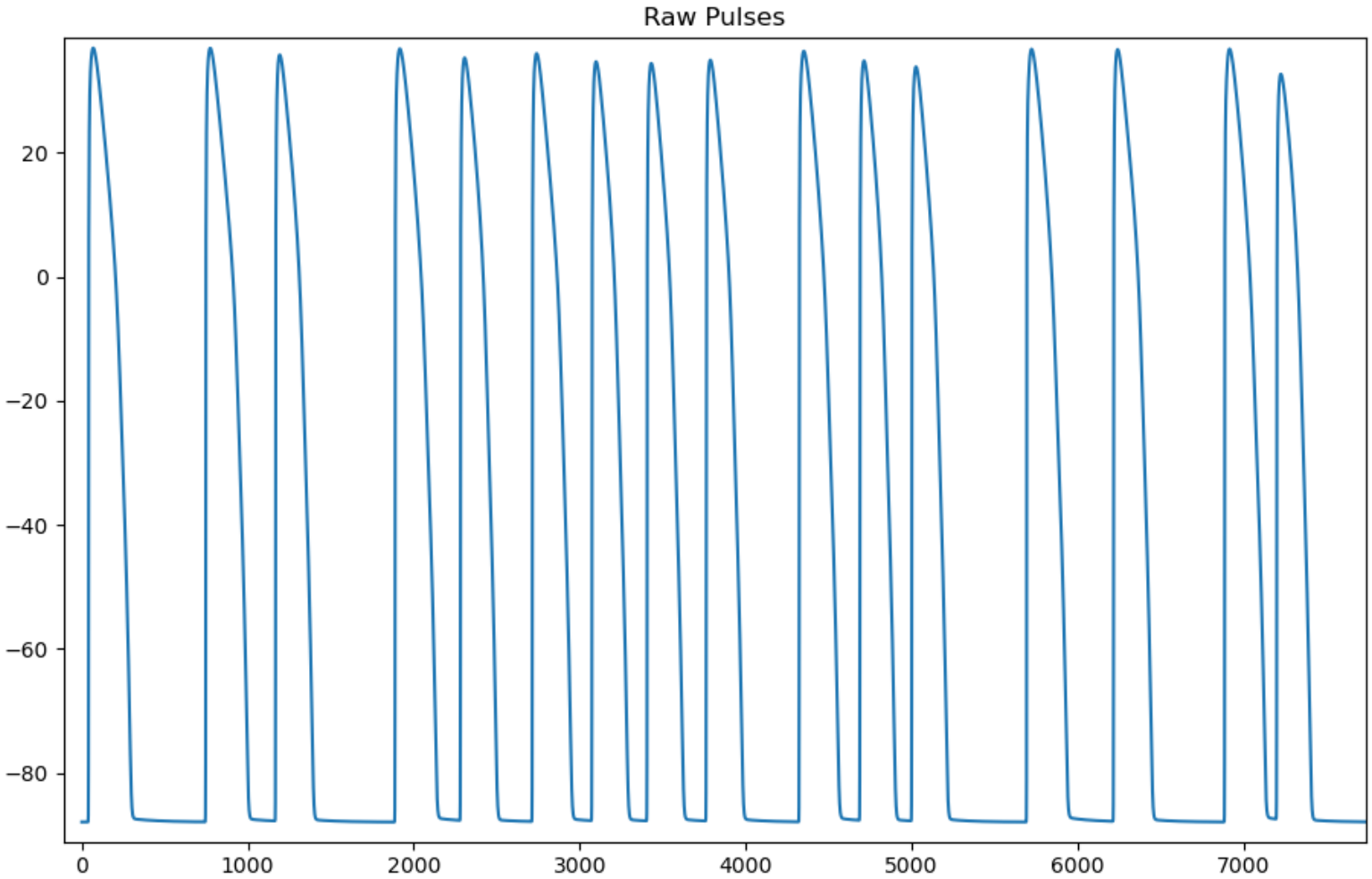
Machine Learning Models for Predicting Action Potential Sequences

Ines Christa

February 25, 2025

Raw Pulse Data



Segementation

- A thresholding method is applied to segement the sequence.
- The Segmentation is divided into Mode 0 and Mode 1.
- The Action Potential Duration (APD) varies across pulses.
- The goal is to create a matrix that represents the individual pulses.

Segmentation Modes

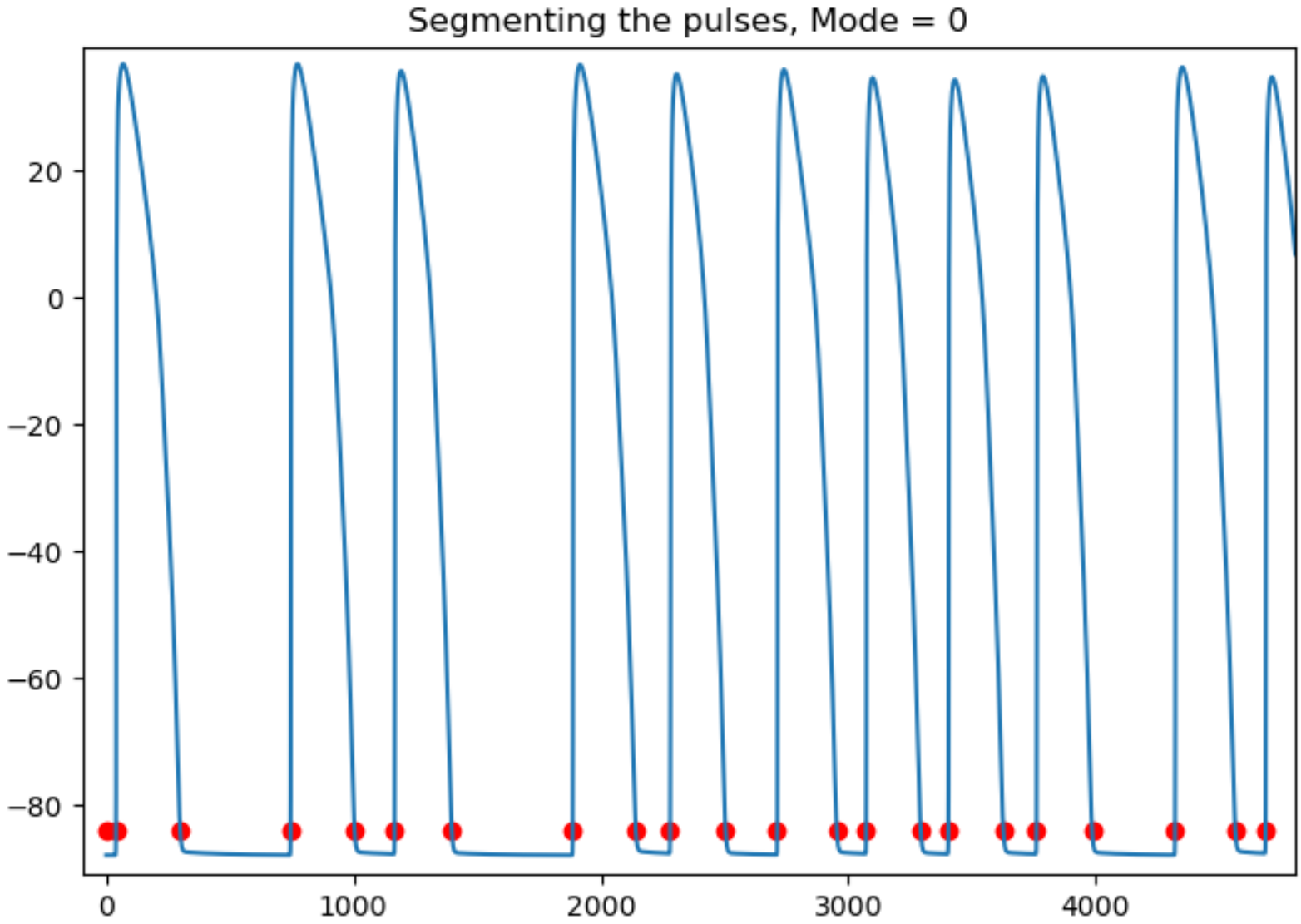
Mode 0:

- The sequence of pulses is segmented into individual pulses.
- Diastolic intervals are removed.

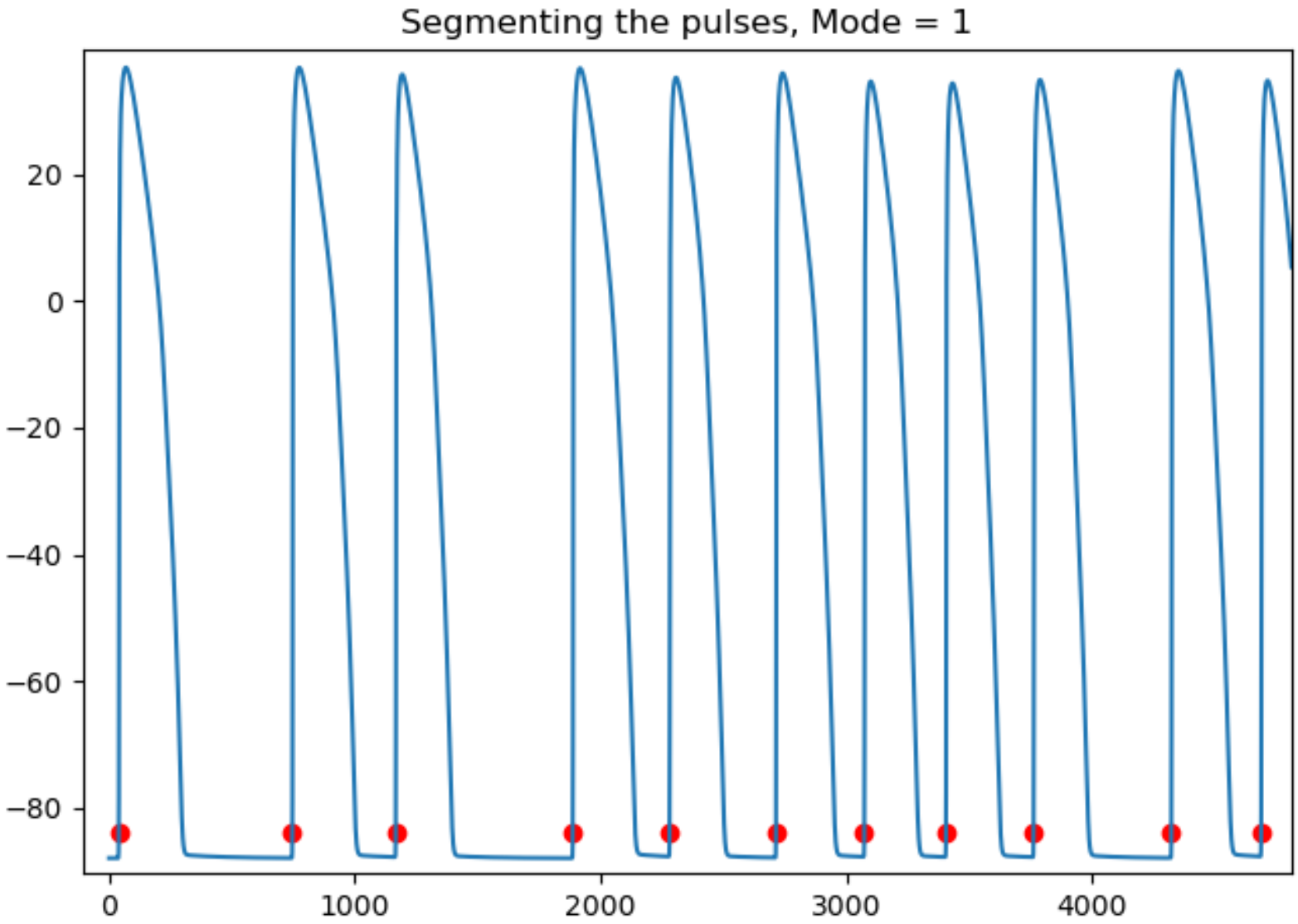
Mode 1:

- The sequence of pulses is segmented into individual pulses and the following diastolic interval.

Mode 0


Segmenting the pulses, Mode = 0
20
0
−20
−40
−60
−80
0
1000
2000
3000
4000

Mode 1


Segmenting the pulses, Mode = 1
20
0
−20
−40
−60
−80
0
1000
2000
3000
4000

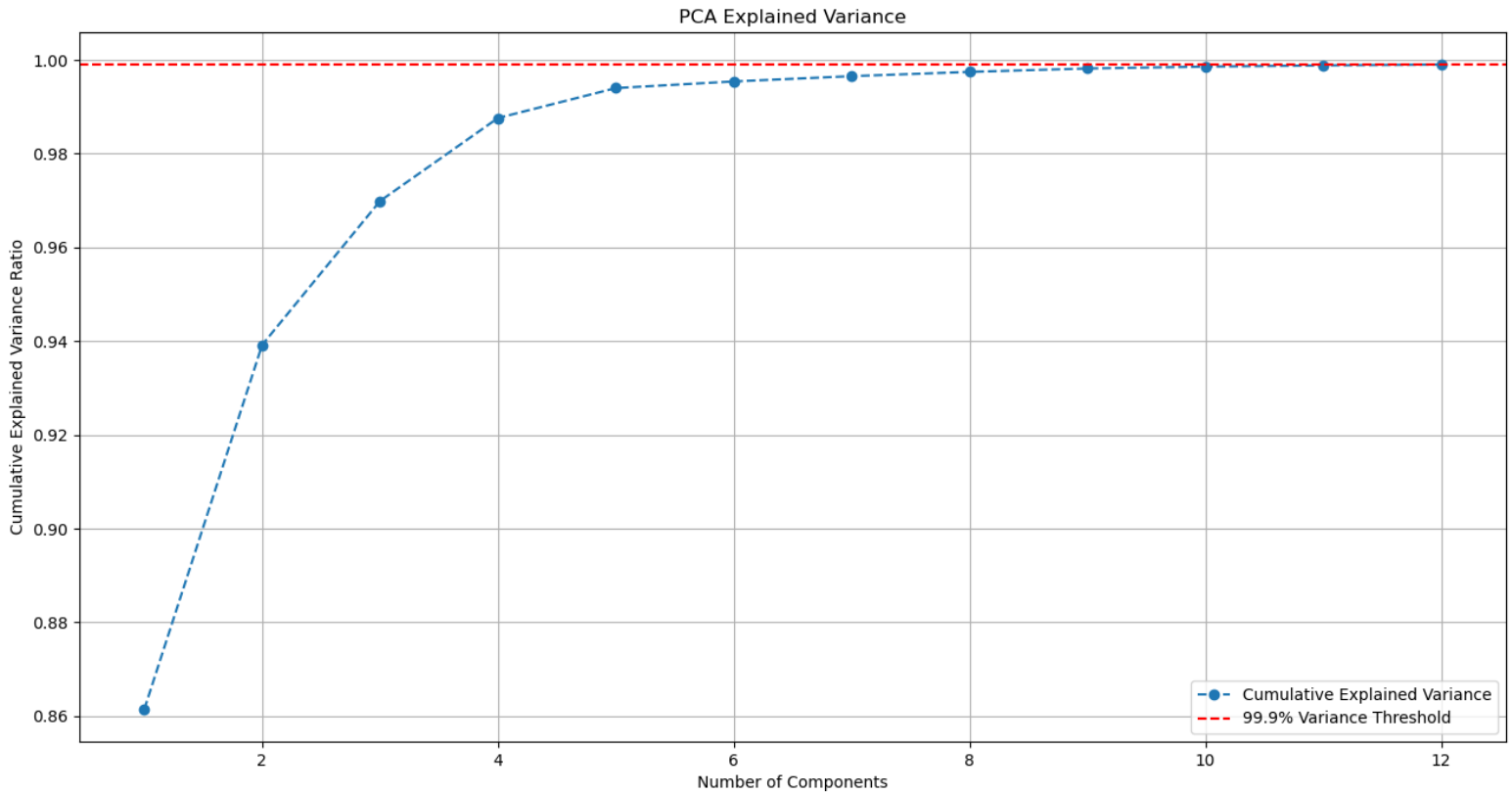
Building the Matrix

- Each segmented pulse is placed into a separate row of the matrix.
- Rows are zero-padded to a fixed length, **max_length + 10**.
- The final matrix shape, for Mode 0, is **(48074, 270)**, representing 48,074 pulses, each with 270 data points.
- The final matrix shape, for Mode 1, is **(48074, 989)**, representing 48,074 pulses, each with 989 data points.

Applying PCA to the Matrix

- Principal Component Analysis (PCA) is applied to the pulse matrix to reduce dimensionality.
- A total of 12 components are selected.
- These components explain **99.9%** of the variance in the dataset.

Variance Explained by PCA Components



The 12 selected PCA components account for 99.9% of the variance.

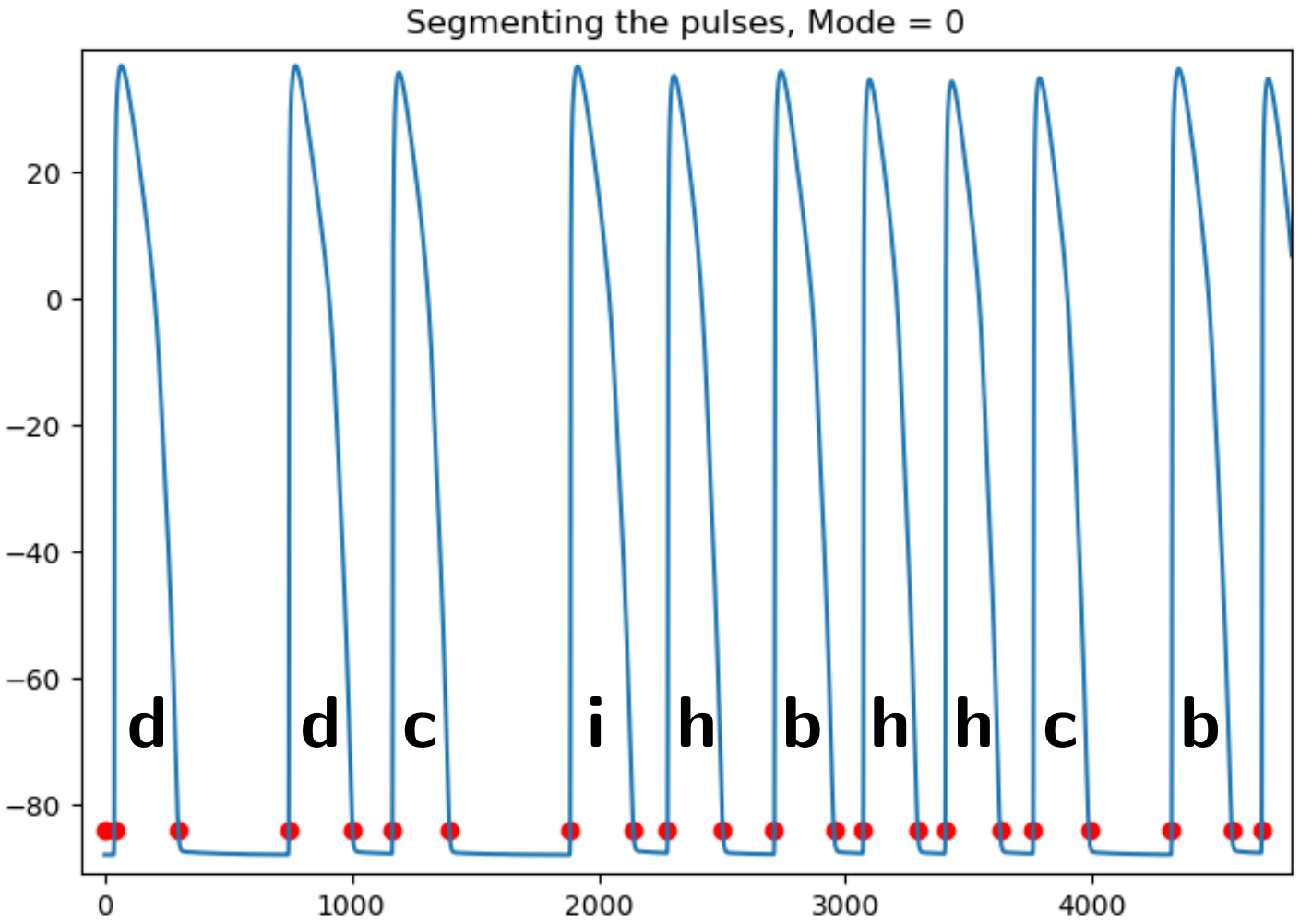
Clustering of Pulses

- The pulses are clustered into discrete groups.
- Clustering is performed using KMeans with 10 and 20 clusters.
- The goal is to ensure accurate pulse reconstruction while maintaining good clustering quality.

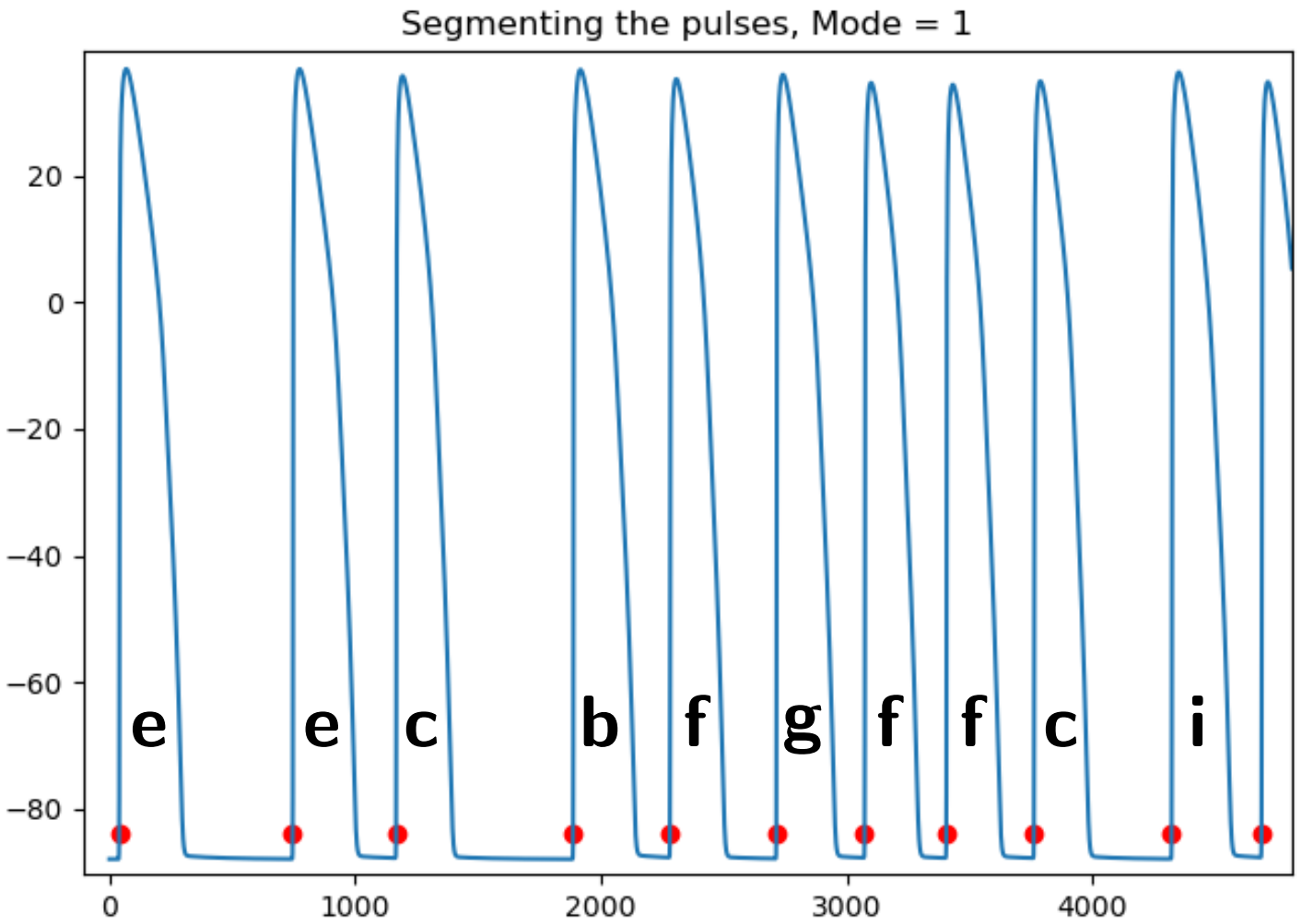
Alphabet

- Each segmented pulse (or pulse with diastolic interval) is assigned to a cluster, represented by a unique label.
- These labels form a sequential representation of the pulse data, preserving their temporal order.
- To utilize this sequence for modeling, input-output matrices X and y must be constructed.
- X represents past observations (lookback window), while y represents the predicted future values (forecast horizon).

Mode 0


Segmenting the pulses, Mode = 0
20
0
−20
−40
−60
−80
0
1000
2000
3000
4000
d
d
c
i
h
b
h
h
c
b

Mode 1


Segmenting the pulses, Mode = 1
20
0
−20
−40
−60
−80
0
1000
2000
3000
4000
e
e
c
b
f
g
f
f
c
i

Train/Test Data Split

d d c i h b h h c b d g b e e a a b c g e b a g d

90% Training Data 10% Test Data

The dataset, consisting of 48,074 pulses sorted into clusters, is split into **training** and **test subsets** to avoid data leakage and ensure accurate model evaluation.

Train/Test Data Split

d d c i h b h h c b d g b e e a a b c g e b a g d

90% Training Data 10% Test Data

The dataset, consisting of 48,074 pulses sorted into clusters, is split into **training** and **test subsets** to avoid data leakage and ensure accurate model evaluation.

Creating X and y

d d c i h b h h c b d g b e e a a b c

X lookback = 5

y horizon = 2

The model utilizes **X**, the lookback window, as input to predict **y**, the forecast horizon. During training, X and y derived from the **training sequence** are used to optimize the model. The trained model is then evaluated using X and y from the **test sequence** to assess its predictive performance.

Creating X and y

d d c i h b h h c b d g b e e a a b c

X lookback = 5

y horizon = 2

The model utilizes **X**, the lookback window, as input to predict **y**, the forecast horizon. During training, X and y derived from the **training sequence** are used to optimize the model. The trained model is then evaluated using X and y from the **test sequence** to assess its predictive performance.

Creating X and y

d d c i h b h h c b d g b e e a a b c

X lookback = 5

y horizon = 2

The model utilizes **X**, the lookback window, as input to predict **y**, the forecast horizon. During training, X and y derived from the **training sequence** are used to optimize the model. The trained model is then evaluated using X and y from the **test sequence** to assess its predictive performance.

Overview of Models

We use different models for classification:

- **Random Forest** – Traditional machine learning
- **LSTM** – Captures sequential dependencies
- **GRU** – A lightweight alternative to LSTMs
- **CNN** – Extracts spatial features from sequences

Random Forest Classifier

How it works:

- Uses **200 decision trees**.
- Each tree makes a prediction, and the final classification is based on majority voting.

Why Random Forest?

- Simple to use and quick to train.
- A **large number of clusters** or **long horizon prediction** lead to excessive memory usage. Crashes on my laptop.
- Performs well when **temporal dependencies are minimal** and the data is not highly sequential.

LSTM Model

Why LSTM (Long Short-term Memory)?

- Designed for **sequential data**.
- Uses **memory cells** to retain long-term dependencies.
- Optimized using the **Adam** optimizer and categorical cross-entropy loss.
- Ideal for learning and predicting **long-term temporal dependencies** in sequential patterns.

LSTM Model Architecture

LSTM-based Classification Model:

```
model = Sequential()
model.add(LSTM(64, return_sequences=True,
               input_shape=(self.look_back,self.nclusters)))
model.add(Dropout(0.2))

model.add(LSTM(32, return_sequences=False))
model.add(Dropout(0.2))

model.add(Dense(64, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(self.nclusters*self.hp, activation='softmax'
    ))
model.add(Reshape((self.hp, self.nclusters)))

optimizer = Adam()
model.compile(optimizer=optimizer,
    loss='categorical_crossentropy', metrics=['accuracy'])
```

GRU Model

Why GRU (Gated Recurrent Units)?

- A **simpler alternative** to LSTMs with fewer parameters.
- Captures sequential patterns while being more computationally efficient.
- Optimized using Adam optimizer and categorical cross-entropy loss.
- Suitable for modeling **long-term temporal dependencies**, similar to LSTM but with reduced computational overhead.

CNN Model

Why CNN (Convolutional Neural Network)?

- Extracts spatial features from sequential data.
- Uses **three 1D convolutional layers** with max pooling.
- Optimized using Adam optimizer and categorical cross-entropy loss.
- Best suited for time-series data with **strong local patterns** and **short lookback windows**, as CNNs excel at learning local dependencies in sequences.

Model Evaluation Metrics

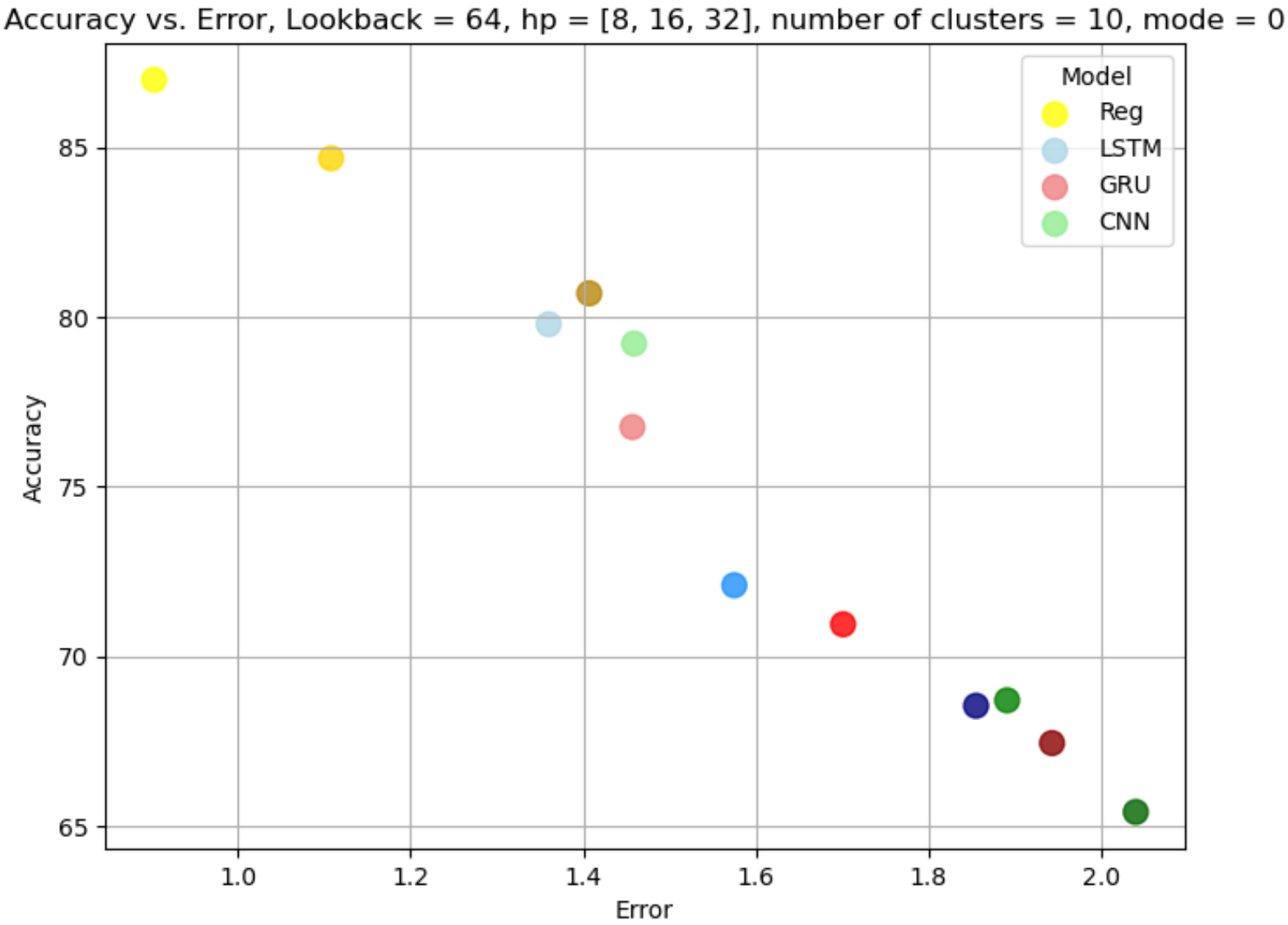
Accuracy – Represents the percentage of correctly classified pulses. This metric is typically optimized during model training.

Mean Difference – Calculates the average difference between each predicted centroid and the corresponding true pulse. (Error)

Prediction Plot – Provides a visual comparison between the true pulse, the true centroid, and the predicted centroid, offering a subjective assessment of model performance.

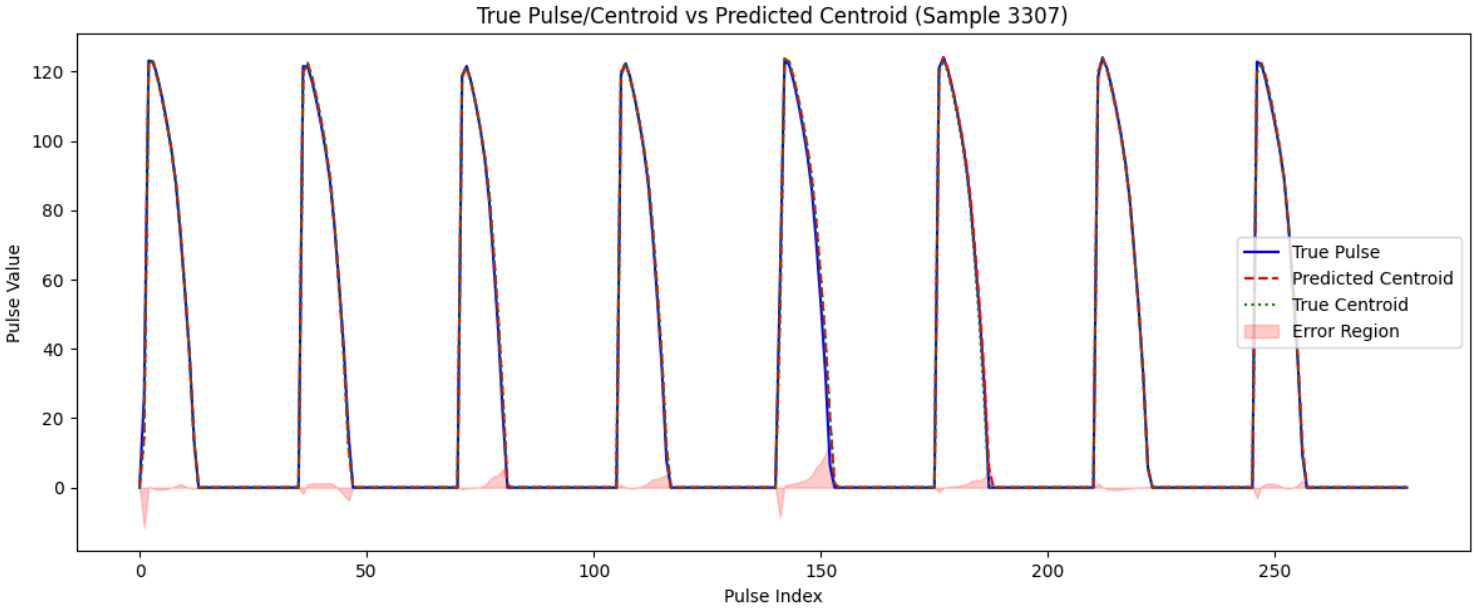
Error Plot – Shows a side-by-side comparison of the true pulse, true centroid, and predicted centroid, with an additional visualization of the error between the predicted centroid and the true pulse.

Accuracy vs Error

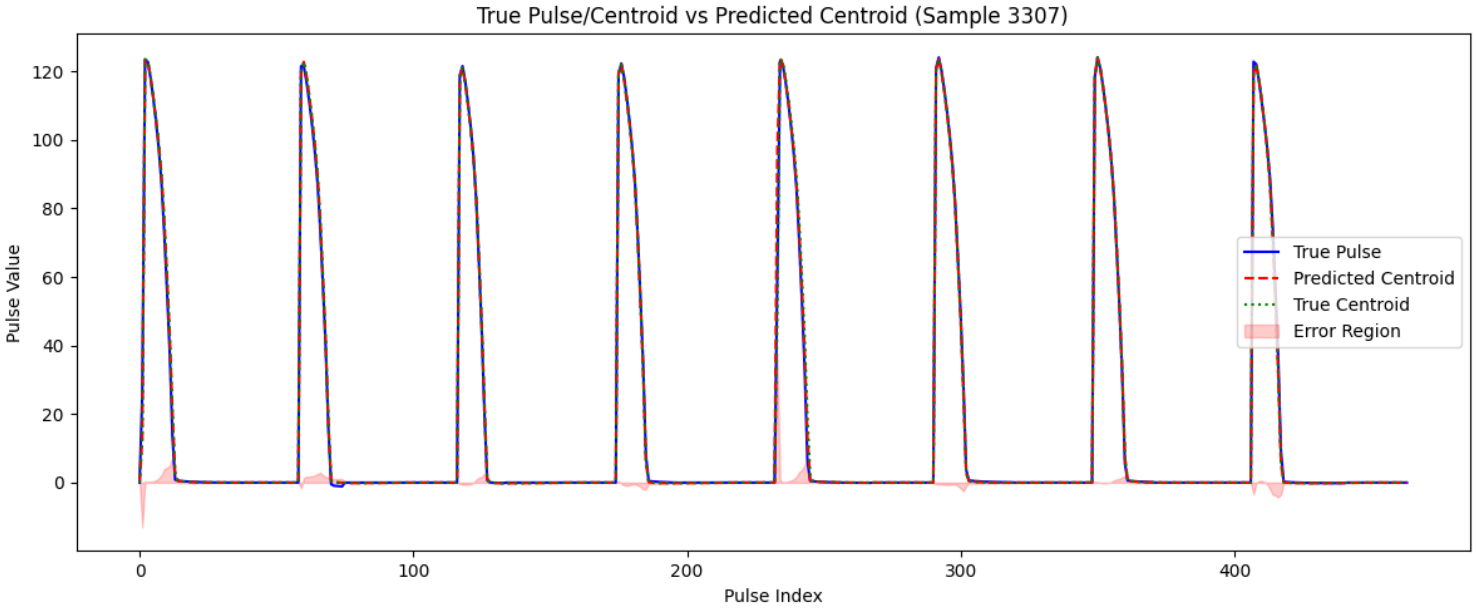


Mode 0 and Mode 1

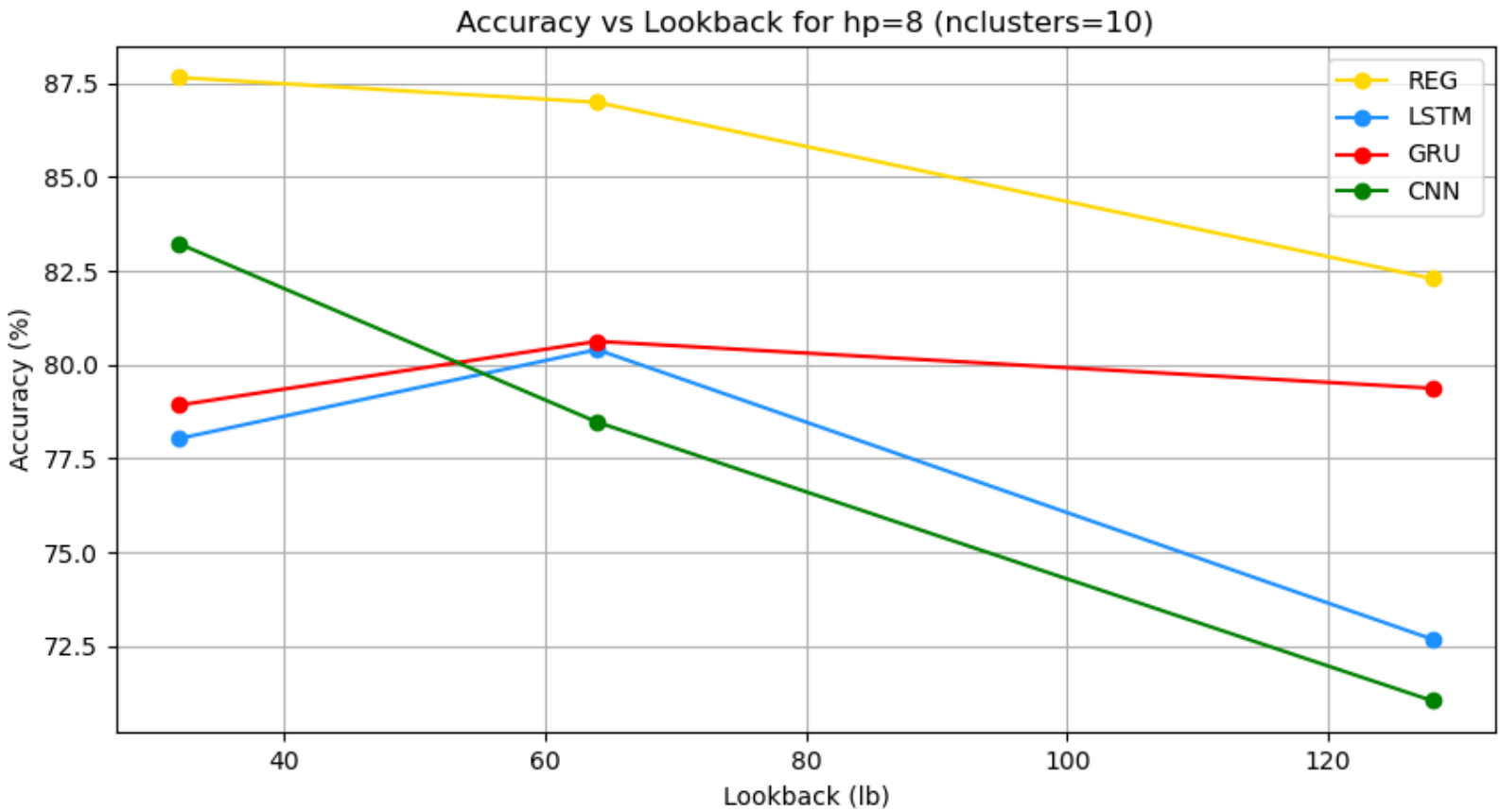
Mode 0
83.80%



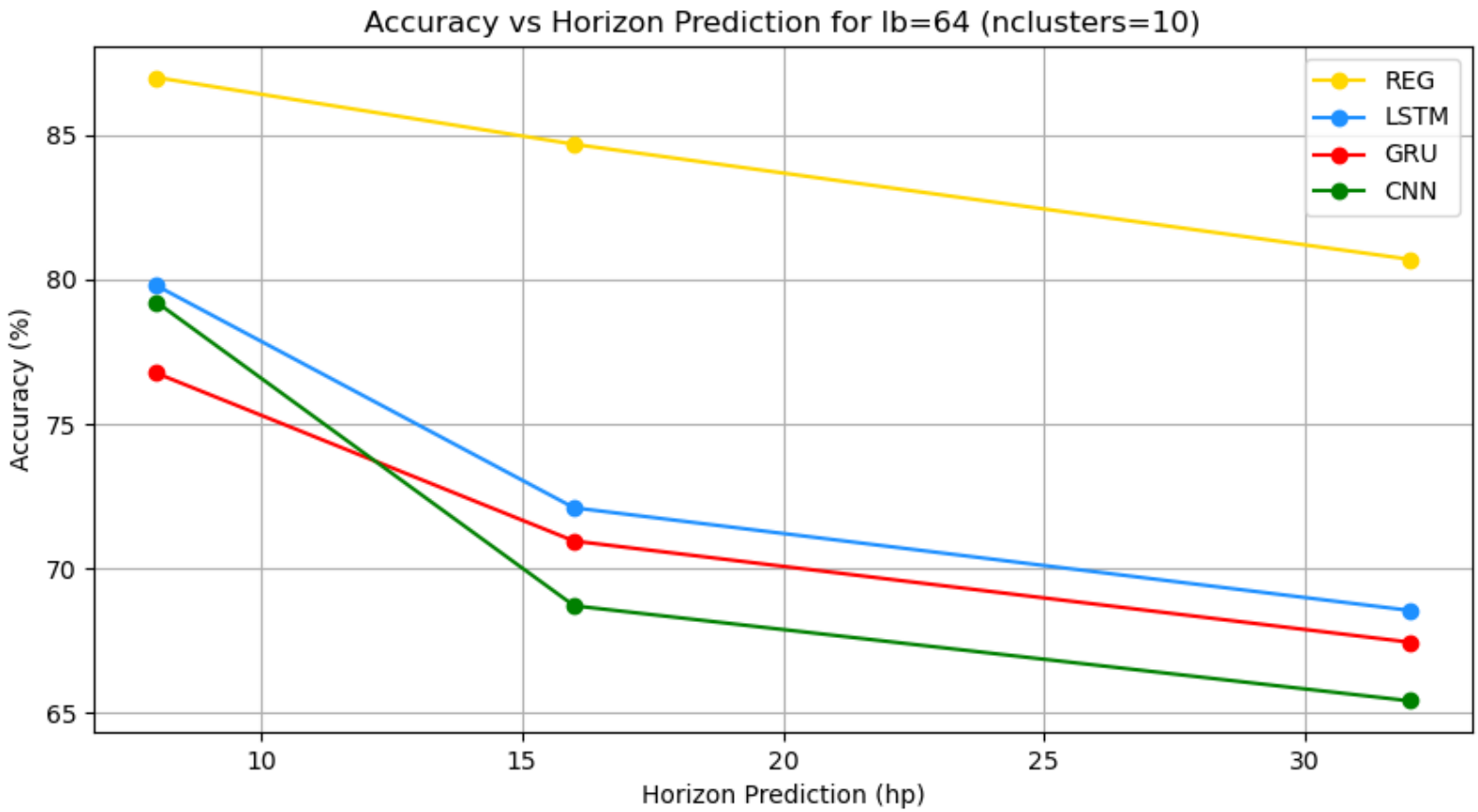
Mode 1
81.69%



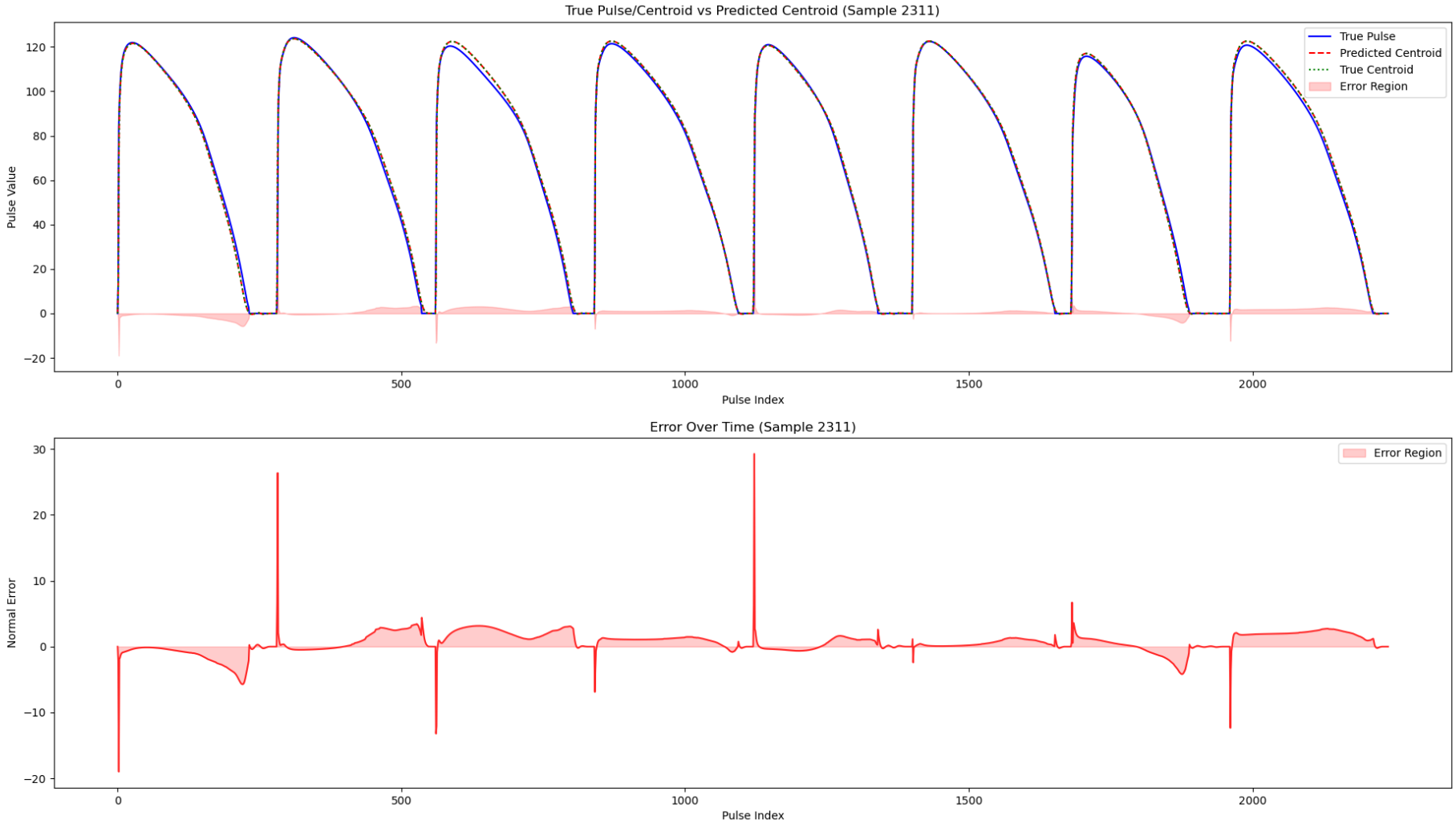
Accuracy vs Lookback



Accuracy vs Horizon Prediction

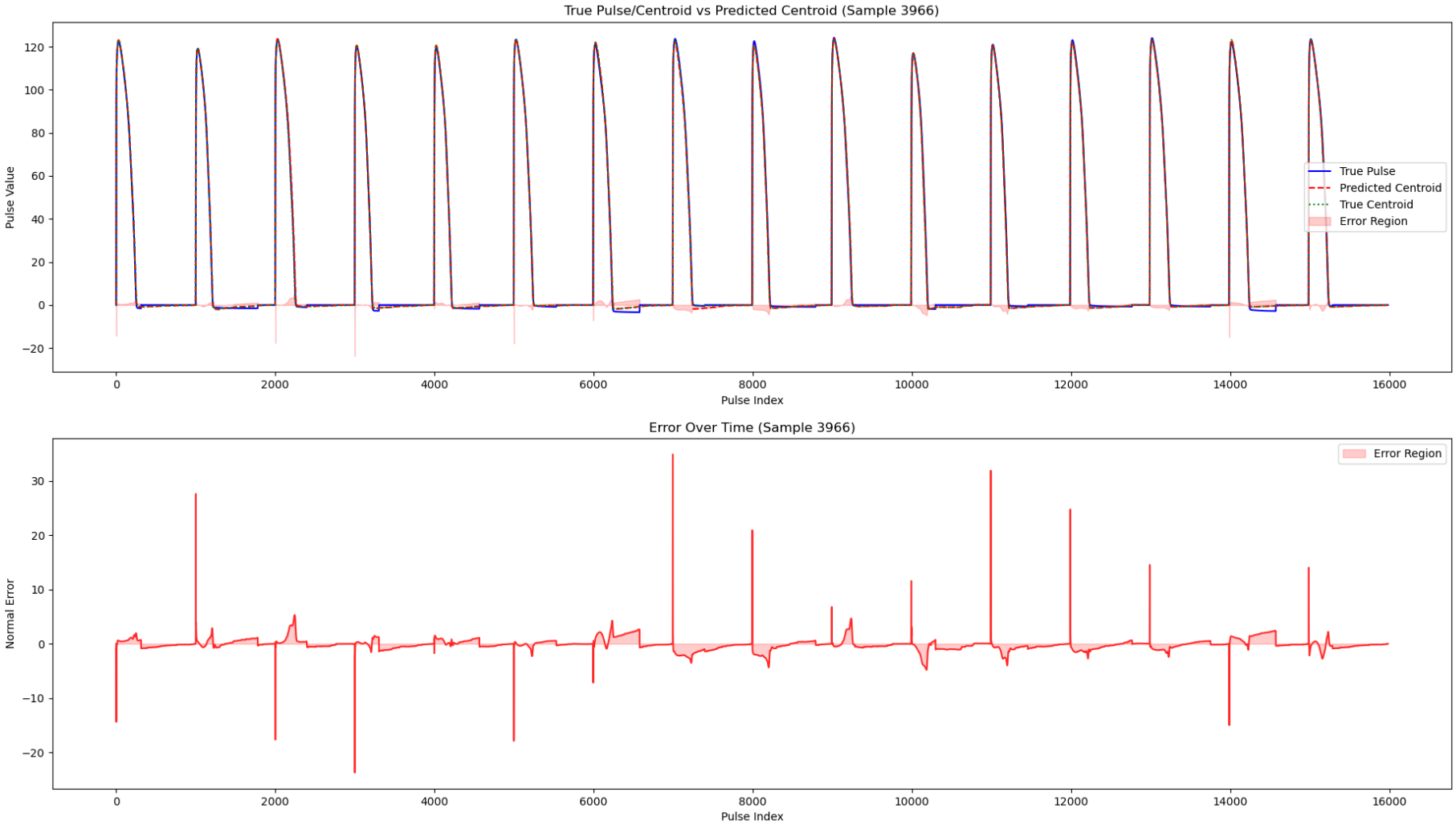


Random Forest lb=64, hp=8, mode=0, nclusters=10



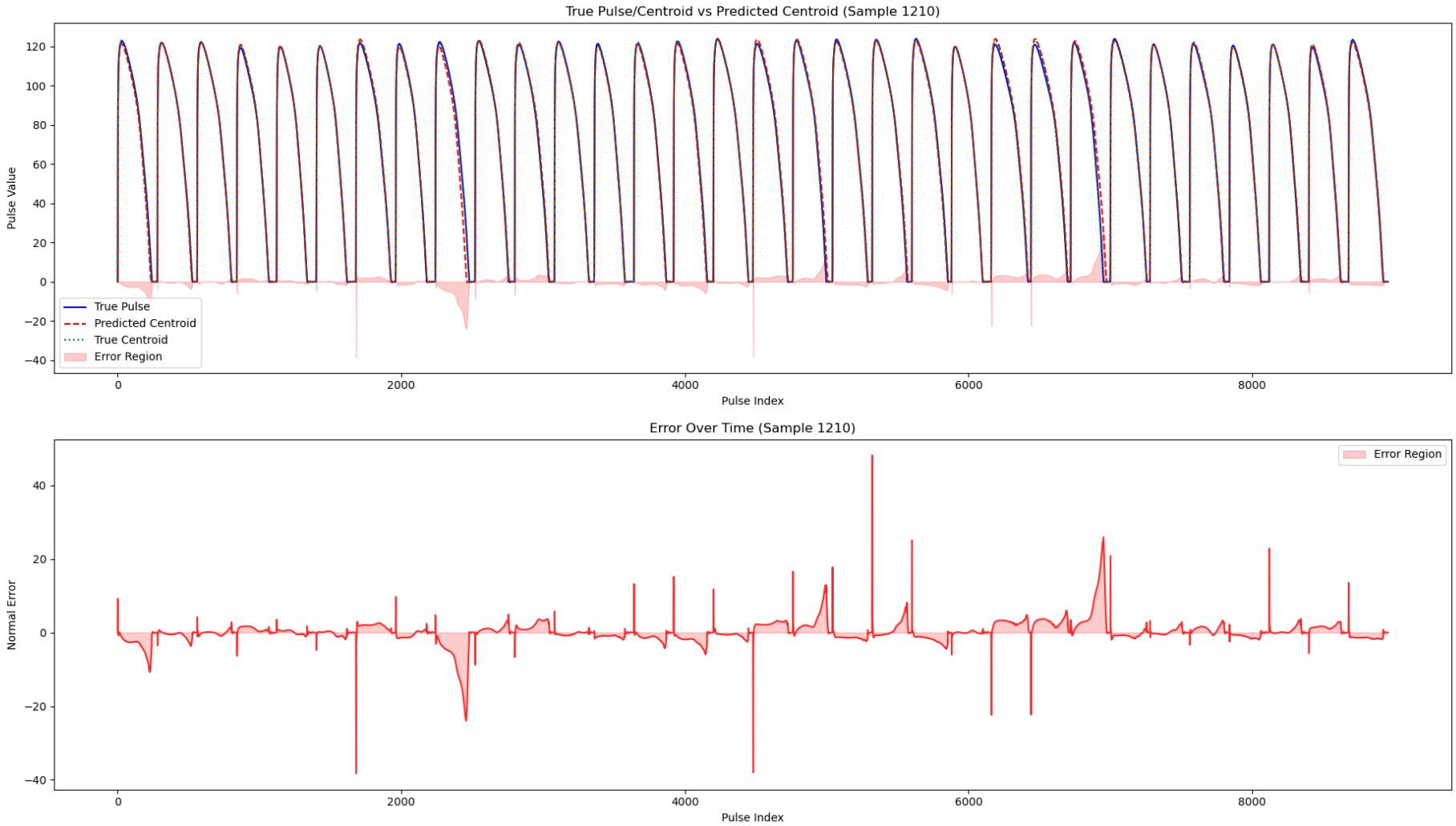
86.99%

GRU lb=64, hp=16, mode=1, nclusters=10



68.66%

LSTM lb=64, hp=32, mode=0, nclusters=20



36.08%

Table Random Forest Results

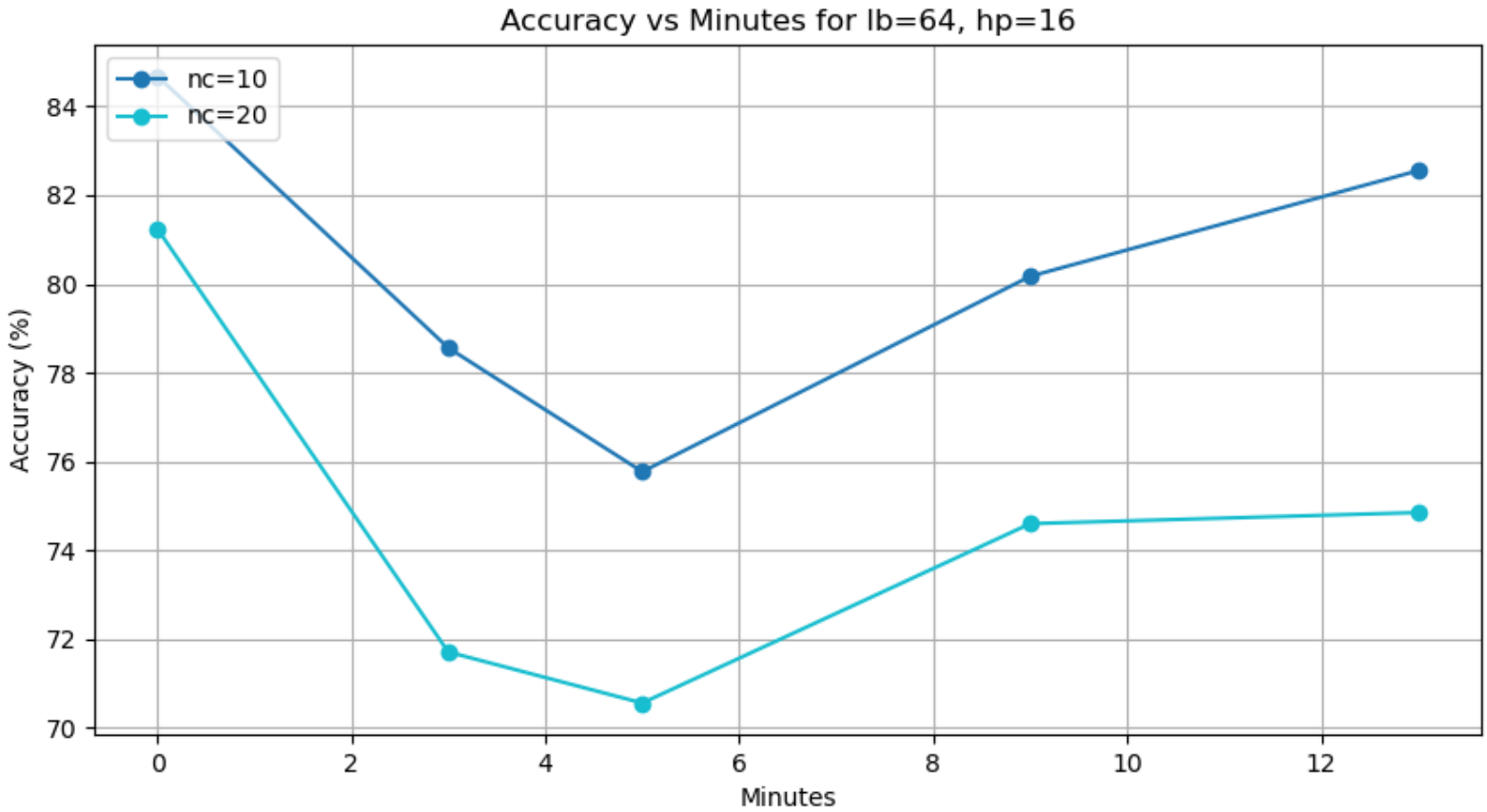
Table: Random Forest Performance for Different Combinations of Horizon Prediction (hp), Modes, and Clusters (nclusters)

Hyperparameters		Mode 0			Mode 1		
hp	*lb*	Acc	Err	Rec	Acc	Err	Rec
Number of Clusters = 10							
8	64	86.99	0.903		86.99	0.613	
16	64	84.68	0.942	3.53	85.25	0.626	1.48
32	64	80.70	1.007		81.76	0.661	
Number of Clusters = 20							
8	64	83.80	0.747		81.69	0.521	
16	64	81.24	0.793	1.93	79.55	0.543	0.96
32	64	76.61	0.867		75.72	0.580	

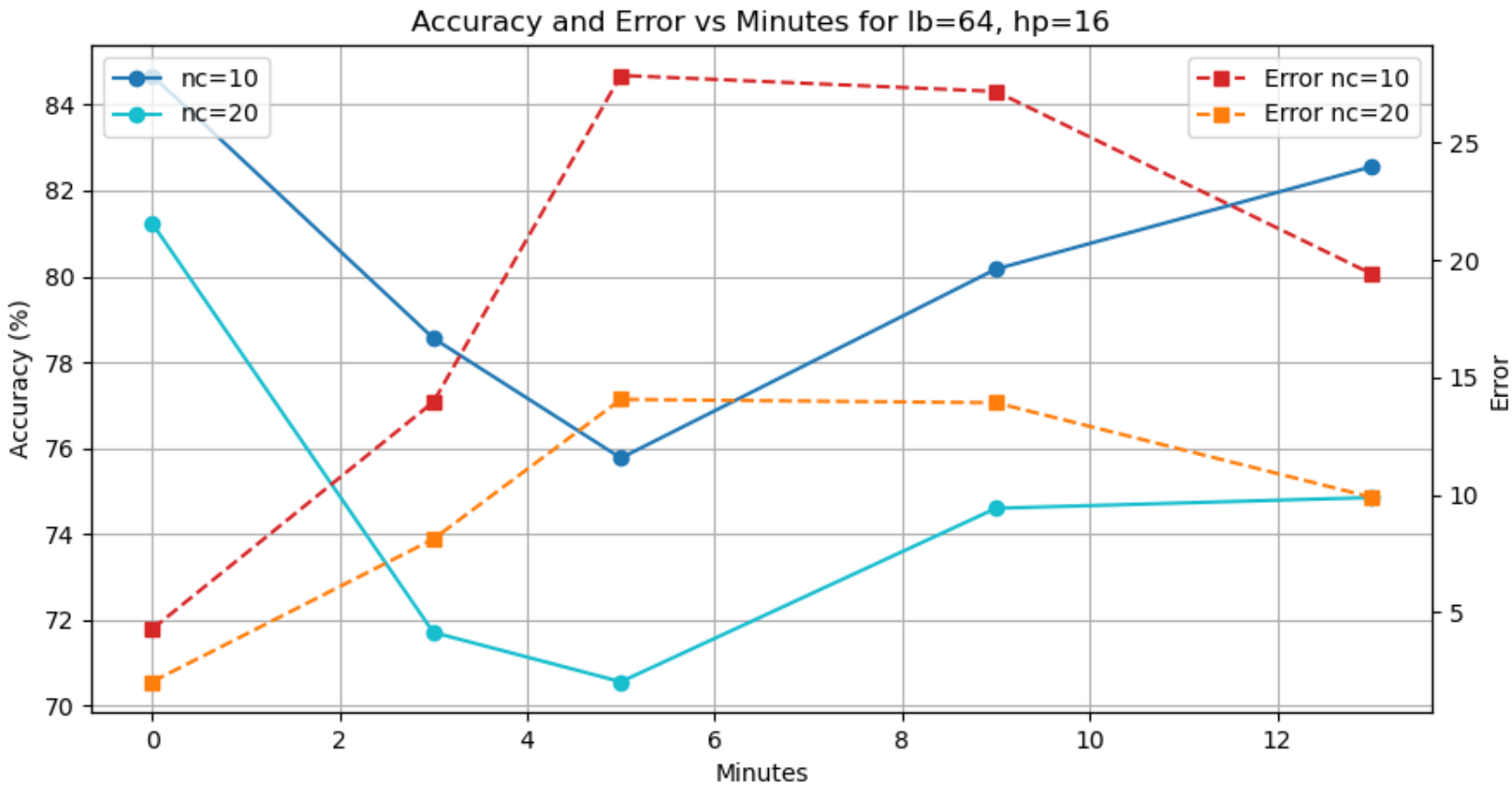
Conclusions

- Accuracy and Error are inversely related.
- Mode 0 and Mode 1 have similar performances.
- A lookback of 64 yields the best results, while 128 often reduces performance.
- Higher Horizon Prediction (HP) generally lowers Accuracy (e.g. around 7% drop for Random Forest with 20 clusters).
- Random Forest performs the best with accuracy ranges from 90.66% (lookback = 64, hp = 8, mode = 0, nclusters = 10) to 74.51% (lookback = 128, hp = 8, mode = 1, nclusters = 20).
- Using Random Forest, increasing the number of clusters reduces accuracy but results in a lower mean difference from true pulses, due to a decrease in reconstruction error.

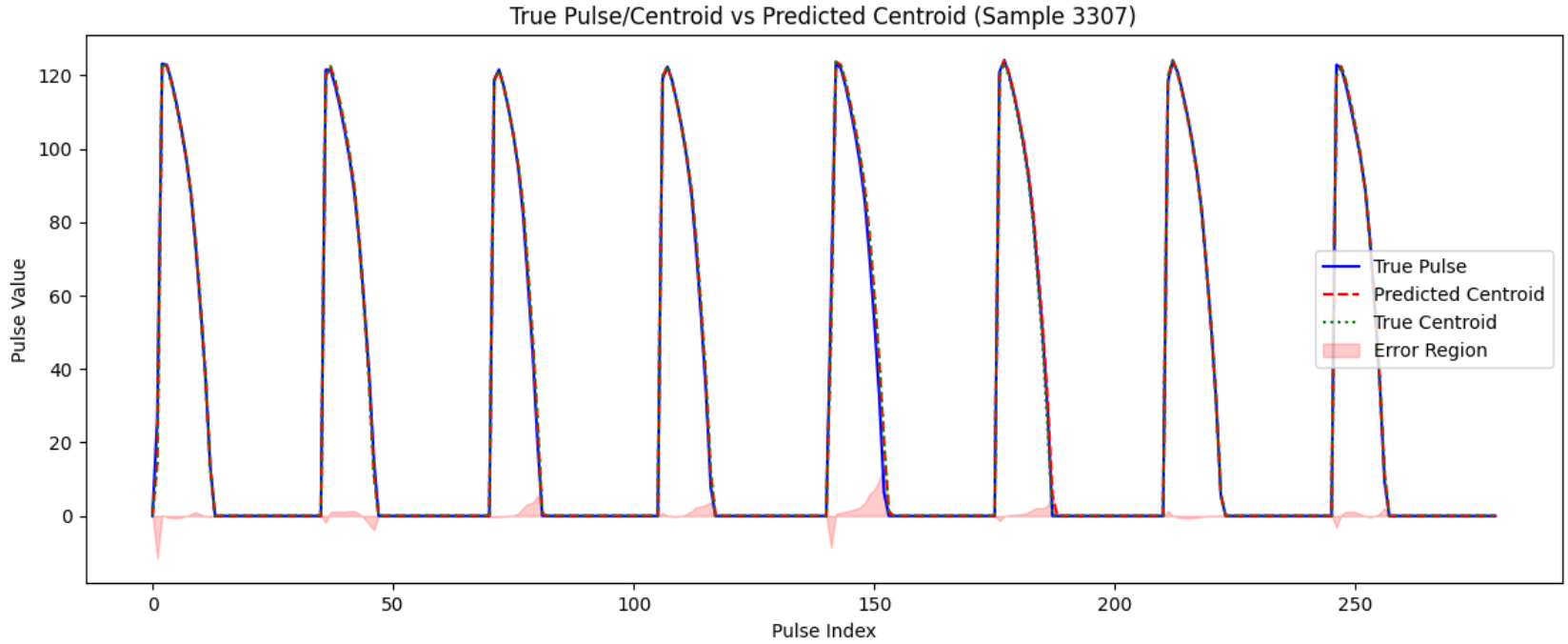
Accuracy vs Minute

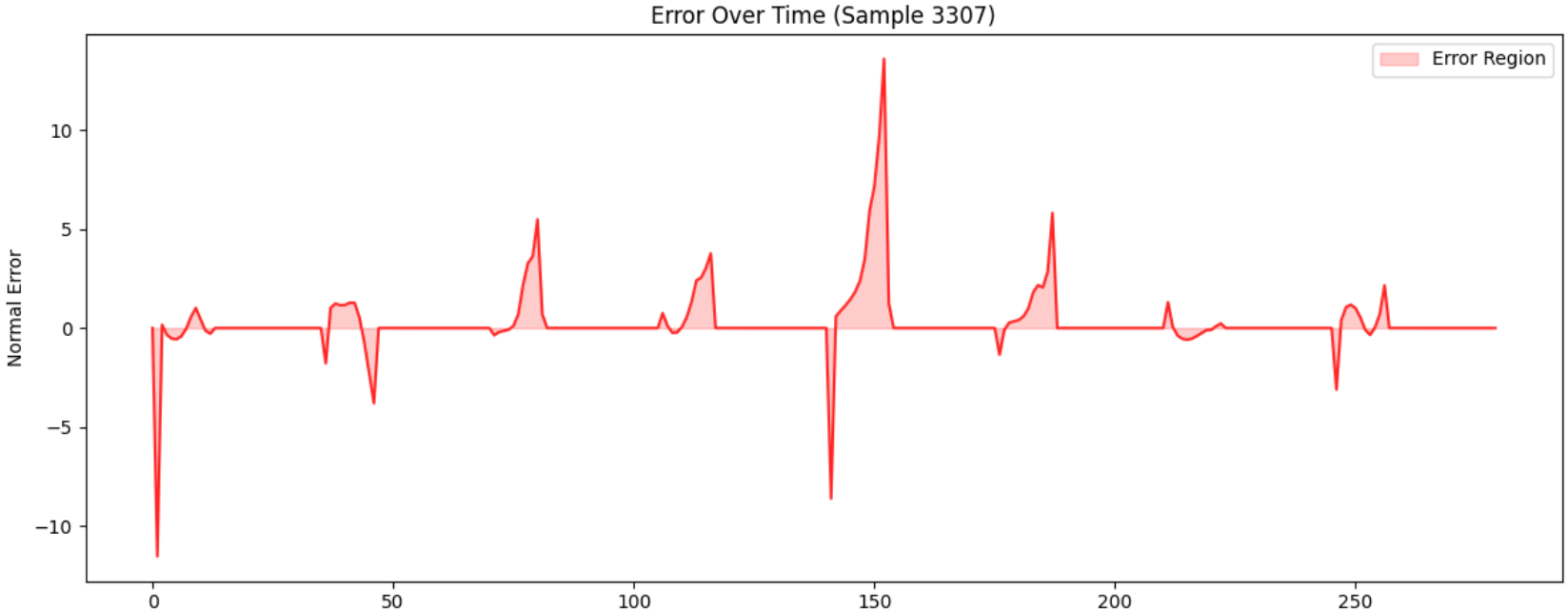


Accuracy and Reconstruction Error vs Minute

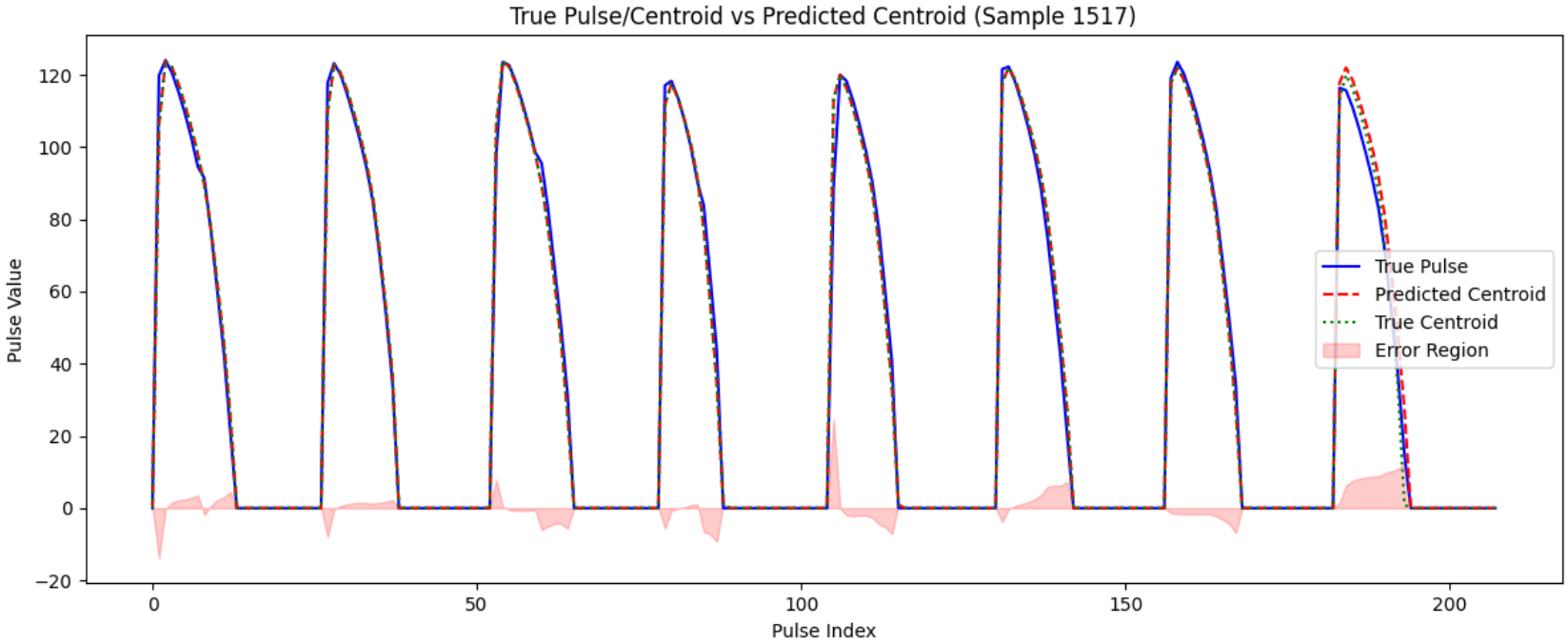


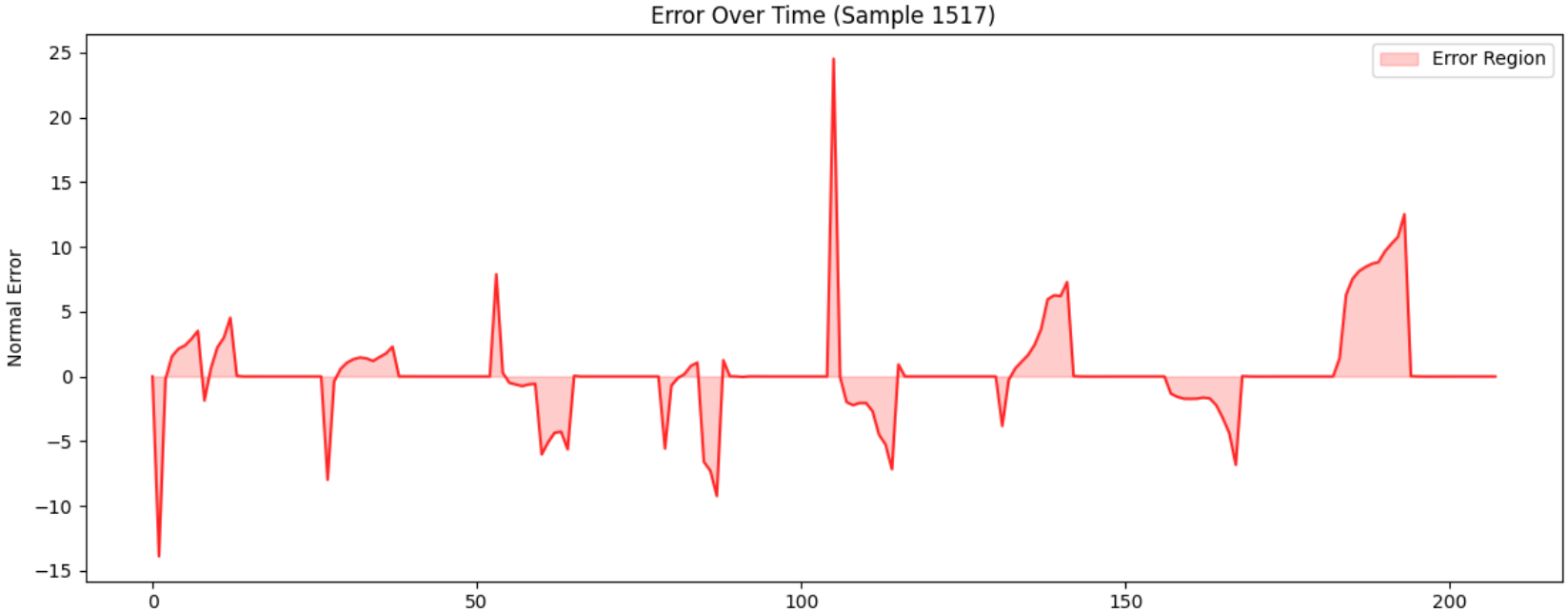
Minute 0 **78.69%**



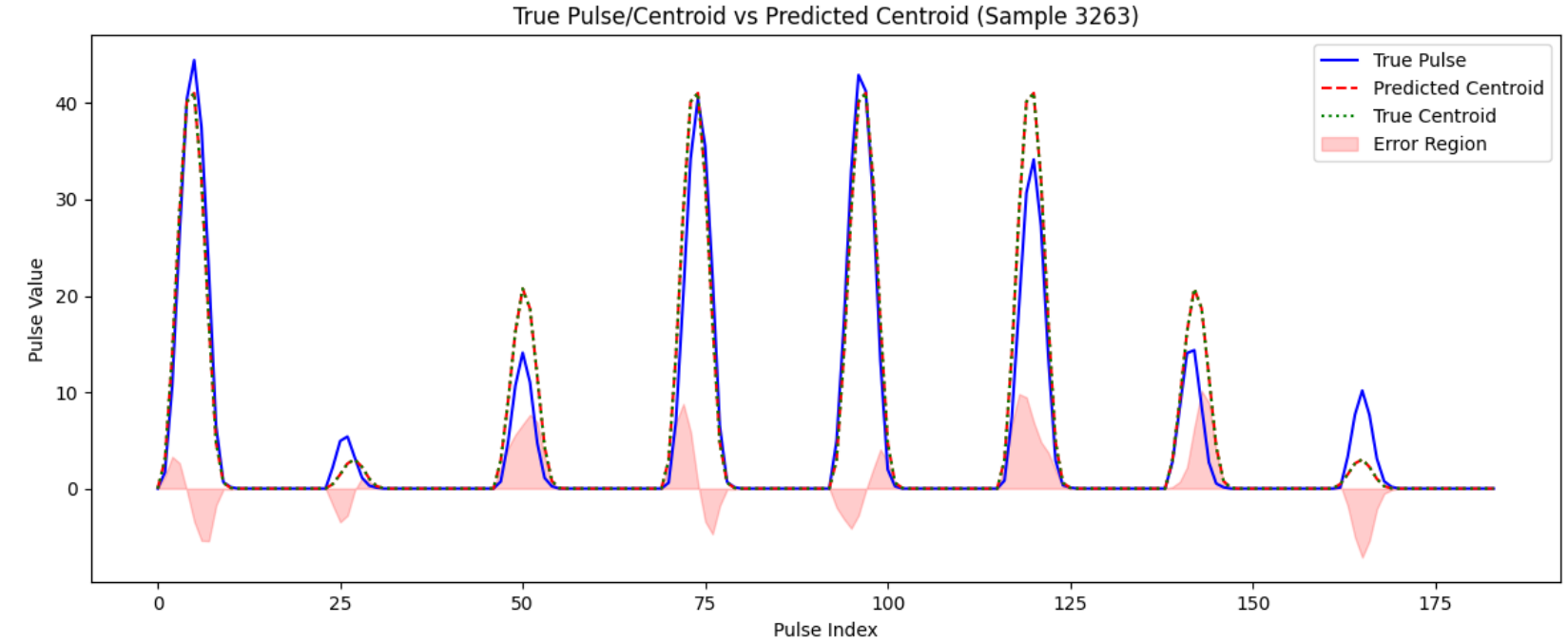


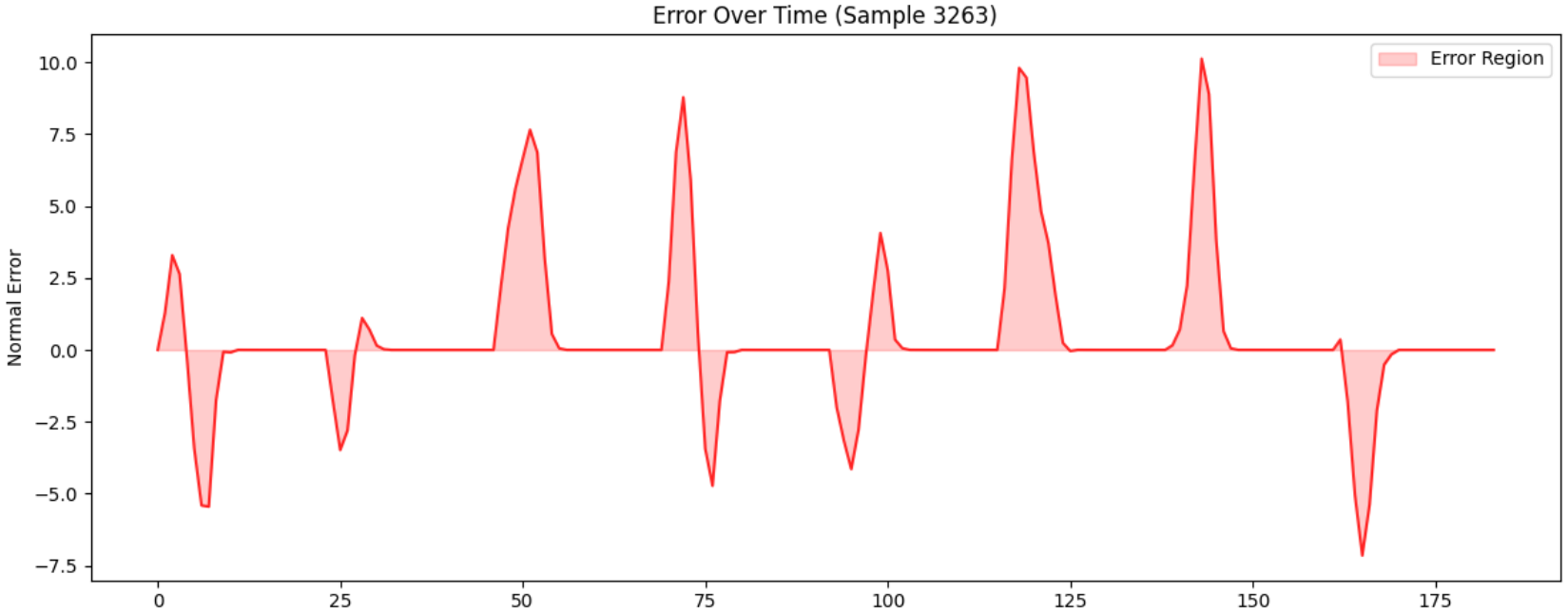
Minute 5 **70.48%**





Minute 13 **73.43%**





Mode Comparison

Table: Mean Accuracy of the Modes for Random Forest

Minute	Mode 0	Mode 1
0	83.23	82.25
3	75.57	76.06
5	74.22	73.21
9	77.71	78.84
13	79.23	81.99

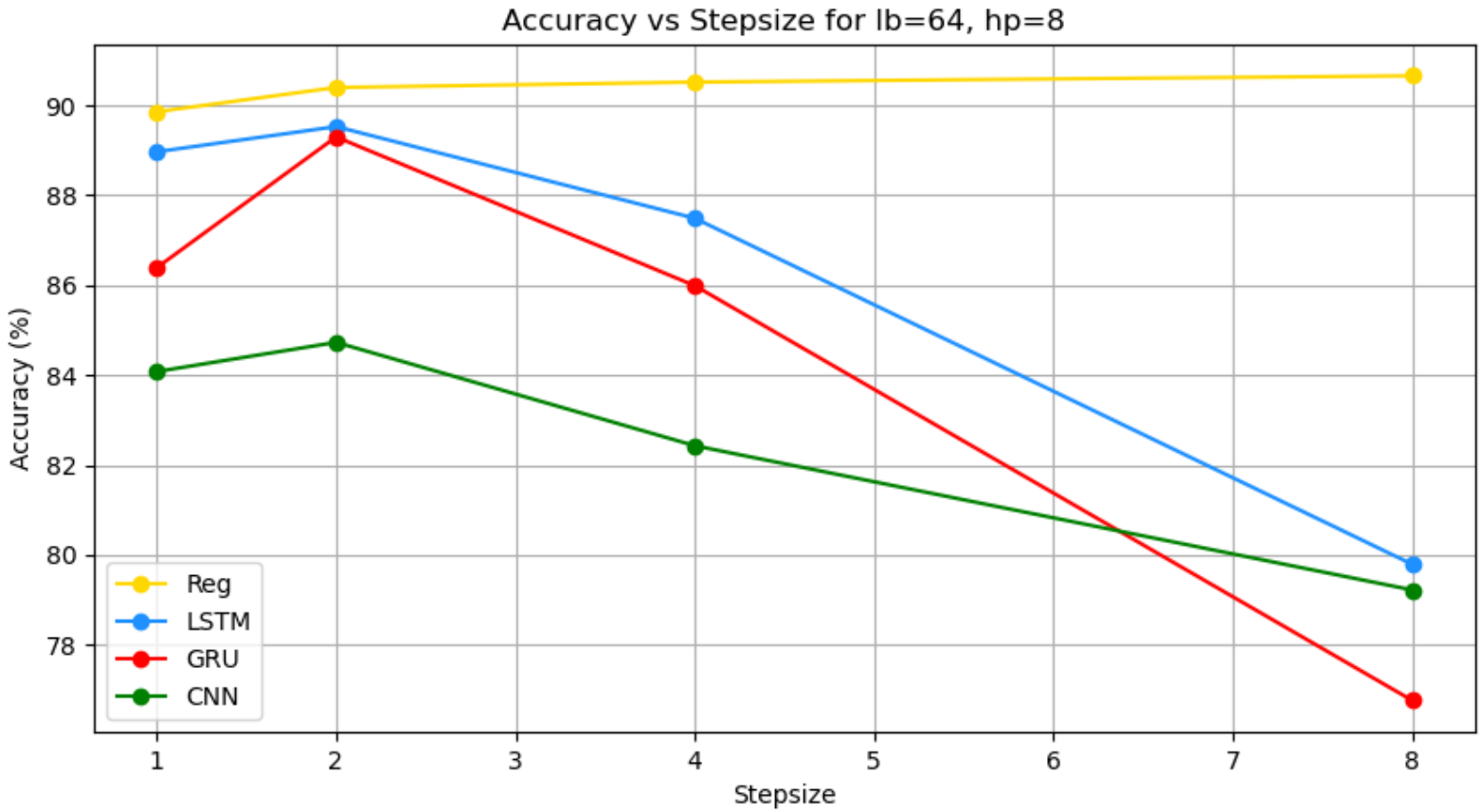
Conclusions

- Accuracy alone is not sufficient for comparison across minutes, as reconstruction error varies significantly.
- Increasing the number of clusters can help reduce reconstruction error.
- The highest reconstruction error is observed at minute 5.
- At minute 13, predictions exhibit high errors, though the overall shape is still captured well.
- The Modes have similar performances for all the Minutes.

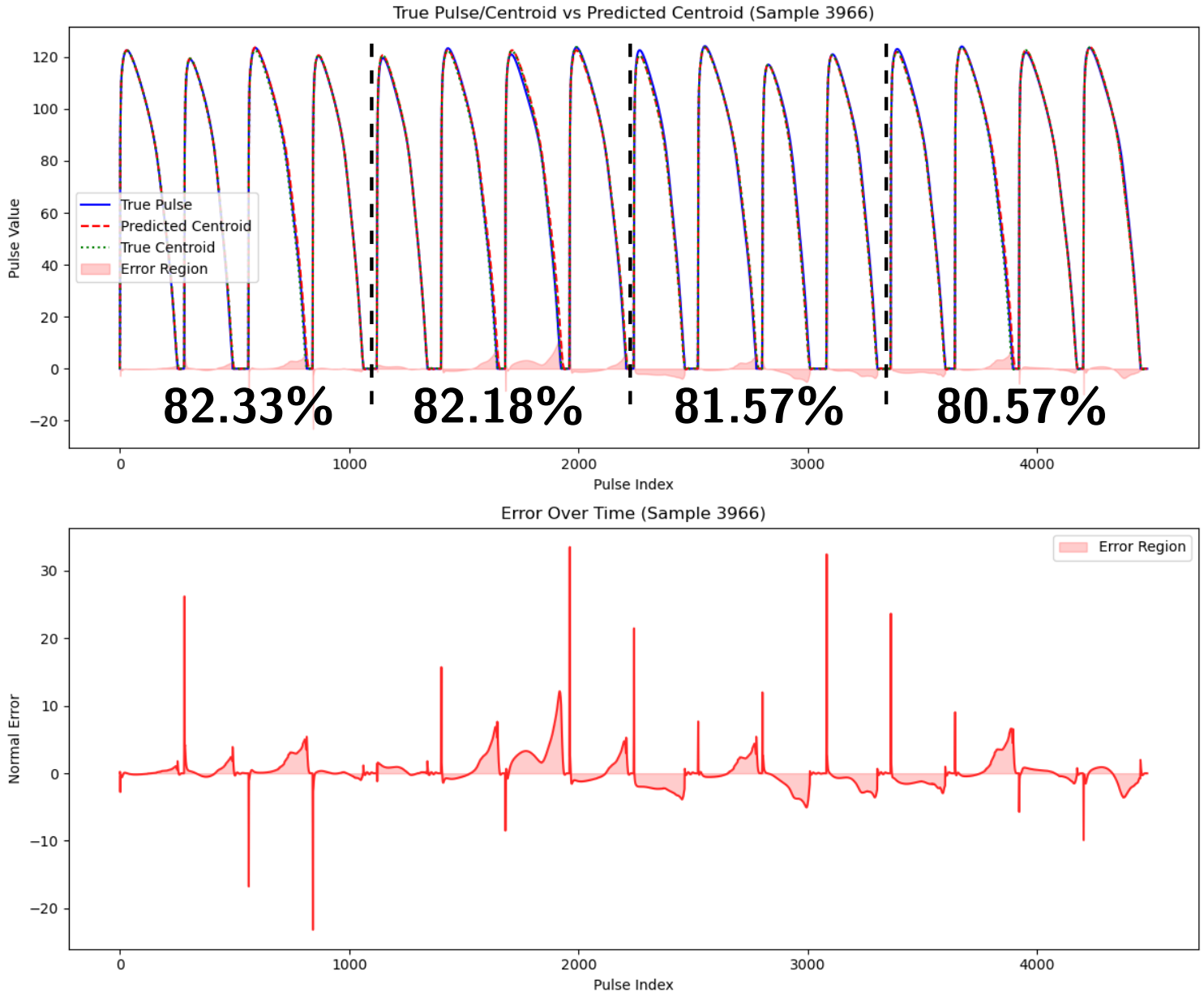
Stepwise Prediction for Horizon Estimation

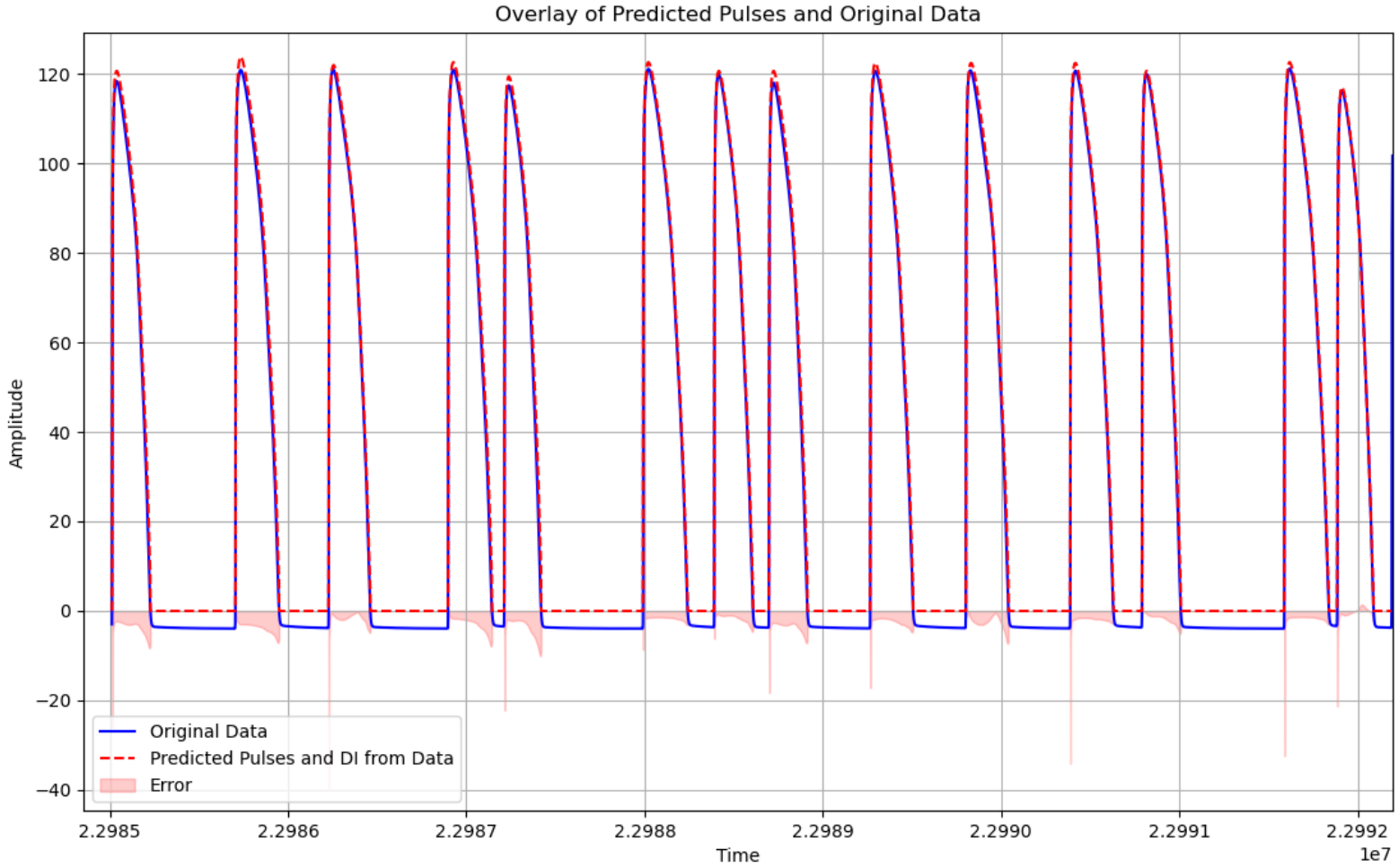
- **Stepwise Prediction** involves predicting the horizon step by step. These steps can be one or multiple pulses.
- Unlike the original horizon prediction, which predicts all steps in one go, the stepwise technique iterates and uses the previous step's prediction as input for the next.
- **Advantages**: Can predict further into the future without increasing the model size and thereby reduce the need to use more training samples.
- **Disadvantages**: Using a prediction as input can increase error propagation.

Accuracy vs Stepsize



Stepwise Prediction




Overlay of Predicted Pulses and Original Data
Amplitude
Time
1e7
Original Data
Predicted Pulses and DI from Data
Error

Thank you!

Baseline

Baseline Approach:

- The baseline model predicts the most frequent centroid in the training set as the predicted centroid for the entire horizon.
- **For nclusters = 10:** The most frequent centroid is 1, which represents **17.66%** of all pulses in the training set.
- **For nclusters = 20:** The most frequent centroid is 8, which represents **8.54%** of all pulses in the training set.

Performance:

- For **nclusters = 10**, the mean accuracy is **16.48%** and the mean difference between the true pulses and predicted centroids is **12.626**.
- For **nclusters = 20**, the mean accuracy is **8.60%** and the mean difference between the true pulses and predicted centroids is **11.661**.

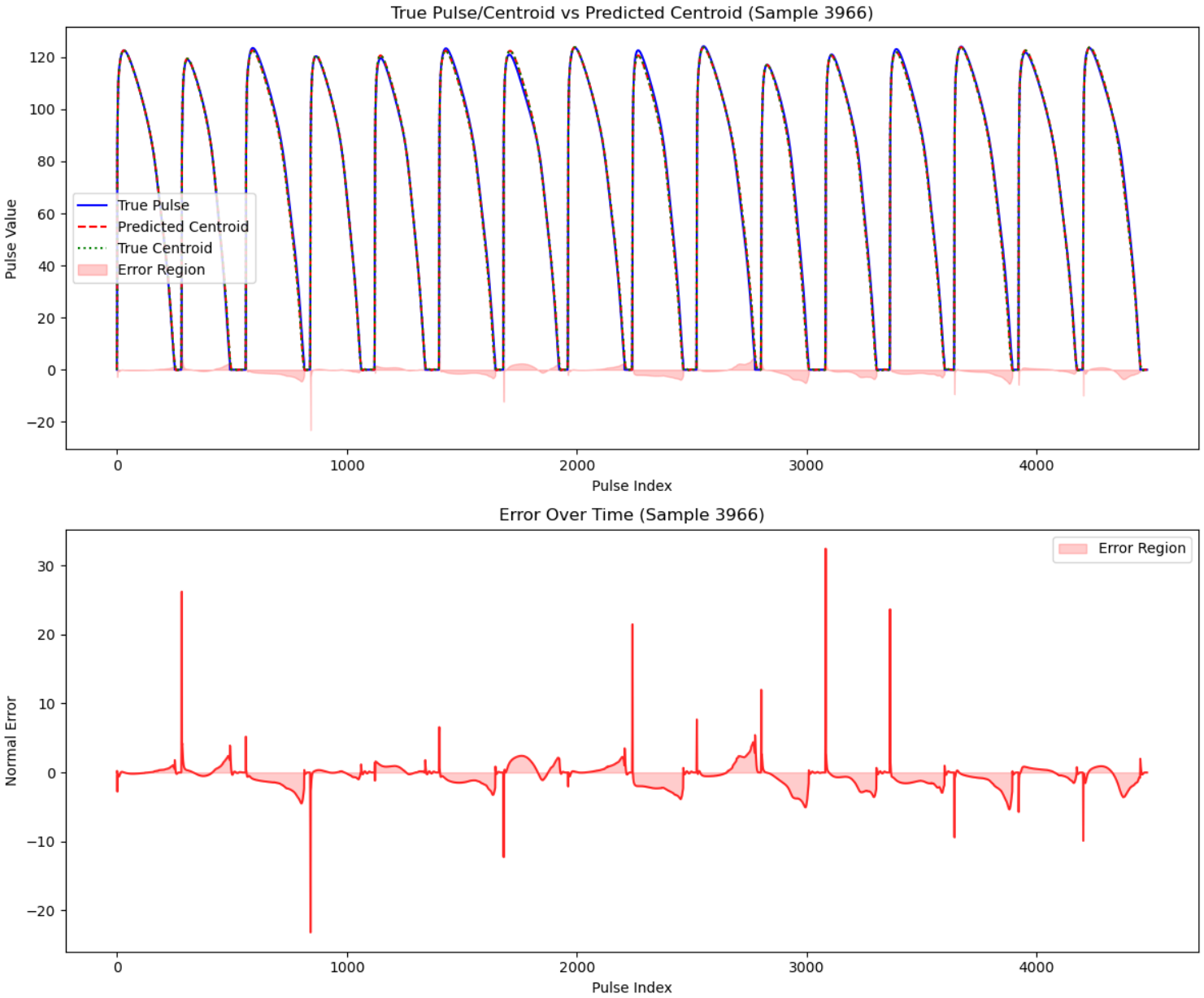
Conclusions

- Models: RandomForest at the best
- lookback: influence depends on the model, randomforest back with long lookback, lstm rather independed
- hp: long horizon makes prediction worse
- overall: quite good prediction for the rather easy case of min0

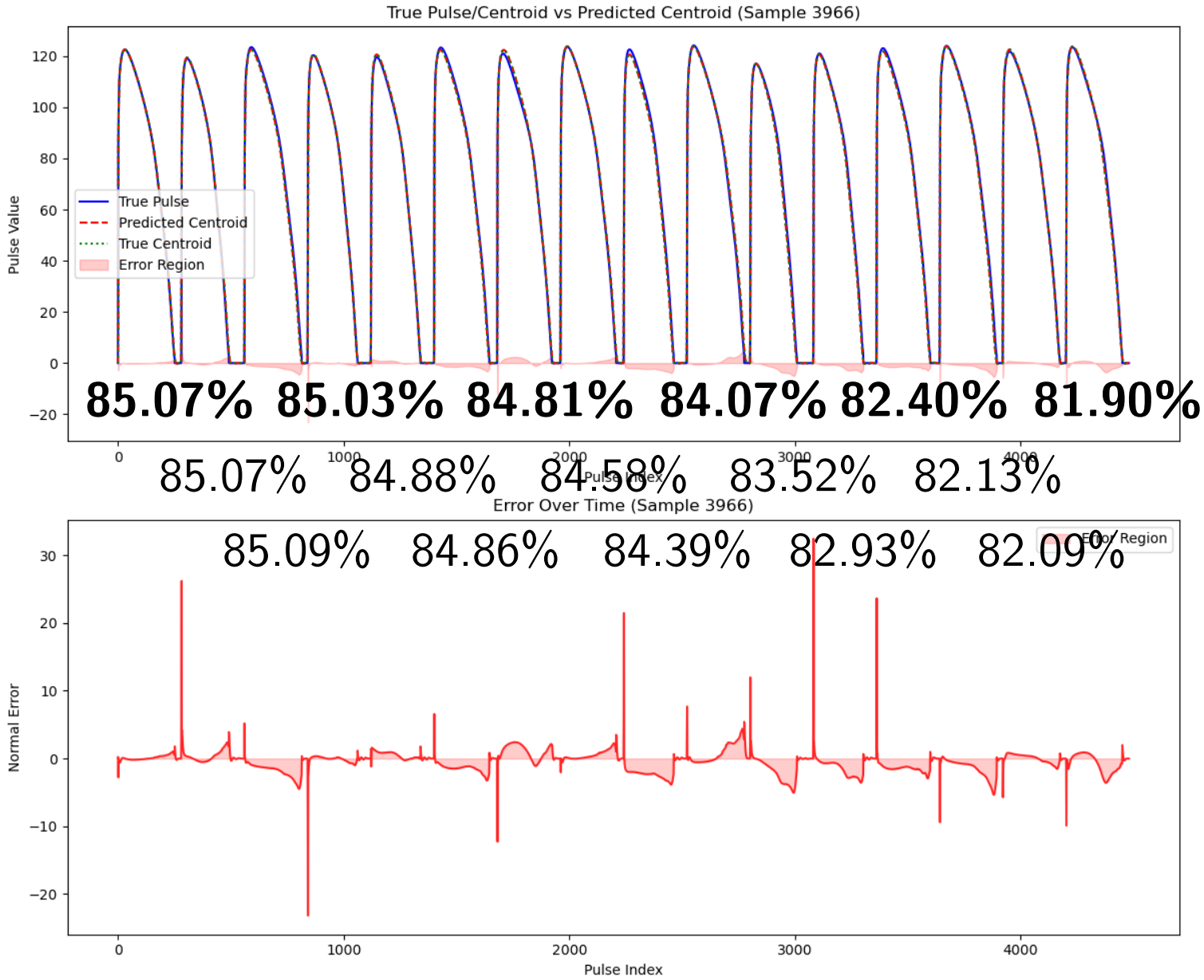
Thank you

Thank you for your attention!

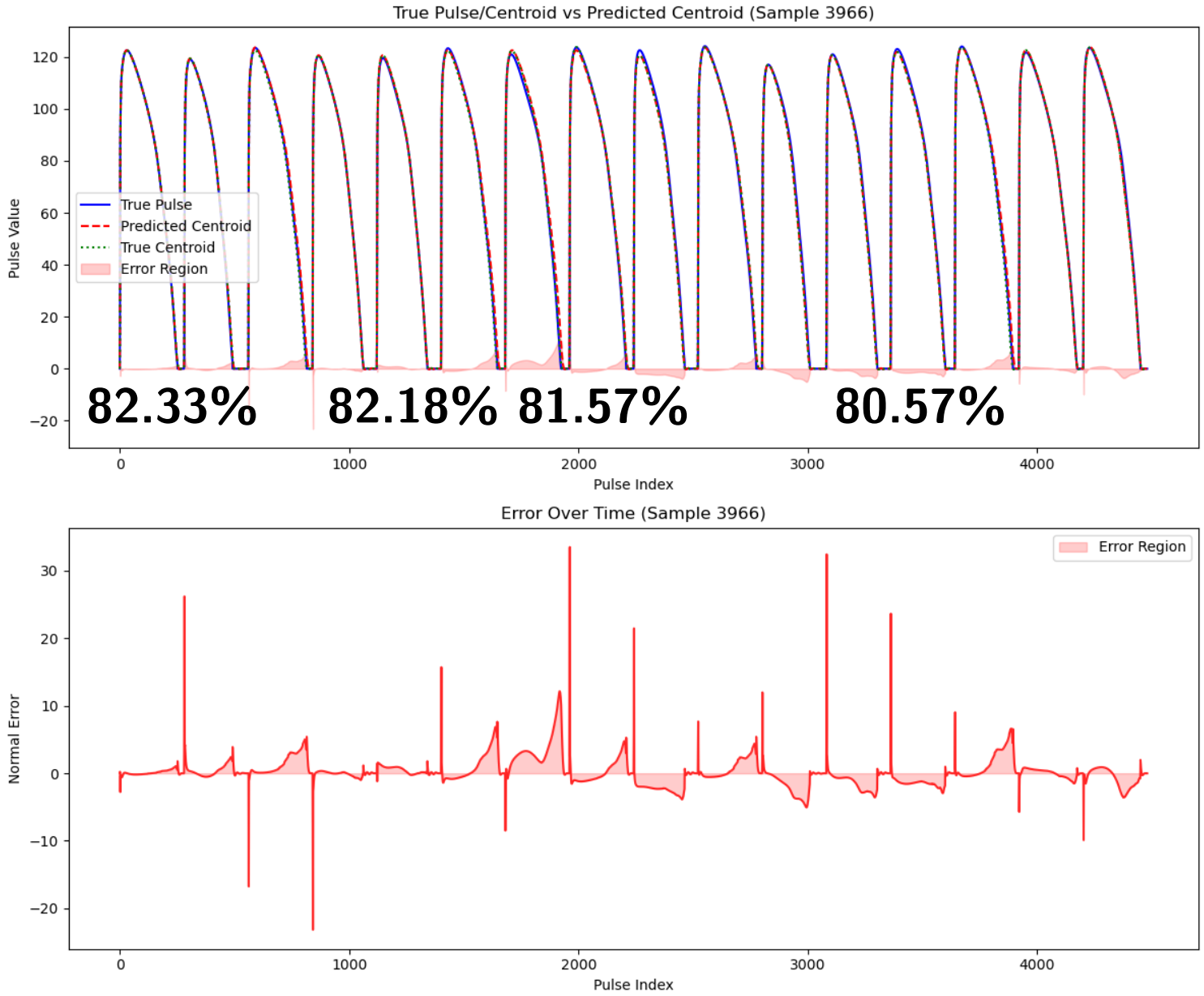
stepwise Prediction



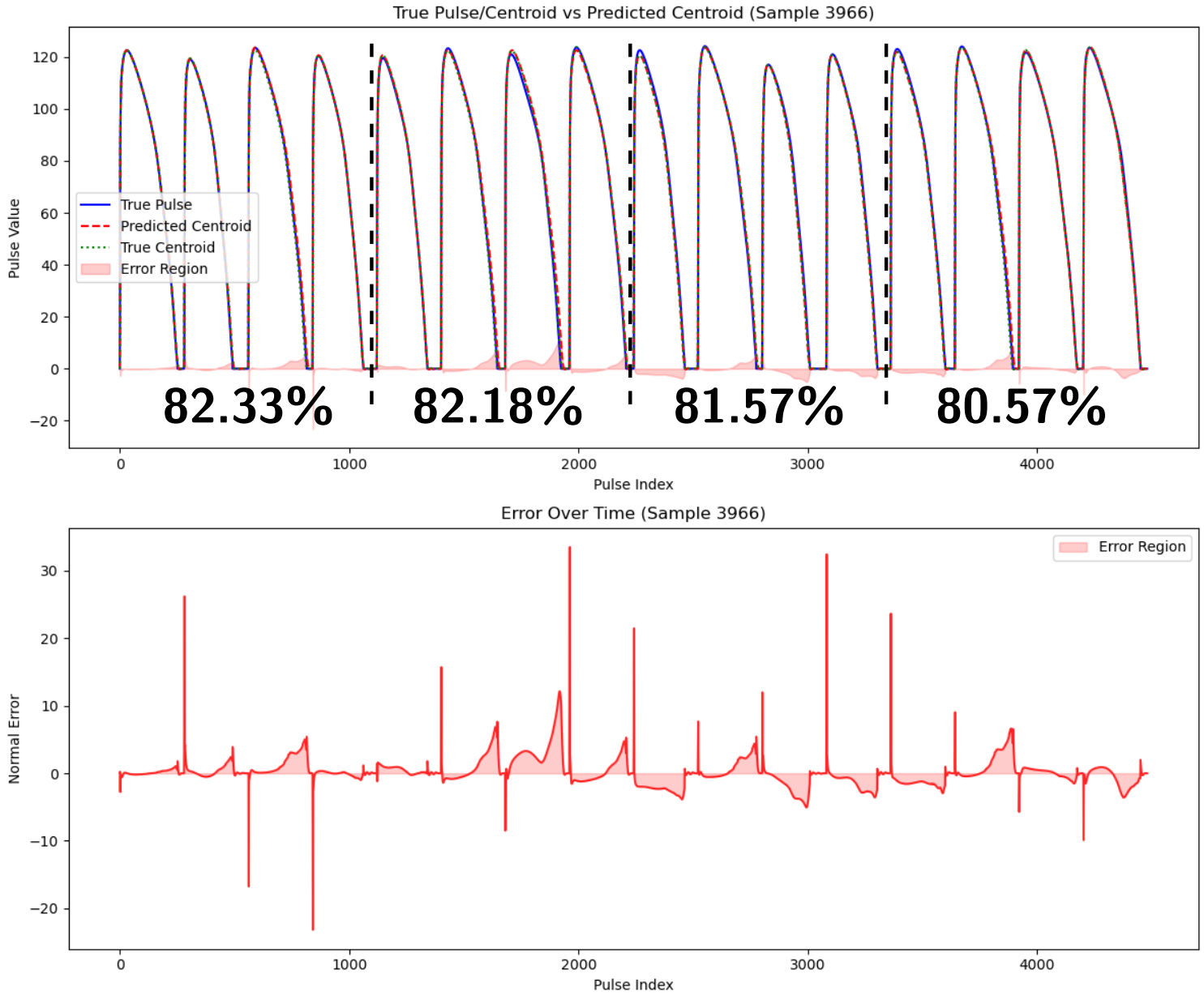
stepwise Prediction



stepwise Prediction



stepwise Prediction



stepwise Prediction Accuracy

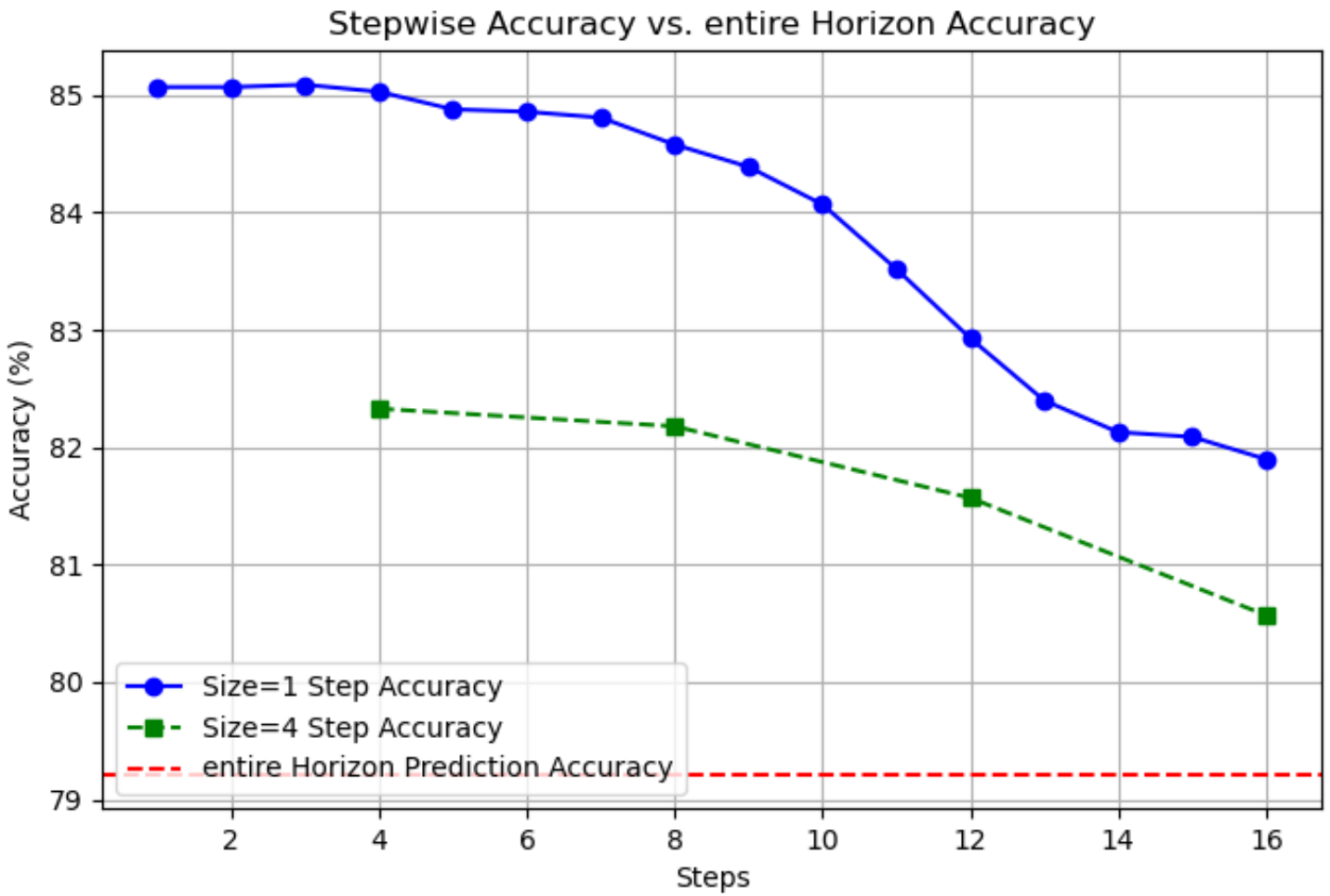


Figure: lb = 64, hp = 16, stepsize = 1&4, mode = 0, Model = CNN

Test

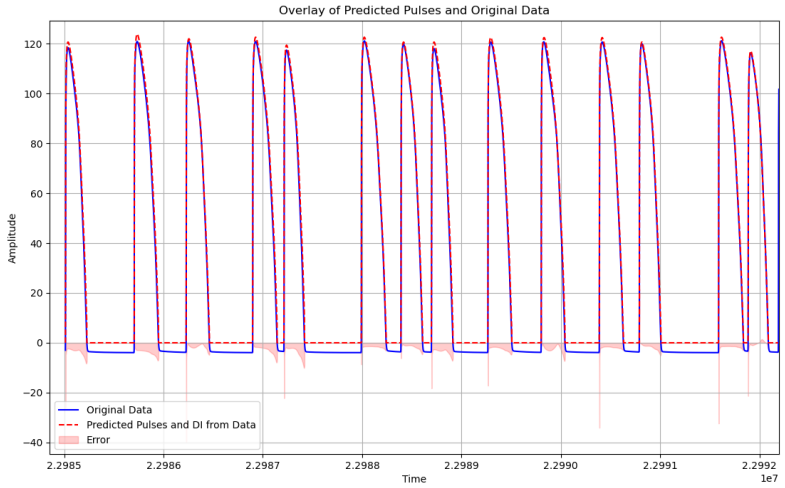
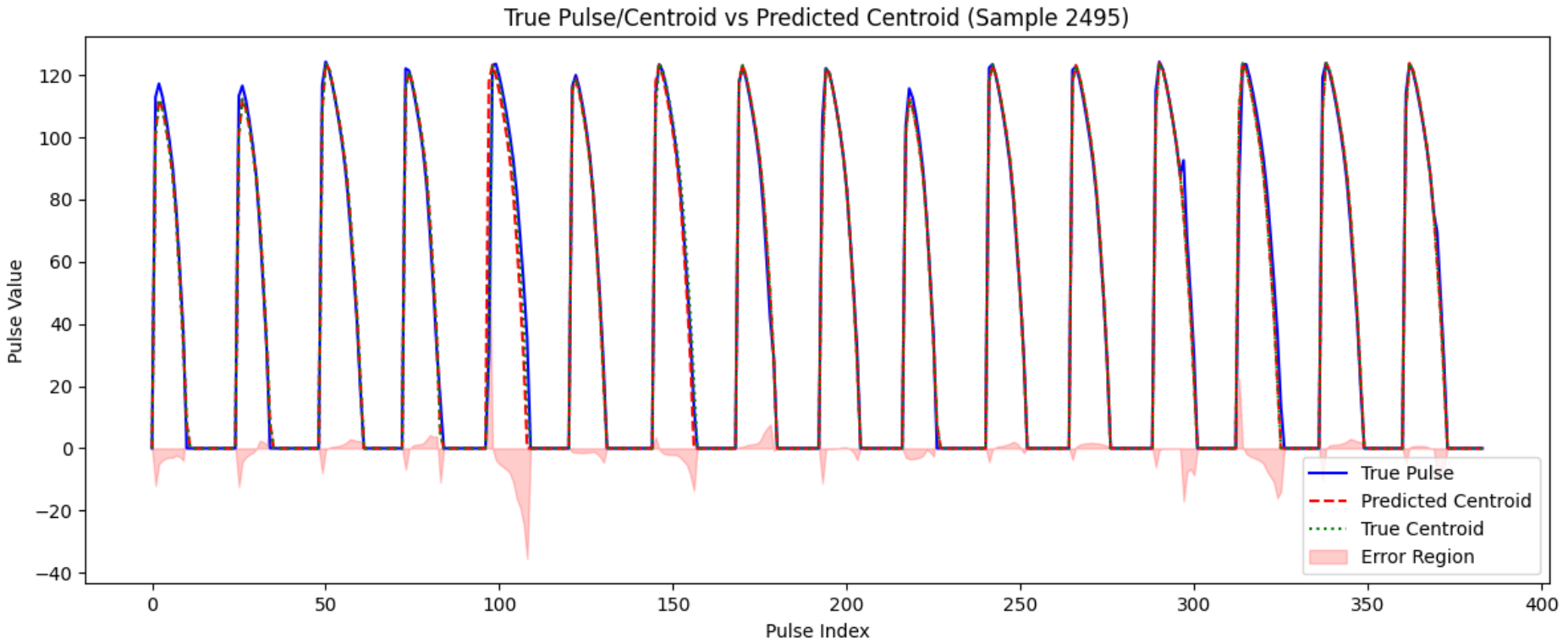
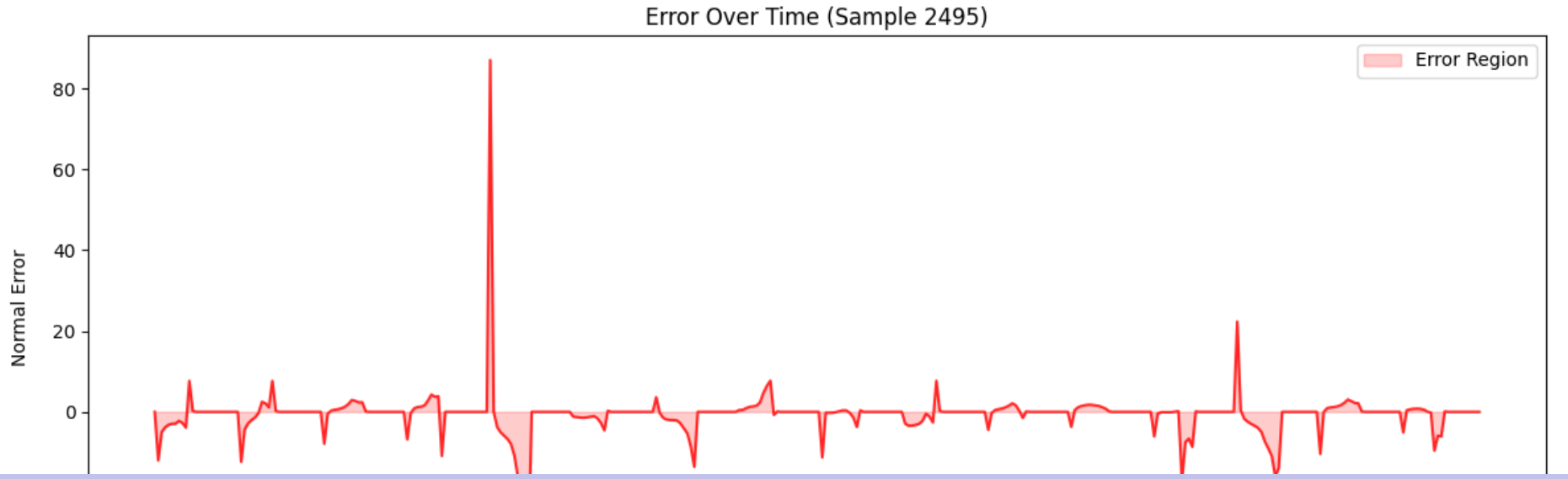


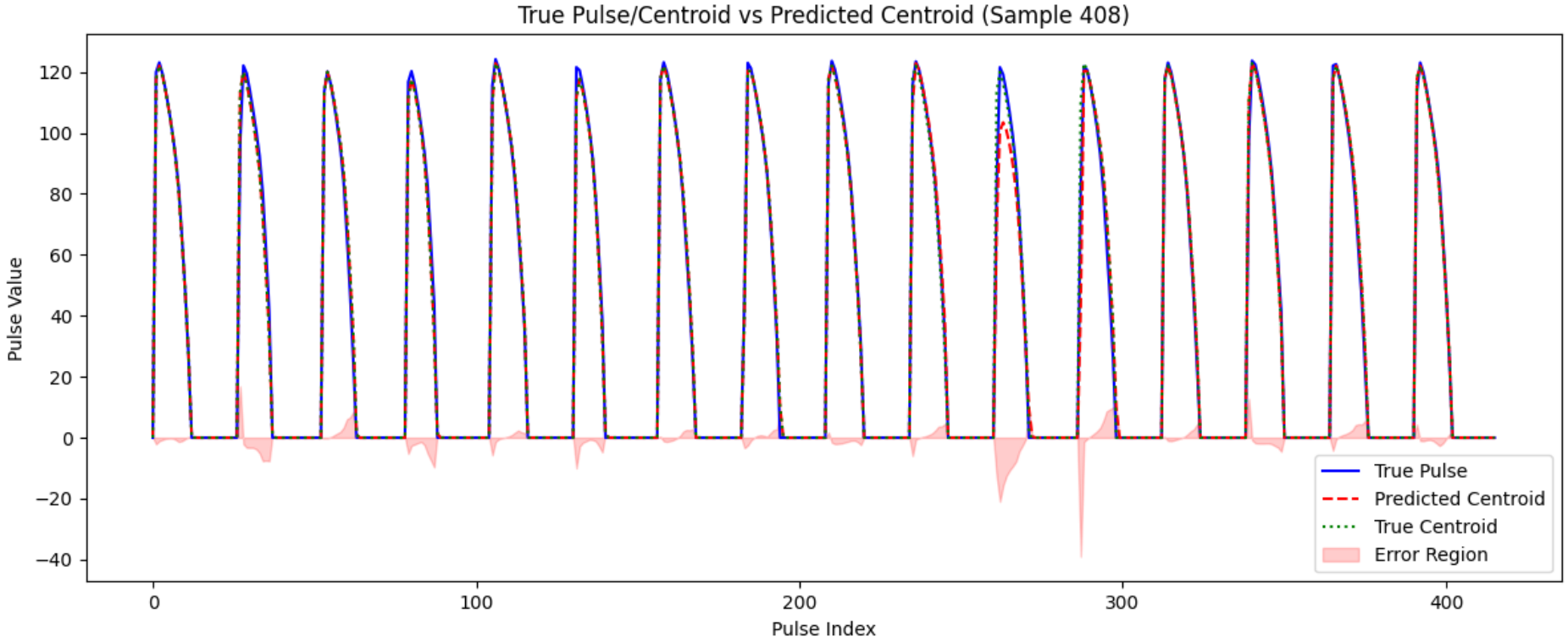
Figure: the raw data overlayed with the predicted pulses and the di length from data, with the error of the pulse, Reg hp=14, 89.80% and 1.237

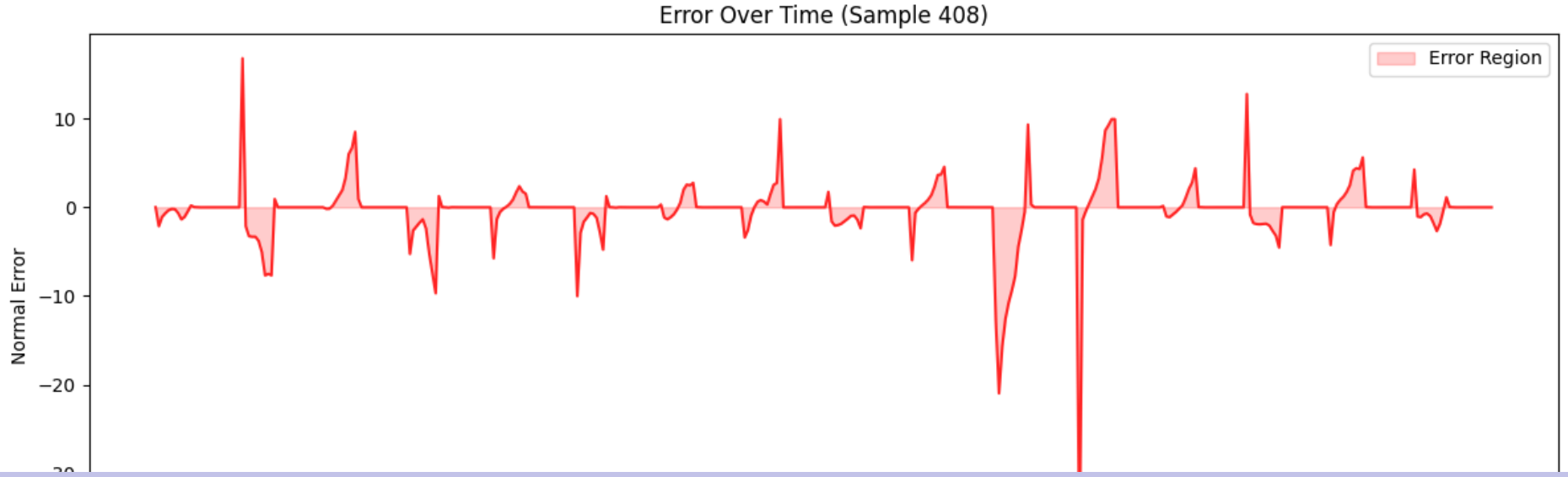
Minute 3


True Pulse/Centroid vs Predicted Centroid (Sample 2495)
Pulse Value
Pulse Index
True Pulse
Predicted Centroid
True Centroid
Error Region

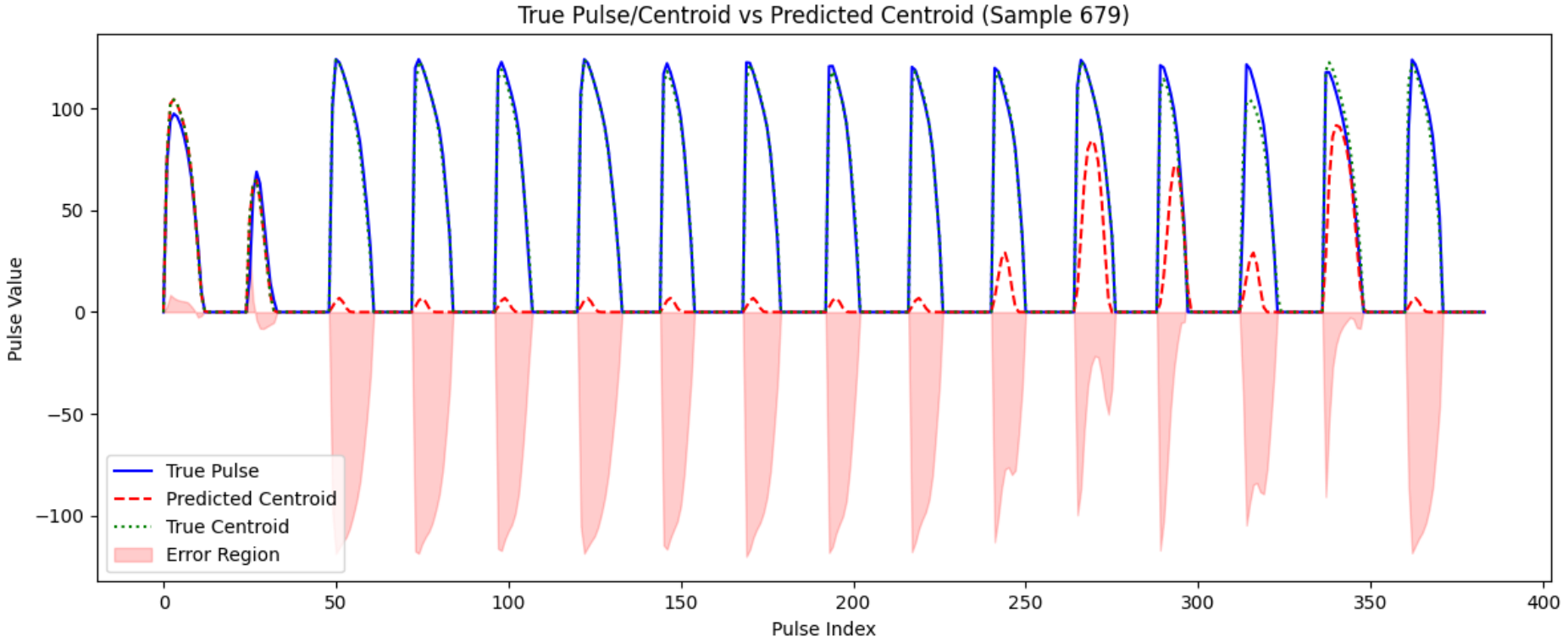

Error Over Time (Sample 2495)
Normal Error
Error Region

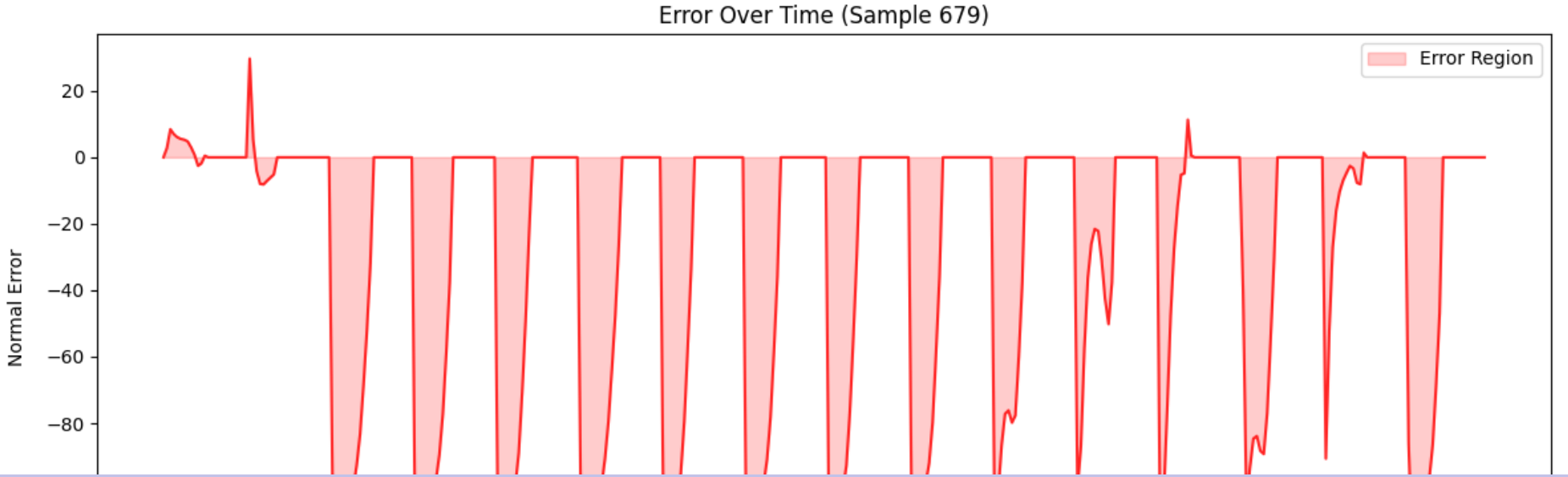
Minute 5


True Pulse/Centroid vs Predicted Centroid (Sample 408)
Pulse Value
Pulse Index
True Pulse
Predicted Centroid
True Centroid
Error Region

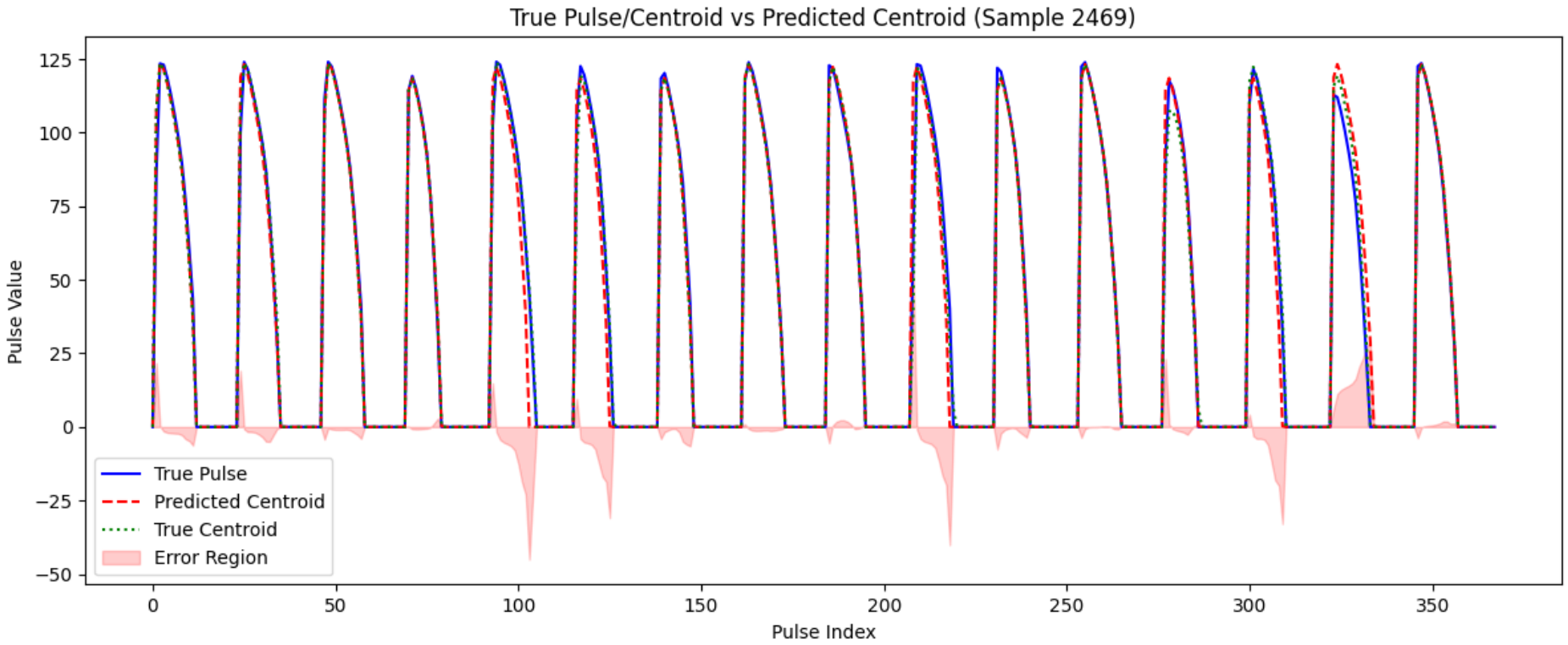

Error Over Time (Sample 408)
Normal Error
Error Region

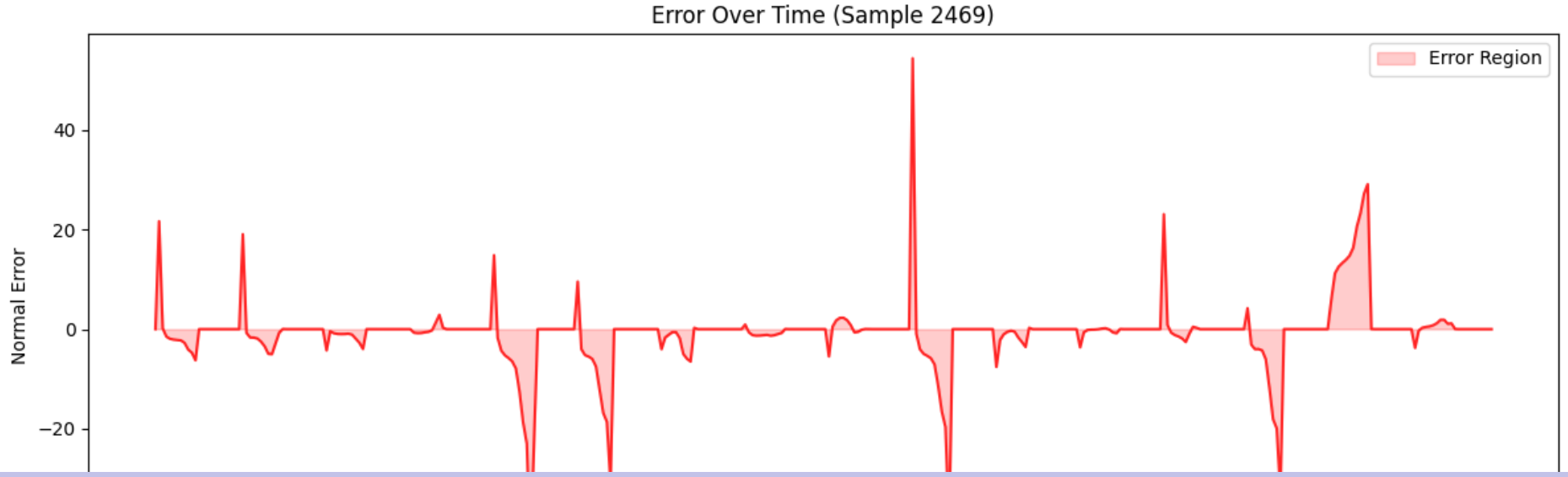
Minute 9


True Pulse/Centroid vs Predicted Centroid (Sample 679)
Pulse Value
Pulse Index
True Pulse
Predicted Centroid
True Centroid
Error Region
100
50
0
−50
−100
0
50
100
150
200
250
300
350
400


Error Over Time (Sample 679)
Normal Error
Error Region
20
0
−20
−40
−60
−80

Minute 13


True Pulse/Centroid vs Predicted Centroid (Sample 2469)
Pulse Value
Pulse Index
True Pulse
Predicted Centroid
True Centroid
Error Region
125
100
75
50
25
0
−25
−50
0
50
100
150
200
250
300
350


Error Over Time (Sample 2469)
Normal Error
Error Region
40
20
0
−20

Alphabet

Number of PCA Components: 12, Number of Clusters: 10, Reconstruction Error: 3.534902218547978, Counts: [4042 8492 4747 4560 50 4235 5568 5481 6344 4555] Mode 0
Number of PCA Components: 12, Number of Clusters: 10, Reconstruction Error: 1.480825570350 1159, Counts: [4025 8178 4780 50 4105 5522 4781 5861 6530 4242] Mode 1
Number of PCA Components: 12, Number of Clusters: 20, Reconstruction Error: 1.932618365092 3838, Counts: [1968 4023 1876 2986 50 2724 2108 2374 4105 3646 1085 2875 3938 2545 2459 1300 2406 1644 2025 1937] Mode 0
Number of PCA Components: 12, Number of Clusters: 20, Reconstruction Error: 0.961531285165 7231, Counts: [1238 4697 2103 50 2924 2597 2972 2541 4074 1730 2264 2032 1425 2118 3585 2271 2605 1124 3297 2427] Mode 1