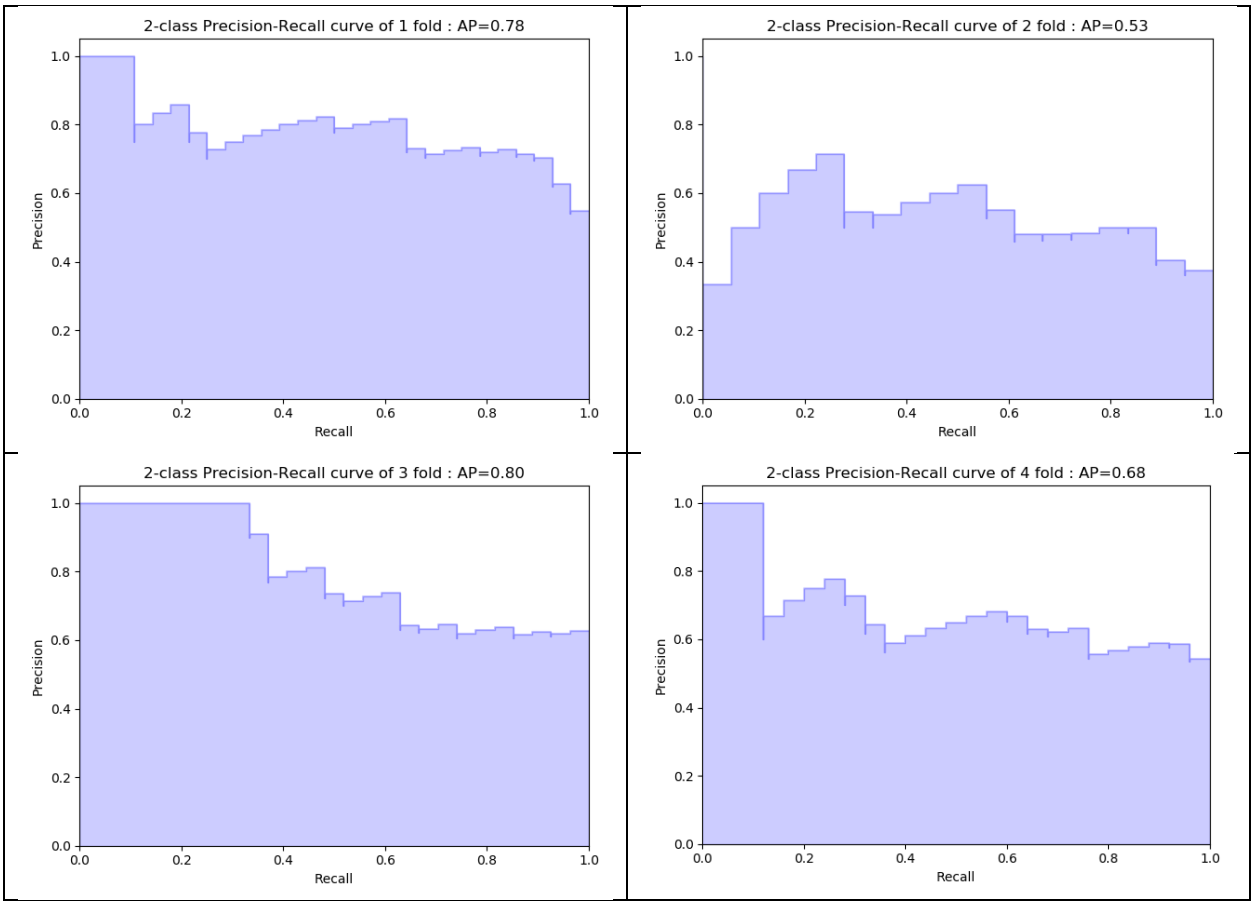
The project folder contains three kinds of files:

a. Files for training model
    i. train_model.py: the file for training the model
    ii. echo_state_network.py: the file of the model
b. Files for testing
    i. load_model.py: the file containing the function for testing
    ii. example_load_model.py: the example file about how to test
c. Files of the pre-trained model: pca.pkl, reservoir.pkl, ridge.pkl and ridgeclf.pkl

## 2. Evaluation

- The chosen model's score: 0.706

| | **1-Fold** | **2-Fold** | **3-Fold** | **4-Fold** | **5-Fold** | **6-Fold** | **7-Fold** | **8-Fold** | **9-Fold** | **10-Fold** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Test Accuracy** | 0.686 | 0.627 | 0.647 | 0.667 | 0.706 | 0.608 | 0.706 | 0.647 | 0.569 | 0.529 |
| **F1 Score** | 0.692 | 0.596 | 0.654 | 0.653 | 0.681 | 0.583 | 0.706 | 0.654 | 0.522 | 0.538 |




## 3. How to use a predict function for testing

For testing, you can just import the function in load_model.py, which is “predict()” as follows.

```
From load_model import predict

## X_test is test data
Prediction = predict(X_test)
```

You can check the file, “example_load_model.py”, in detail to know how to use predict function.

## 4. References

[1] Bianchi FM, Scardapane S, Løkse S, Jenssen R. Reservoir computing approaches for representation and classification of multivariate time series. arXiv preprint arXiv:1803.07870. 2018 Mar 21.