# Comparing Neural Network Methods to Predict Hypoglycemic Events in Type II Diabetic Patients

Tom Kirsh

October 29, 2023

## Contents

**1 Introduction** **2**

1.1 Predictive Modeling . . . 3

1.1.1 Logistic Regression . . . 3

1.1.2 Random Forest . . . 3

1.1.3 Artificial Neural Network . . . 4

1.1.4 RNN . . . 4

1.1.5 LSTM . . . 4

1.2 Anomaly Detection . . . 5

1.2.1 Autoencoder . . . 5

1.2.2 One-Class SVM . . . 5

**2 Methods** **5**

2.1 Data Processing . . . 5

2.1.1 Model Type: . . . 6

2.1.2 Event Horizon: . . . 6

2.1.3 Sensors Used: . . . 6

2.1.4 Data Augmentation: . . . 6

2.2 Hyperparameter Optimization . . . 7

**3 Results** **7**
3.1 MLP . . . . . 7
3.2 RNN . . . . . 7
3.3 LSTM . . . . . 10
3.4 Autoencoder . . . . . 11
3.5 One-Class SVM . . . . . 13
**4 Discussion** **15**
**5 Conclusion** **15**

**Abstract**

Abstract is written at the end.

## 1 Introduction

When food in consumed, the body breaks it down into sugar or glucose, which gets released into the bloodstream. If the glucose levels in the bloodstream get too high, the pancreas will release insulin to lower the blood sugar levels. Diabetes is when the pancreas does not produce enough insulin to lower the glucose levels or does so inefficiently. In these cases, glucose that remains in the bloodstream can cause problems in vision, kidneys, and the heart [CDC citation]. Taking medication, exercise and diet, and managing blood sugar levels are steps to prevent adverse outcomes, but there is not really a cure yet.

There are two types of diabetes. Type I diabetes is believed to be an autoimmune response where the immune system attacks [insert answer here] and the insulin production is halted. People with Type I diabetes need to take daily insulin. Type II diabetes is when the body doesn't use insulin well and blood glucose levels are not maintained.

Continuous Glucose Measurement Systems (CGMS) measure blood glucose levels by... These are prevalent in ... CGMS is typically used in people with Type I diabetes [source].

Deep learning models, or models using neural networks with many layers, have been used as state-of-the-art classification models for sequential data, including time series. There are a few neural network architectures, that is the design, that lead to better modeling of time series data.

## 1.1 Predictive Modeling

Multiple types of neural networks are compared to evaluate their performance, training time, and computational power.

### 1.1.1 Logistic Regression

Logistic regression is a binary classification approach that estimates the probability of belonging to either class. The logistic function is given by

$$\log \frac{P(C_1|\mathbf{x})}{1 - P(C_1|\mathbf{x})} = \mathbf{w}^T \mathbf{x} + w_0 \tag{1}$$

where $\mathbf{x}$ is our design matrix, the features in our dataset. $\mathbf{w}$ is the weight vector or coefficients that are estimated in the model. These coefficients are typically exponentiated into the odds ratios that denote the . The logistic function is equivalent the sigmoid function expressed as

$$P(C_1|\mathbf{x}) = sigmoid(\mathbf{w}^T \mathbf{x} + w_0) = \frac{1}{1 + \exp -(\mathbf{w}^T \mathbf{x} + w_0)} \tag{2}$$

The parameters can be estimated by either the maximum-likelihood estimation or by optimizing the parameters as discriminants to minimize the error using gradient descent. [cite book]

### 1.1.2 Random Forest

The Random Forest (RF) classifier is a collection of decision tree predictors that takes as input the data and independent identically distributed random vectors for each tree. These trees vote on the class of the input data and its prediction is assigned based on majority rule [cite RF paper]. Features are randomly selected at each node in the tree to determine the split. As the number of trees increase, the generalization error converges, reducing the overfitting that occurs. Random forests also have the ability to assess feature importance using the values in each feature after the tree has been constructed, randomly permuting the data and the out-of-bag (OOB) sampling is passed dow the tree. That classification is saved and the process is repeated for the remaining features. Afterwards, the misclassification error is calculated and used to get the overall importance by examining the percent increase in the misclassification rate [cite RF paper].

### 1.1.3 Artificial Neural Network

Artificial Neural Networks (ANNs) began by trying to model neurons in the brain. The most basic artificial neural network (ANN) consists of an input layer, a hidden layer, and an output layer. These can correspond to information coming to the neuron cell via its dendrites, the synaptic strengths modeled by weights that are updated with each epoch in the forward and back propagation steps, and the output signals passed through an activation function.

$$\hat{\mathbf{y}} = \sigma(\mathbf{W}\mathbf{X} + \mathbf{b}) \tag{3}$$

These can be used in either classification or regression problems. “Deep” ANNs consist of more than one hidden layer. Feedforward networks have been shown to have good classification predictions [cite] and thus we used different variations to model on. Hyperparameters trained on were the number of layers (what makes it a “deep” network), the number of nodes in each layer, the learning rate, and the dropout rate. Dropout layers were included to reduce overfitting [cite] by randomly removing a fraction of the nodes in the hidden layer.

### 1.1.4 RNN

Recurrent Neural Networks (RNNs) are used for any data that can be modeled as a sequence (language, time series, etc.) where the previous data is important for the current state of the data. [cite] To have the previous information persist, the RNN cell contains a hidden state that contains the “memory”. The equations for Recurrent Neural Networks come from signal processing in electrical engineering. The state system... Time delay is intrinsic to the system and that determines the dynamics.

... This leads to the standard equations of RNNs.

$$\mathbf{H}_t = \phi_h(\mathbf{X}_t\mathbf{W}_{xh} + \mathbf{H}_{t-1}\mathbf{W}_{hh} + \mathbf{b}_h) \tag{4}$$

where $W_r$ is the weight matrix for the previous state and $W_x$ is the weight matrix for the input data.

$$\mathbf{O}_t = \phi_o(\mathbf{H}_t\mathbf{W}_{ho} + \mathbf{b}_o) \tag{5}$$

### 1.1.5 LSTM

The vanishing gradients problem led to the creation of Long Short-Term Memory (LSTM) neural networks. These networks are capable of long-term dependencies, meaning they can retain older information easier than vanilla RNNs [cite]. Like RNNs, LSTMs have a looped cell. Instead of one layer inside the RNN cell, in the LSTM there are four layers: the forget layer, the input layer, the update layer, and the output layer.

### 1.2 Anomaly Detection

Since the number of positive samples in each participant's data is $< 1\%$, another way to build a model is to look at those glucose measurements as anomalous to "normal" glucose measurements.

#### 1.2.1 Autoencoder

#### 1.2.2 One-Class SVM

## 2 Methods

The data was collected using two CGM sensors attached to each participants' arms [cite Lisa's study]. Over a period of 16 weeks, participants' glucose measurements were collected from both sensors. Participants were randomly given a double-blinded CGM system, meaning they couldn't see their glucose measurements, or a blinded/unblinded CGM system. Out of the 40 participants, 22 were classified as high risk for hypoglycemic events, and indeed had multiple hypoglycemic events. We excluded 4 participants that had $<$10 events and used 18 participants' glucose data. Hypoglycemic events are defined as glucose levels $<$54 mg/dL measured by both CGM sensors at the same time lasting $\geq$15 minutes.

### 2.1 Data Processing

Over the measurement time period, both sensors did not measure glucose simultaneously. There were periods where either the left or right sensor measured, neither sensor measured, or both sensors measured. Even when both sensors measured, there were small delays in time between the left and right sensors' measurements. To synchronize the left and right glucose measurements, only the times that both sensors were measuring were kept. To handle the small time delays ($<$15 minutes), linear interpolation was used to map the measurements to a single time array [cite for linear interpolation] [cite for justification in signal processing] [cite in other CGM use?].

After the left and right sensors were interpolated, missing values were imputed using last-operation-carried-forward (LOCF) [cite]. The time series was converted into a a windowed or ”chunked” format where the previous 6 hours (24 measurements) of data were used to predict the outcome [cite]. During modeling, some sequences were padded with NaN values at the beginning of the signals and were not imputed with LOCF. During modeling, these values were dropped.

There were multiple combinations of models constructed. They were:

#### 2.1.1 Model Type:

- Individual: Only uses glucose data collected from a single person.
- General: Uses glucose data collected from everyone except one person, whose data is used as holdout.

#### 2.1.2 Event Horizon:

- 15 minutes before
- 1.25 hours before the event
- 2.25 hours before

#### 2.1.3 Sensors Used:

- Left

#### 2.1.4 Data Augmentation:

- None: Do not change the number of samples.
- Synthetic Minority Oversampling Technique (SMOTE): Uses *k*-nearest neighbors of the minority class to generate more samples of similar minority-set data. [cite]
- Random Undersampling: Randomly selects 4x the number of the minority class of the majority class for training. [cite]

All combinations of these data were modeled and compared for each subject.

## 2.2 Hyperparameter Optimization

Neural networks have many hyperparameters that can be adjusted in order to improve the modeling capabilities.

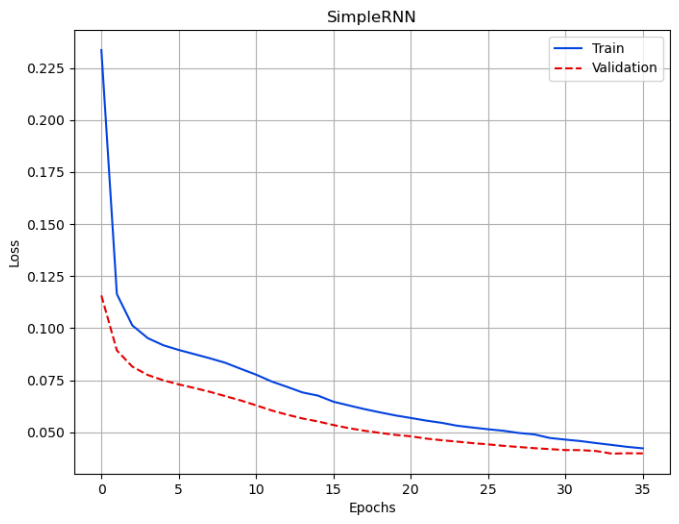
# 3 Results

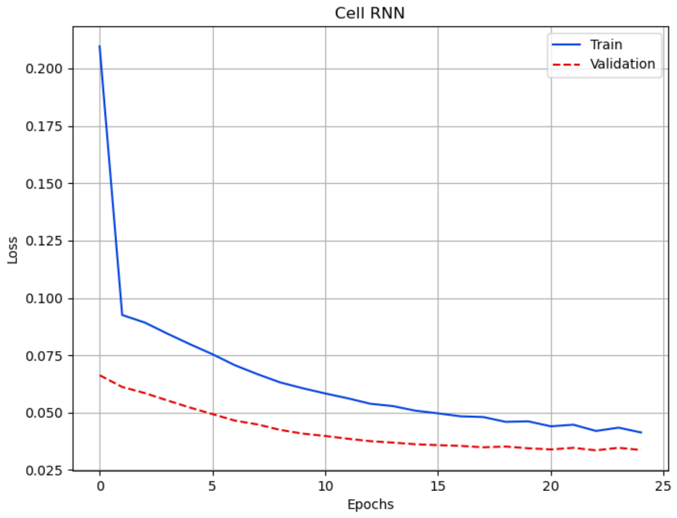
## 3.1 MLP

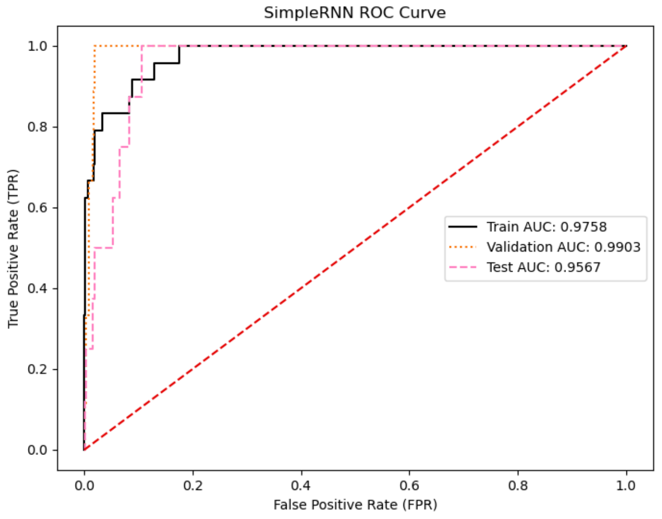
TEST RESULTS with Hyperband Optimization

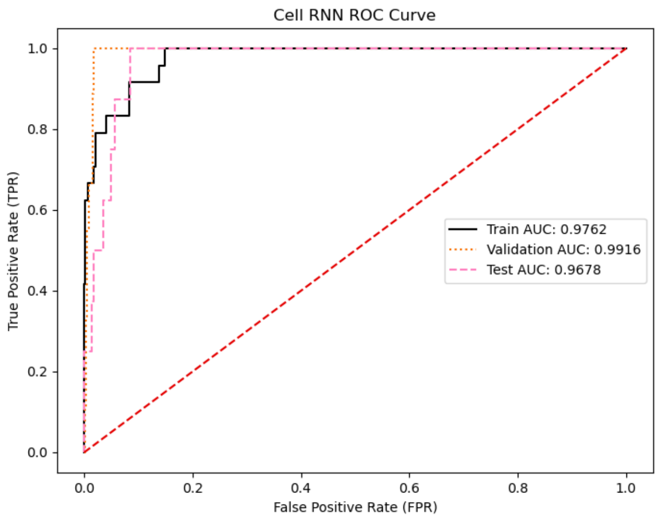
## 3.2 RNN

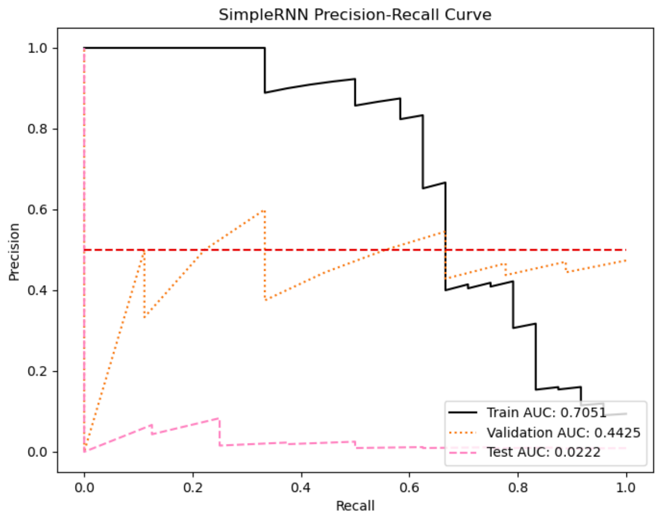
The datasets were run

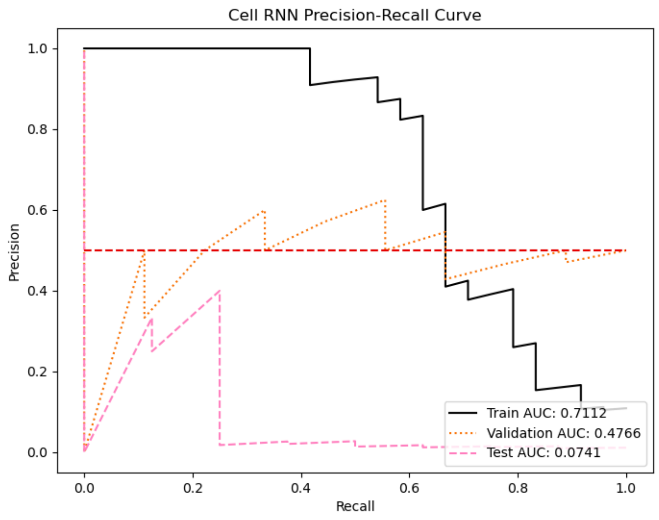



Cell RNN
Train
Validation
Loss
Epochs
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0
5
10
15
20
25

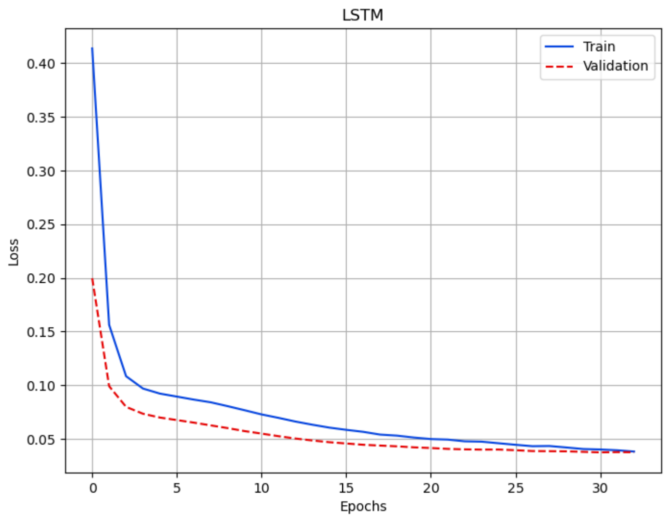

SimpleRNN ROC Curve
Train AUC: 0.9758
Validation AUC: 0.9903
Test AUC: 0.9567
True Positive Rate (TPR)
False Positive Rate (FPR)
0.0
0.2
0.4
0.6
0.8
1.0

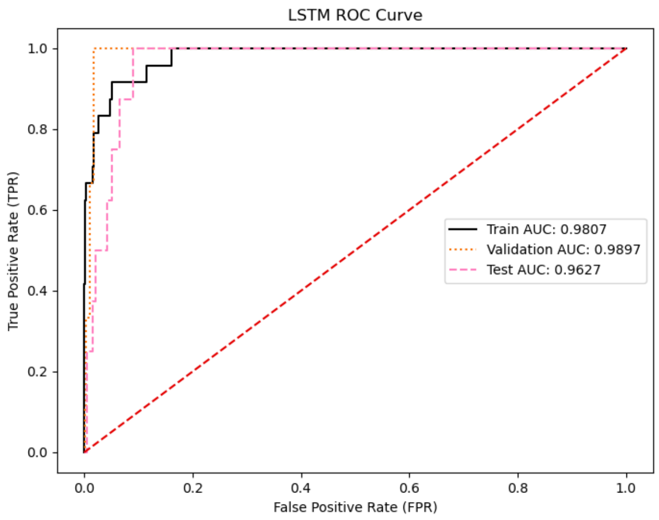

Cell RNN ROC Curve
True Positive Rate (TPR)
False Positive Rate (FPR)
0.0
0.2
0.4
0.6
0.8
1.0
Train AUC: 0.9762
Validation AUC: 0.9916
Test AUC: 0.9678

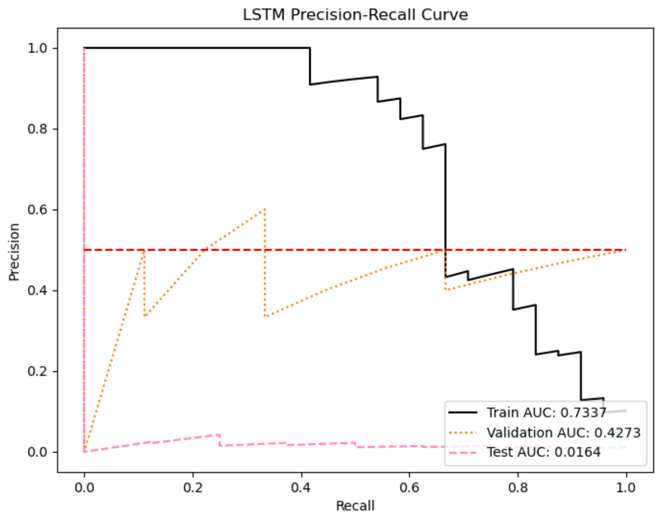

SimpleRNN Precision-Recall Curve
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
Train AUC: 0.7051
Validation AUC: 0.4425
Test AUC: 0.0222


Cell RNN Precision-Recall Curve
Precision
Recall
Train AUC: 0.7112
Validation AUC: 0.4766
Test AUC: 0.0741

## 3.3 LSTM


LSTM
Loss
Epochs
Train
Validation





## 3.4 Autoencoder

From the training data, the results were

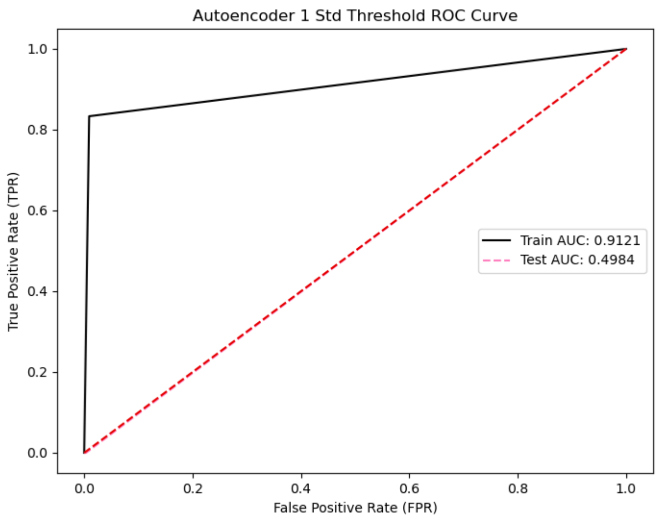
Threshold: 0.05633
Accuracy = 0.988

Precision = 0.625
Recall = 0.833

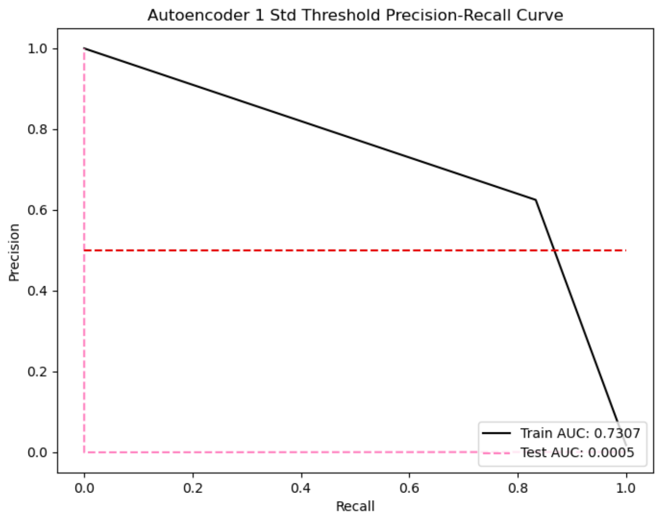
The autoencoder applied to the test data:

Accuracy = 0.99577
Precision = 0.0
Recall = 0.0

The ROC-curve is:



The PRC-curve is:



## 3.5 One-Class SVM

The datasets for the One-Class SVM have the sizes:

Normal train data shape is: (8578, 96)
Normal test data shape is: (1312, 96)
Anomalous train data shape is: (17, 96)
Anomalous test data shape is: (24, 96)

The parameters chosen are
Train Prediction results:
———————————————-
Accuracy = 0.9678
Precision = 0.9980
Recall = 0.9697

Test prediction results:
———————————————-
Accuracy = 0.8847
Precision = 0.9841

Recall = 0.8971

Quantile Threshold: 254.787286
1 standard deviation Threshold: 255.92859

Evaluation of the thresholds on the TRAIN data
——————————————-

TRAIN Quantile Threshold:

Accuracy = 0.9880
Precision = 0.9980
Recall = 0.9900

TRAIN STD Threshold:

Accuracy = 0.1881
Precision = 1.0000
Recall = 0.1865

Evaluation of the thresholds on the TEST data
——————————————-

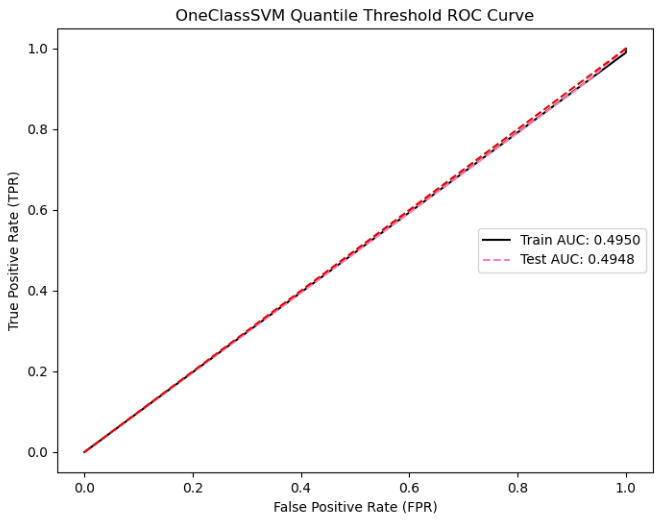
TEST Quantile Threshold:

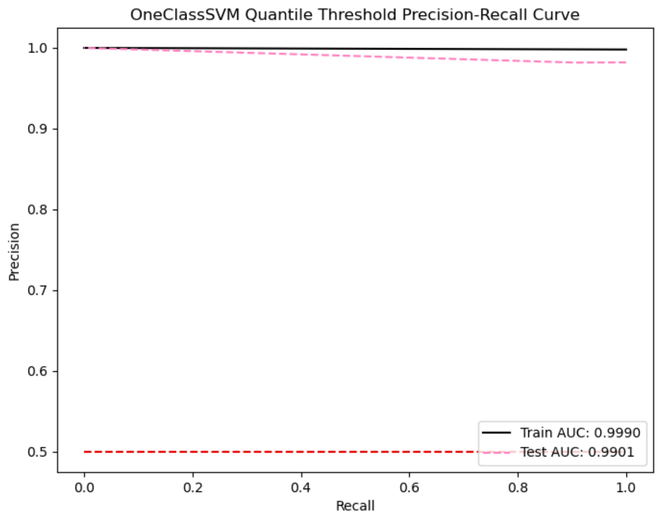
Accuracy = 0.8915
Precision = 0.9818
Recall = 0.9062

TEST STD Threshold:

Accuracy = 0.1018
Precision = 1.0000
Recall = 0.0854

Since using the quantile threshold seemed to be a better cutoff for anomalous measurements, I used that for the ROC and PRC curves.





## 4 Discussion

## 5 Conclusion