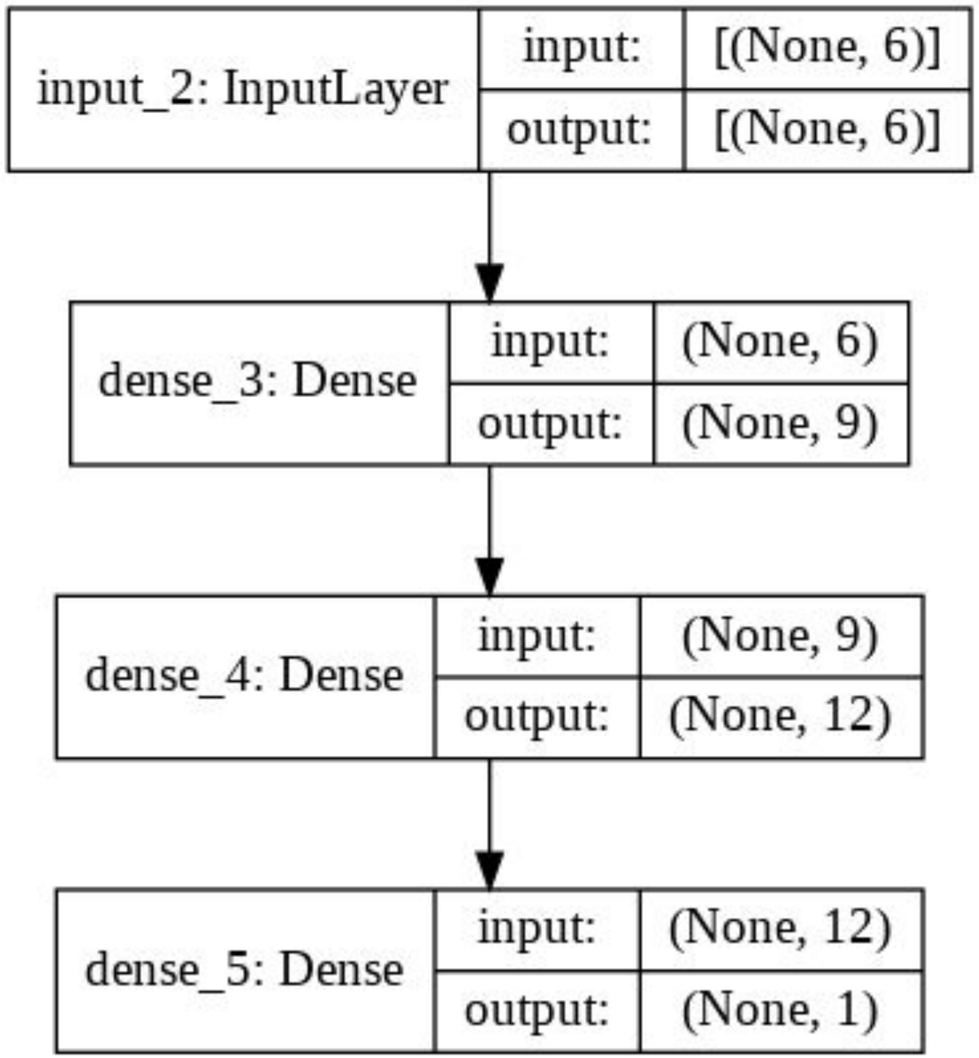
# *ANALYSIS OF DATA AND ENVIRONMENT*

- Number of train data points = `120435`
- Number of Test data points = 54235 (predicting High data points and low data points)
- input features: **x1(n-2), x1(n-1), x1(n),x2(n-2), x2(n-1), x2(n)**
- Output feature = **Force (F1)**
- Optimizer = Adam
- Loss function = Mean Squared Error
- Version of TensorFlow used = 2.6.0

# DEEP LEARNING MODEL STRUCTURE



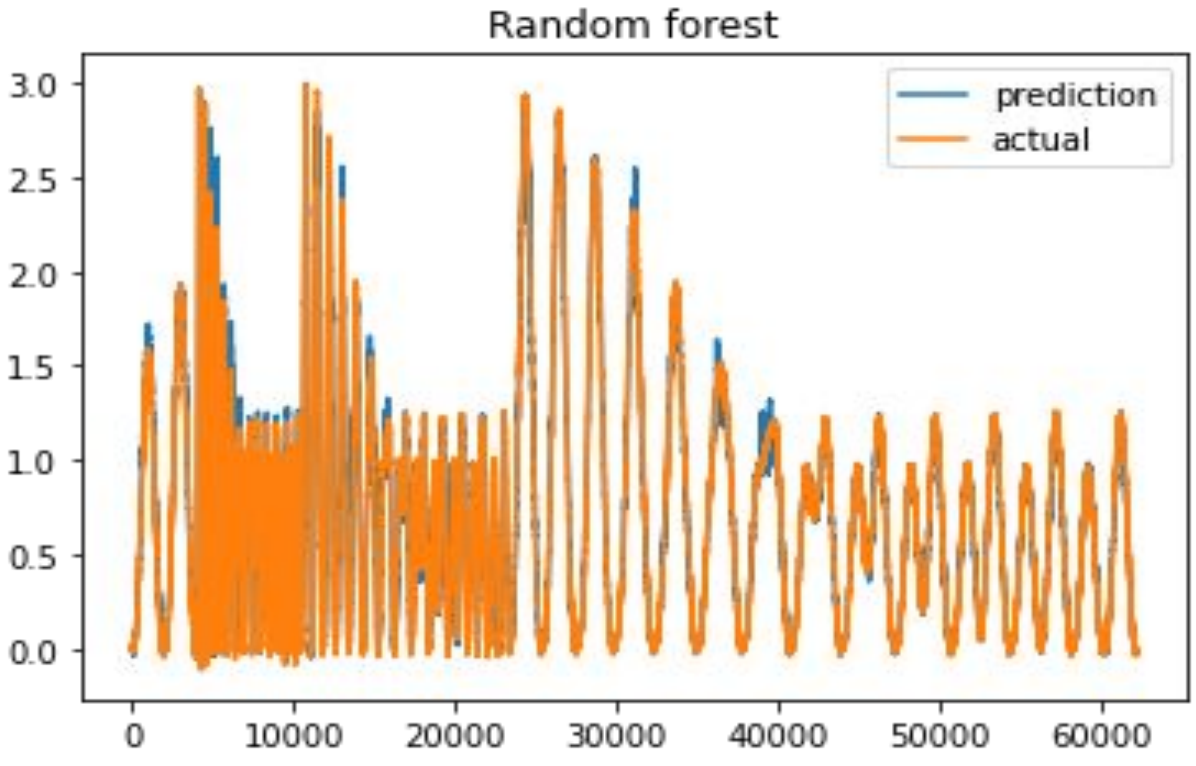
# Predicting Middle Amplitude points(39-78 Interaction points) including past appearances

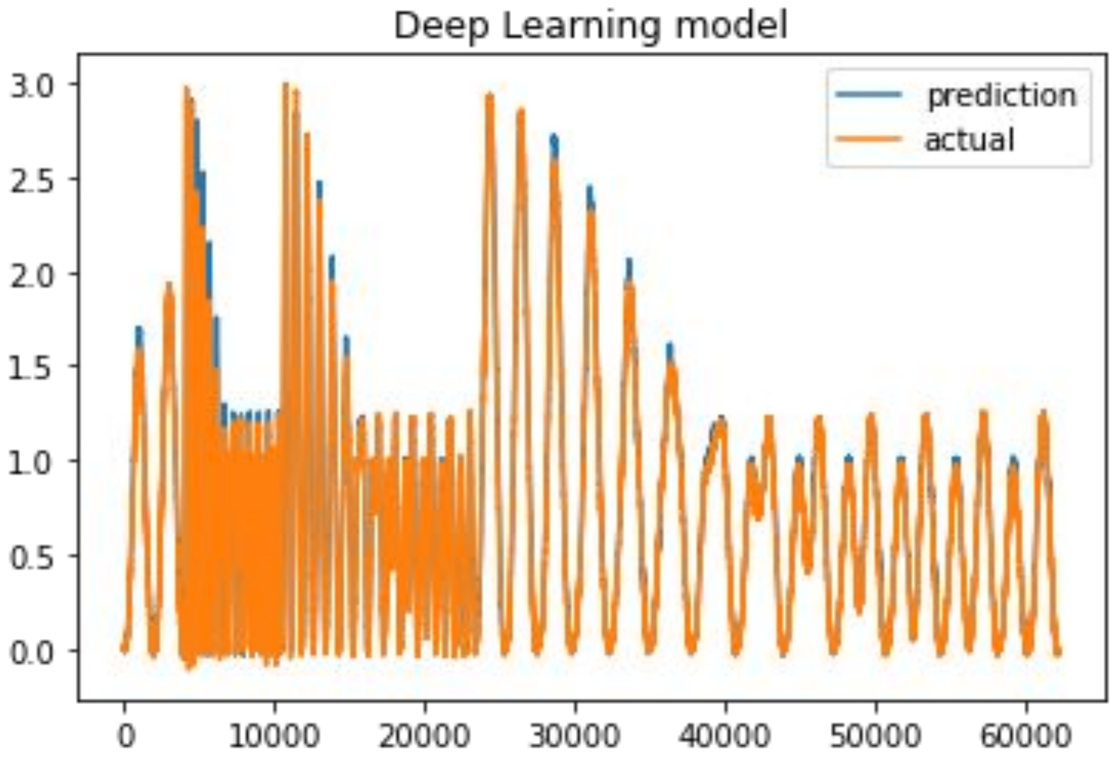
Random Forest

Root Mean squared error = 0.062257

*Deep Learning Model*

*Root Mean squared error = 0.055043*

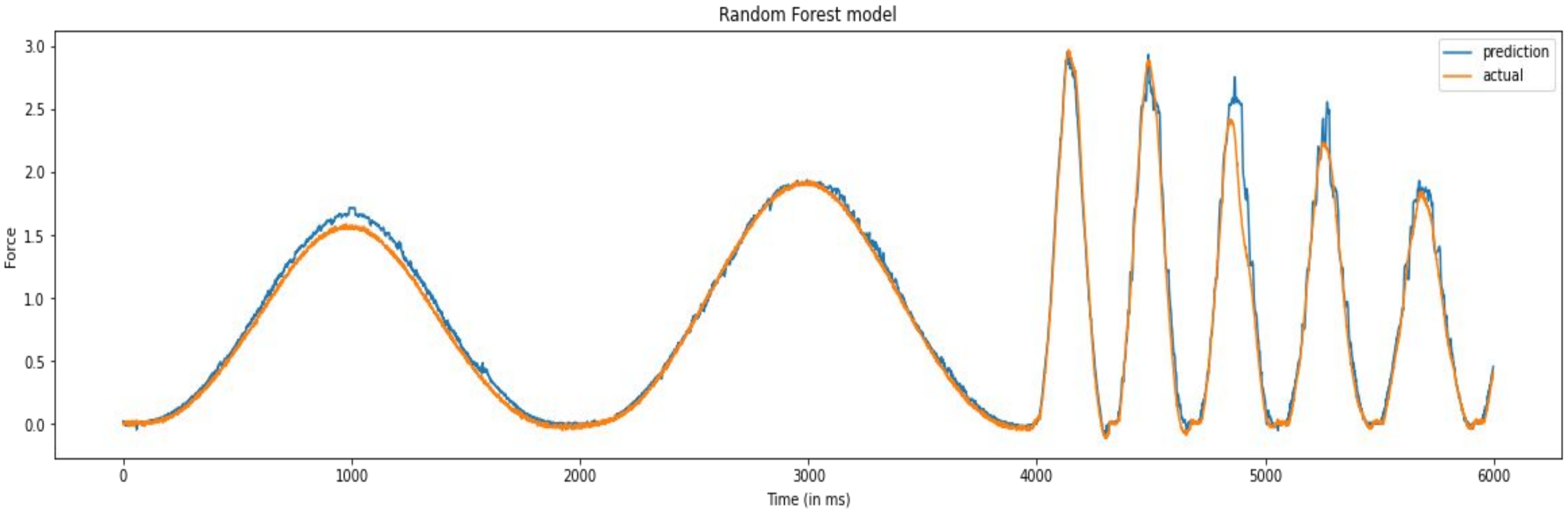


# Predicting Middle Amplitude points(39-78 Interaction points) for a lower time period including past appearances

Random Forest (CLOSER VIEW OF INITIAL POINTS)

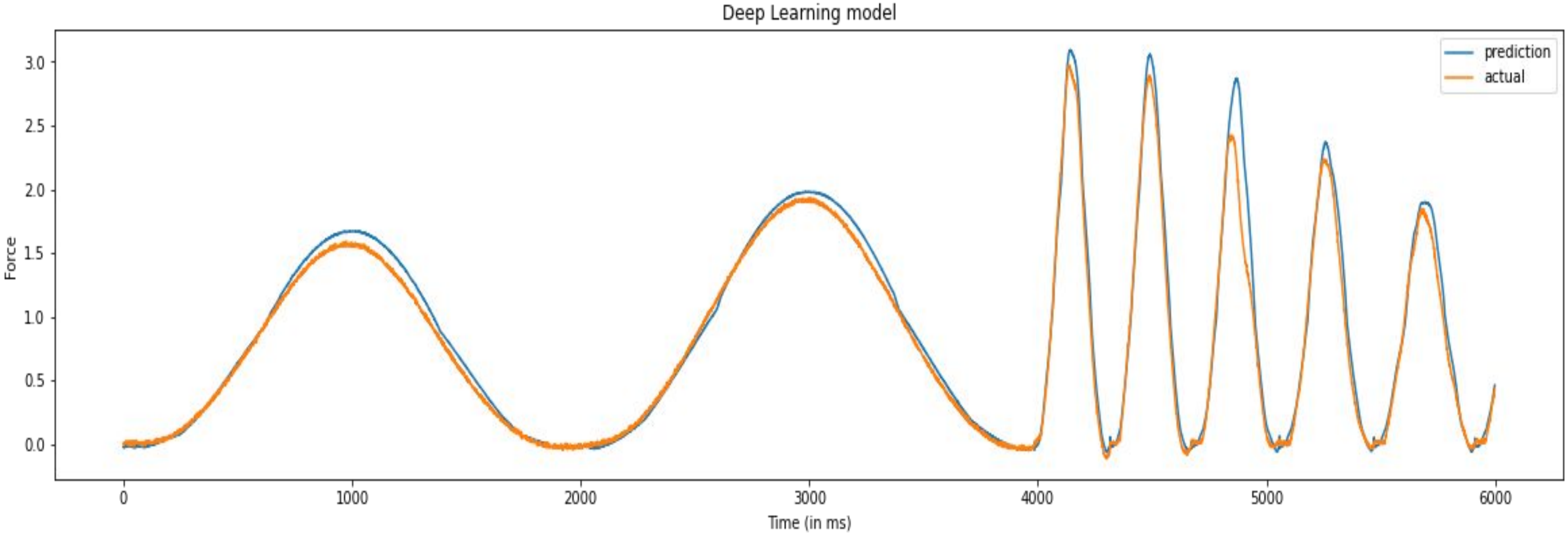
Root Mean squared error = 0.062257



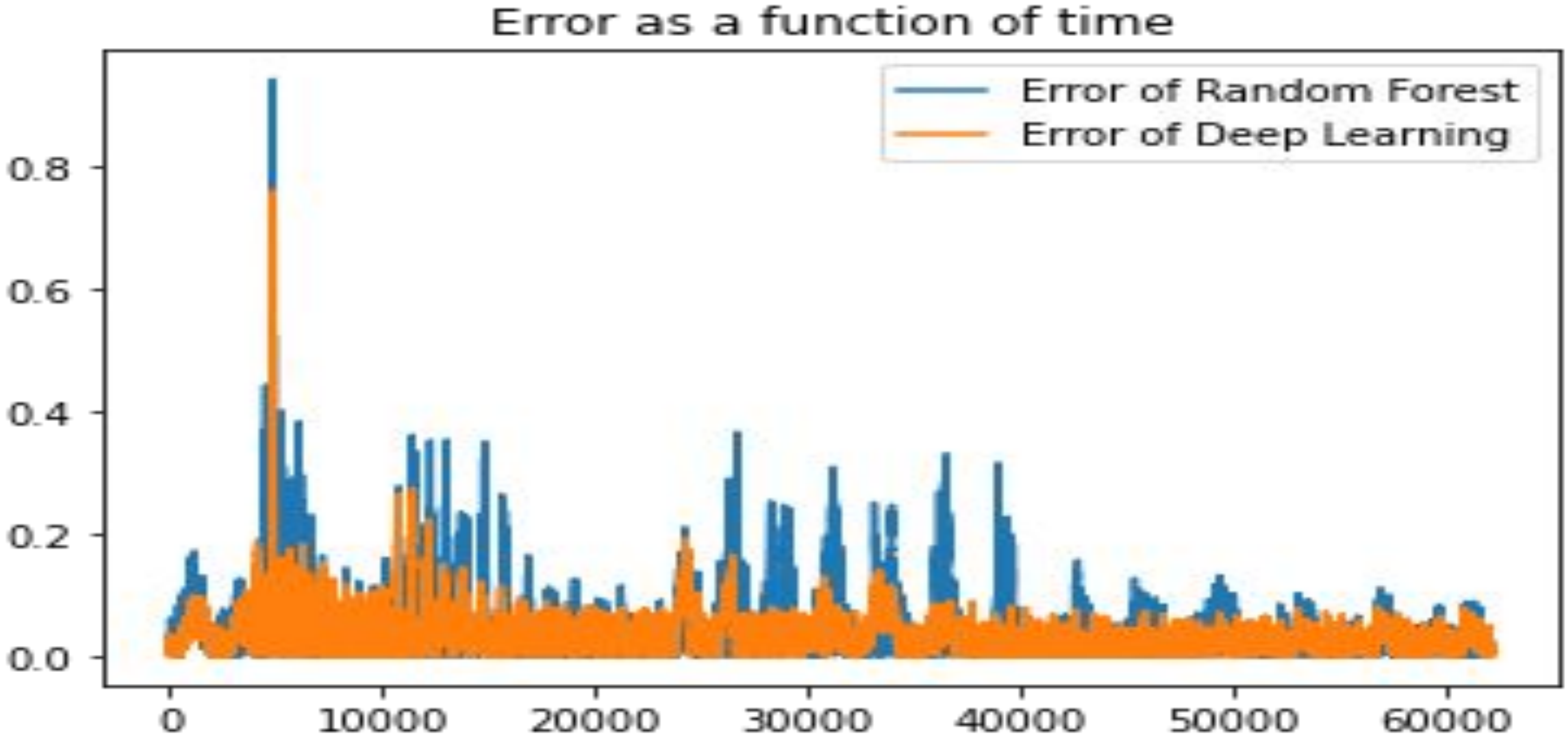
# Predicting Middle Amplitude points(39-78 Interaction points) for a lower time period including past appearances

*Deep Learning Model* (CLOSER VIEW OF INITIAL POINTS)

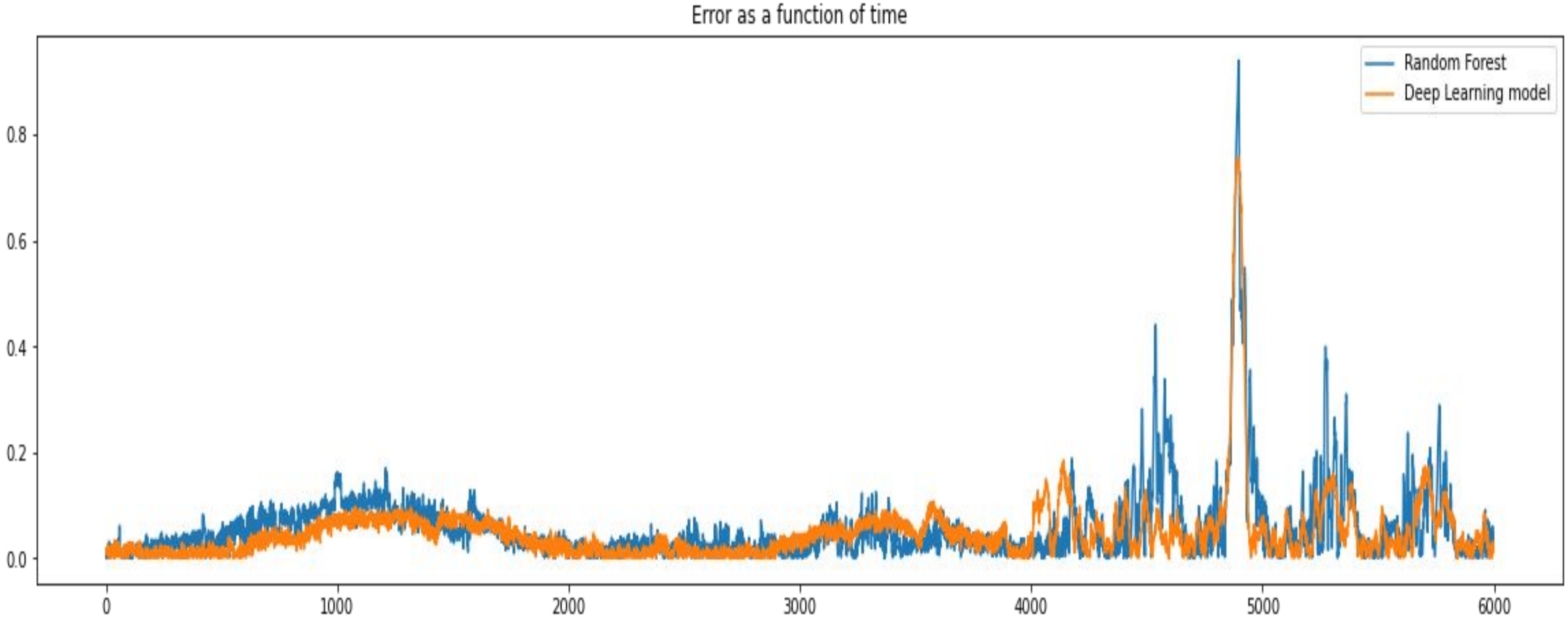
*Root Mean squared error = 0.055043*



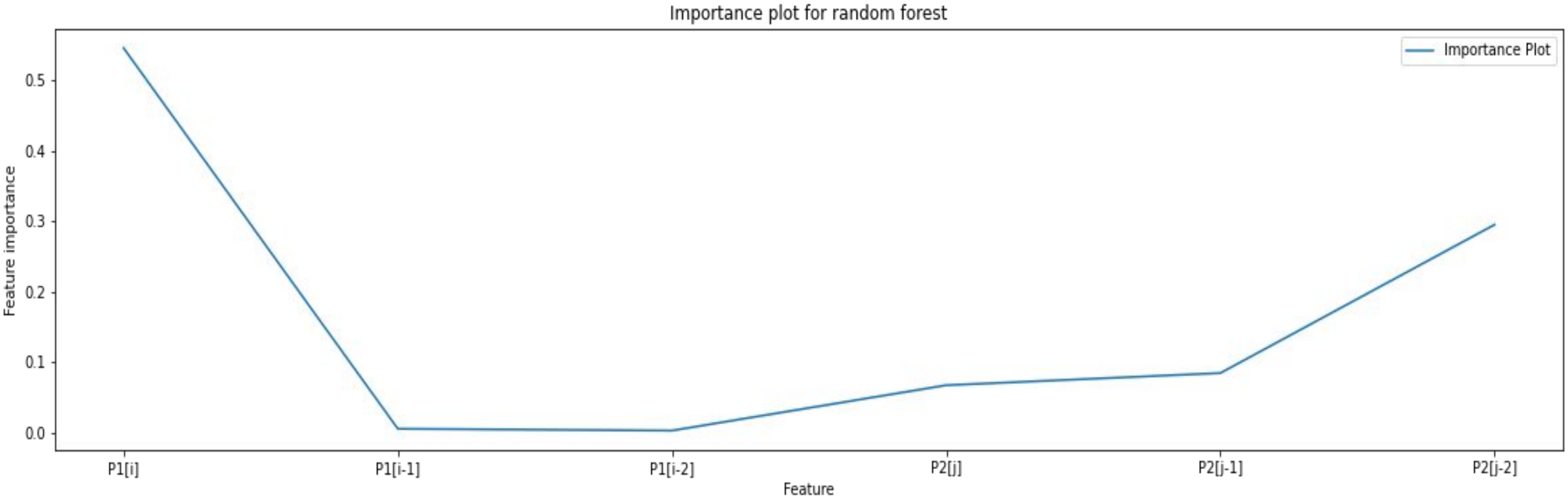
# *Error as a function of time using past appearances*



# ***Error as a function of time using past appearances*** *(closer view of initial points)*



# *Feature Importance plot for random forest using past appearances*



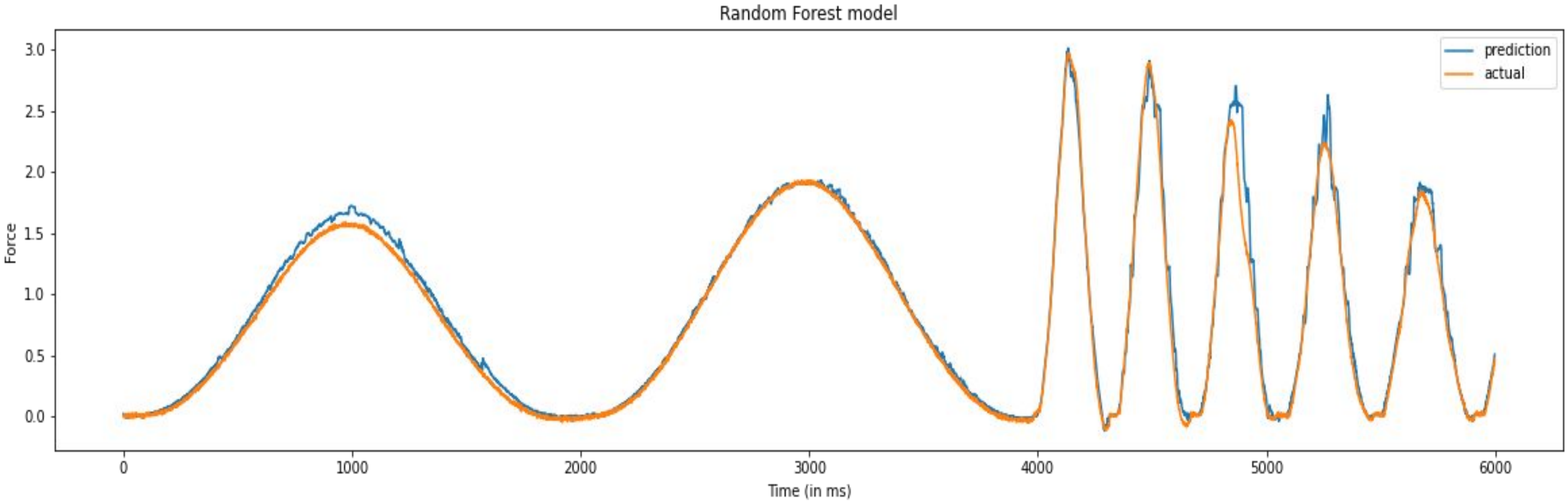
# *ANALYSIS OF DATA AND ENVIRONMENT*

- Number of train data points = 120435
- Number of Test data points = 54235 (predicting High data points and low data points)
- input features: **x1(n-2*$\tau$), x1(n-$\tau$), x1(n), x2(n-2*$\tau$), x2(n-$\tau$), x2(n)**
- Output feature = **Force (F1)**
- Optimizer = Adam
- Loss function = Mean Squared Error
- Version of TensorFlow used = 2.6.0

# Predictions

Random Forest (CLOSER VIEW OF INITIAL POINTS)

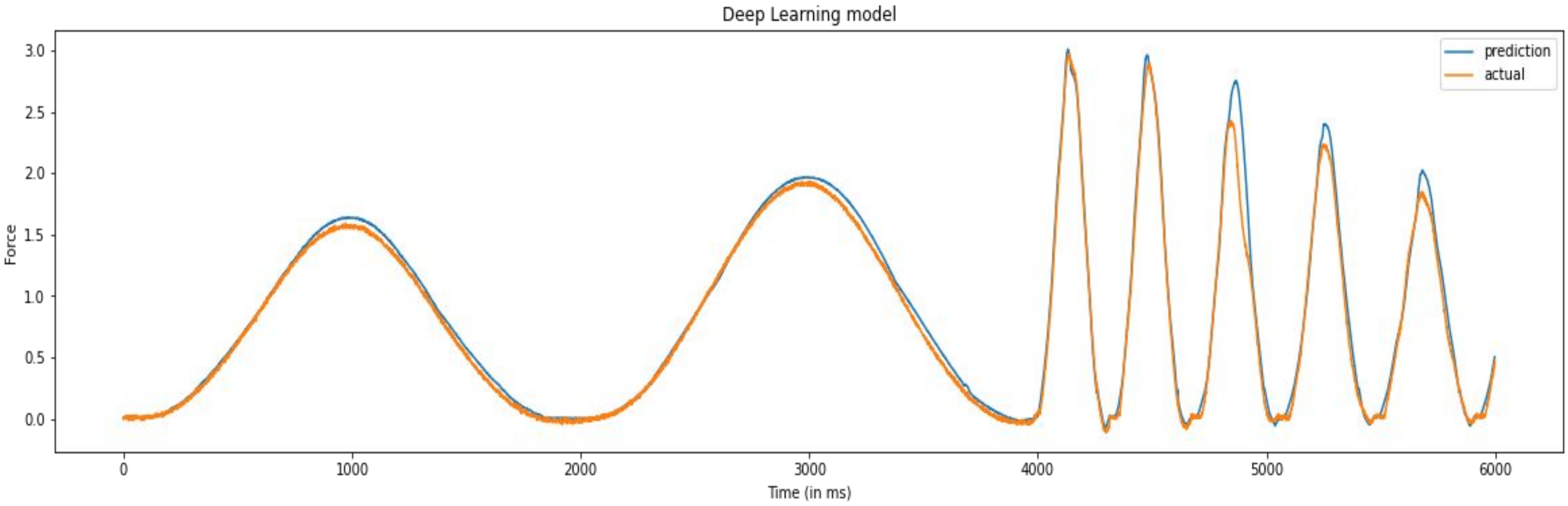
Root Mean squared error = 0.060872



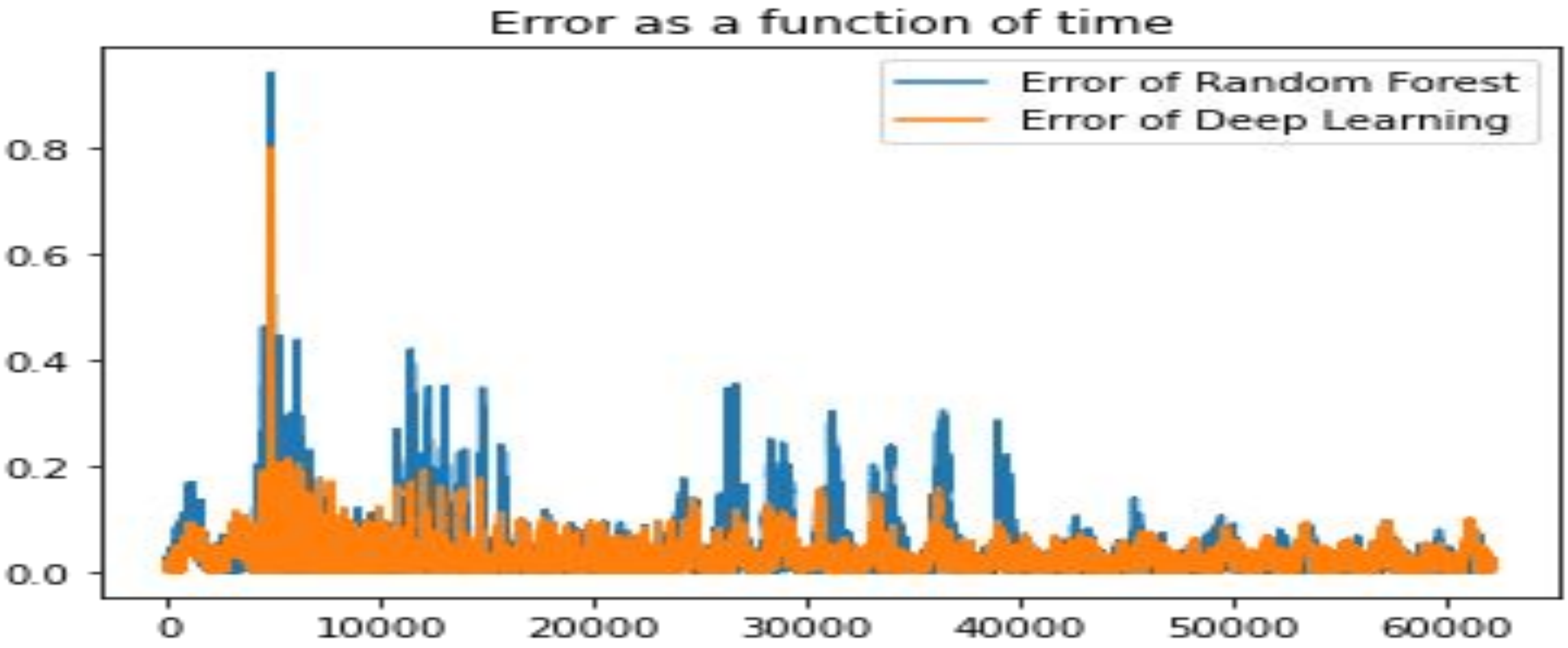
# Predictions

*Deep Learning Model* (CLOSER VIEW OF INITIAL POINTS)

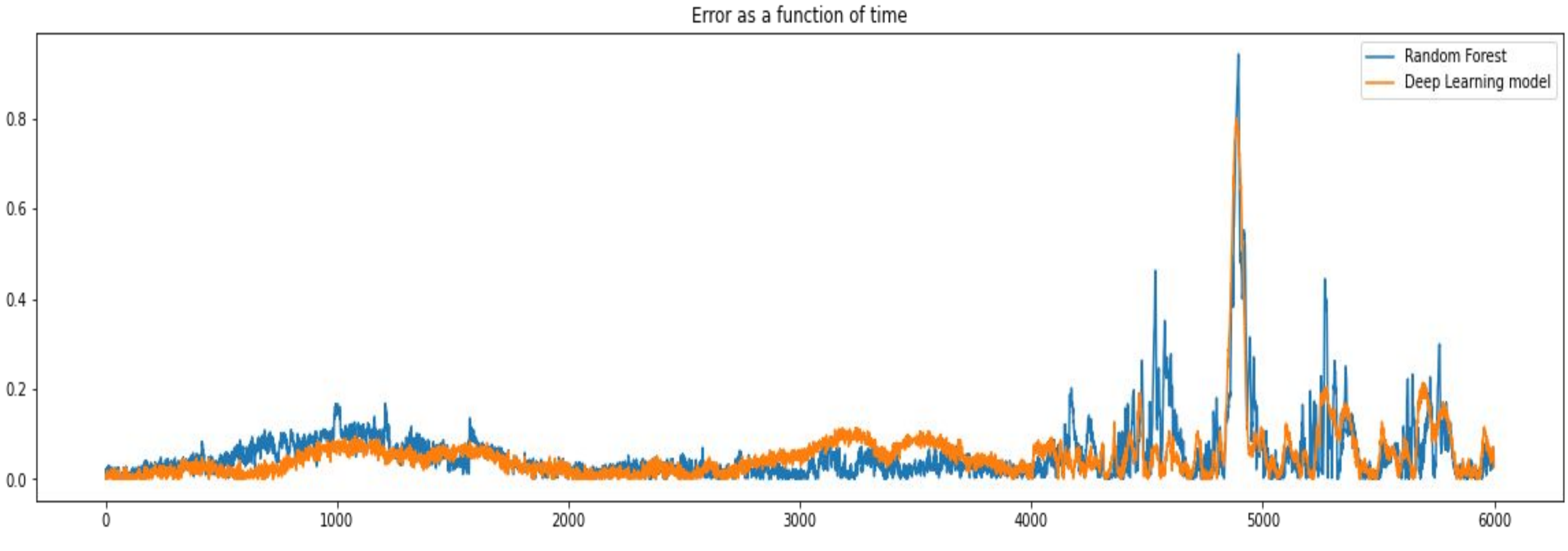
*Root Mean squared error = 0.047670*



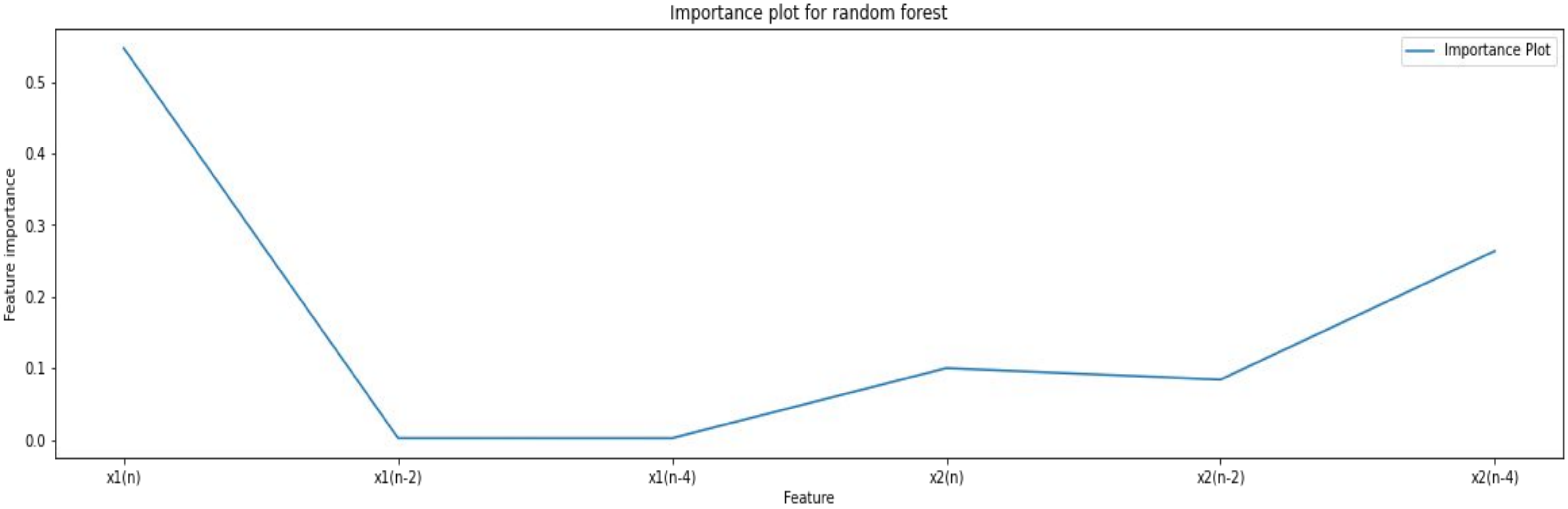
# *Error as a function of time using past appearances for large samples*



# ***Error as a function of time using past appearances*** *(closer view of initial points)*



# *Feature Importance plot for random forest using past appearances*



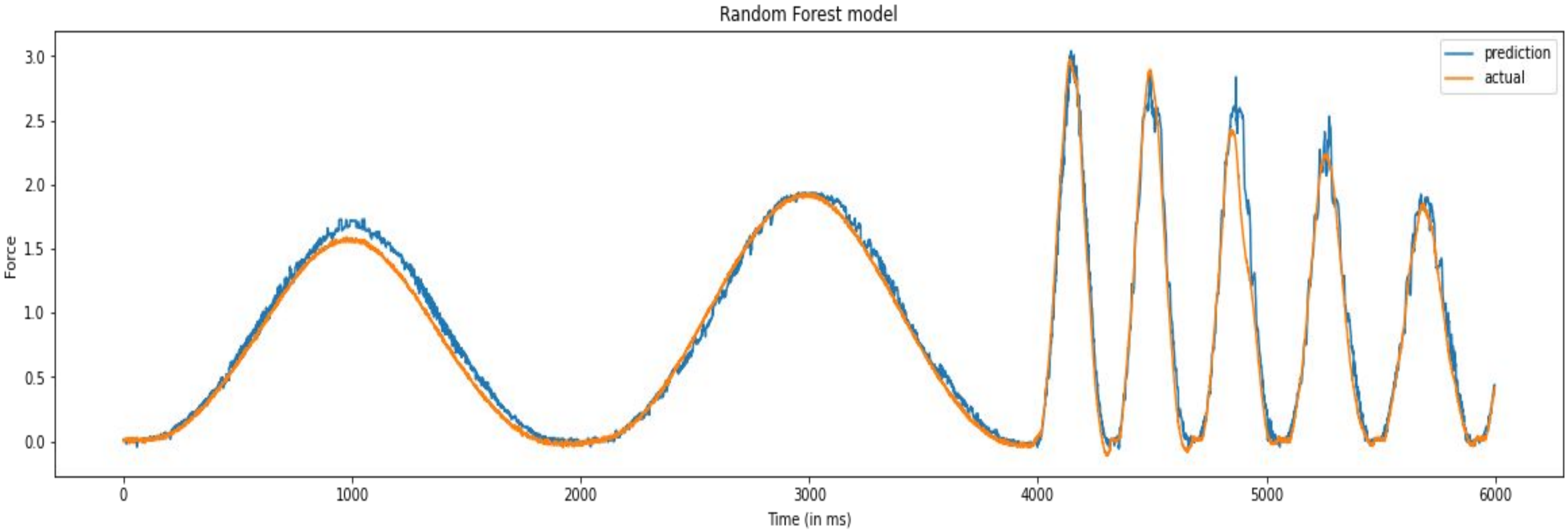
# *ANALYSIS OF DATA AND ENVIRONMENT*

- Number of train data points = `120435`
- Number of Test data points = 54235 (predicting High data points and low data points)
- input features:  **x1,x2**
- Output feature = **Force (F1)**
- Optimizer = Adam
- Loss function = Mean Squared Error
- Version of TensorFlow used  = 2.6.0

# Predicting Middle Amplitude points(39-78 Interaction points) for a lower time period including past appearances

Random Forest (CLOSER VIEW OF INITIAL POINTS)

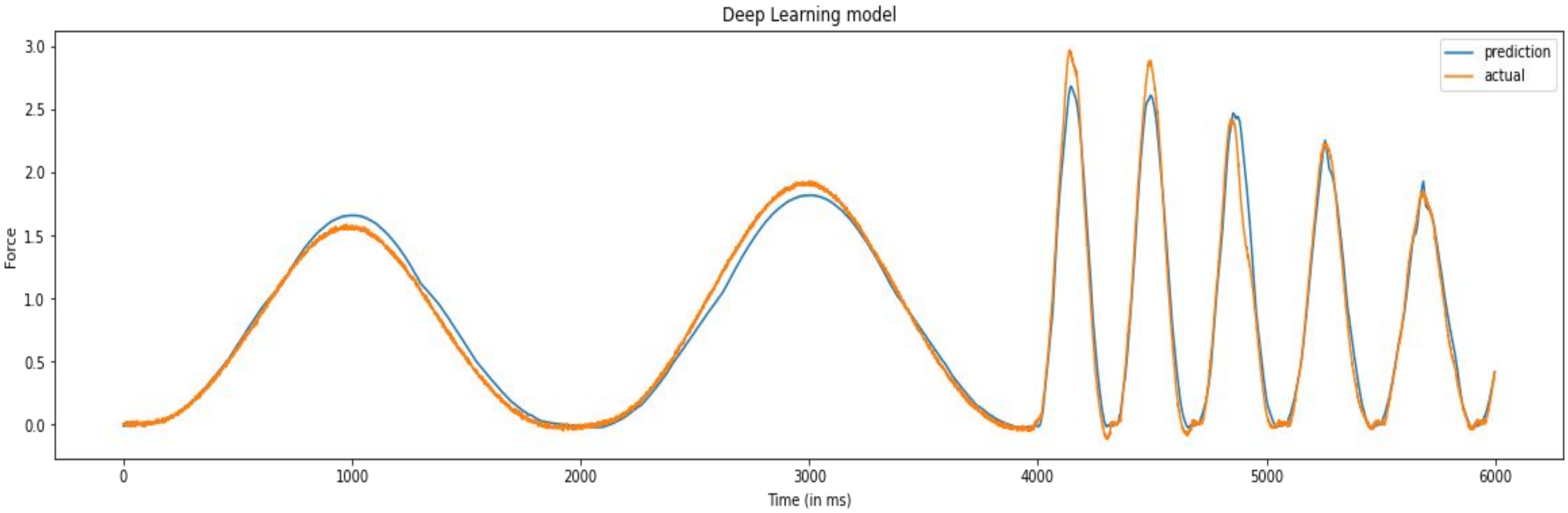
Root Mean squared error = 0.075171



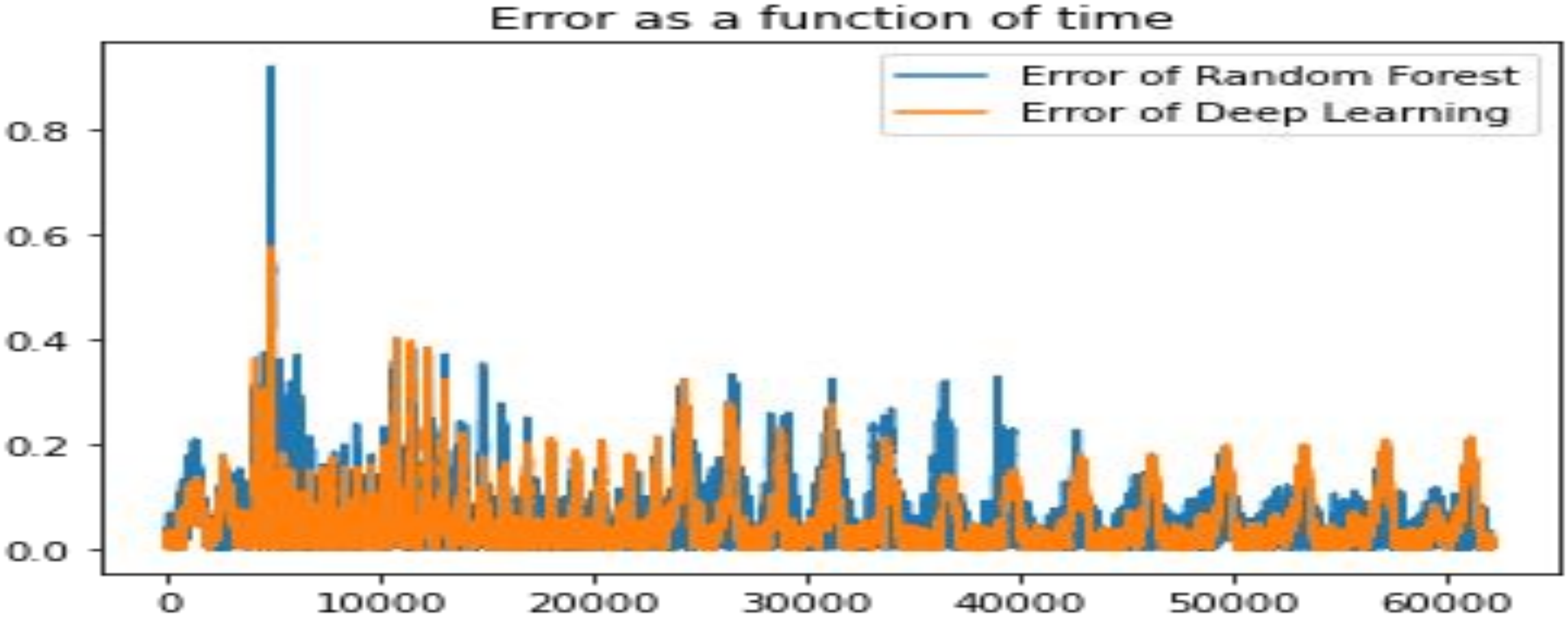
# Predicting Middle Amplitude points(39-78 Interaction points) for a lower time period including past appearances

*Deep Learning Model* (CLOSER VIEW OF INITIAL POINTS)

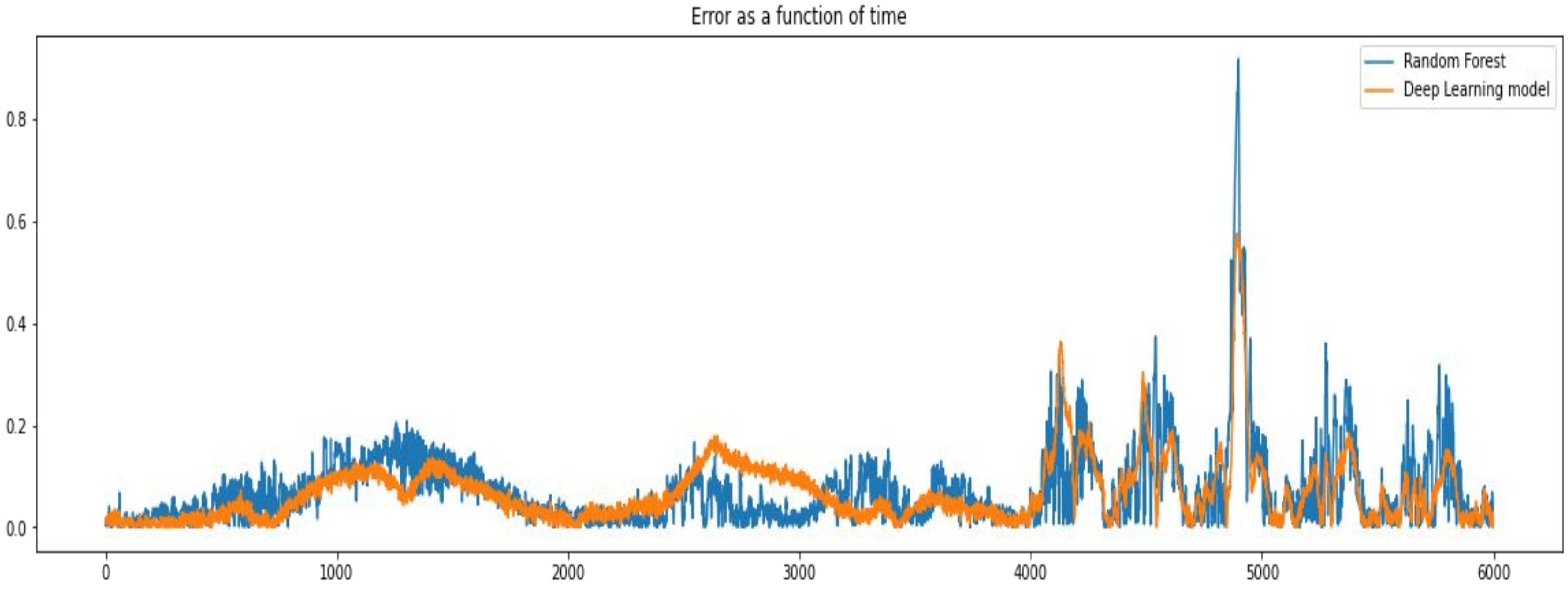
*Root Mean squared error = 0.081841*



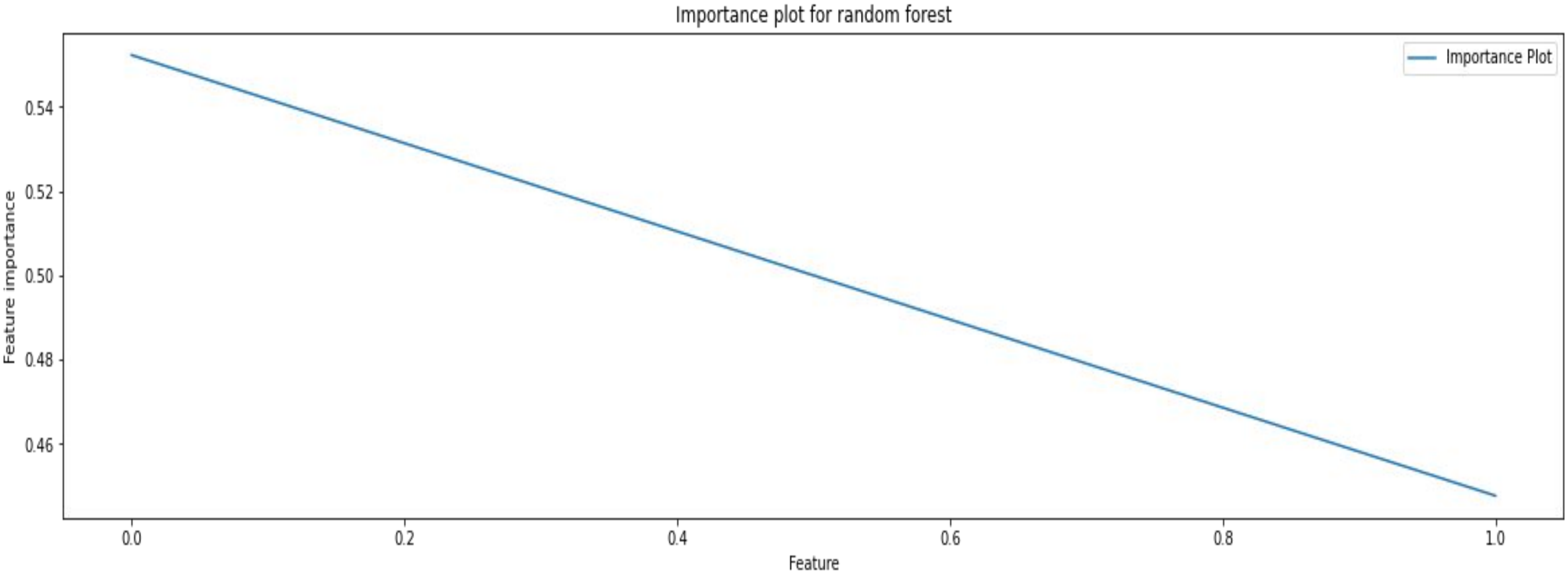
# *Error as a function of time using past appearances for large samples*



# *Error as a function of time using past appearances for small samples*



# Feature Importance plot for random forest using past appearances



# Random Forest Vs Deep Learning

**Models' performances on Middle amplitude points**

| Input feature vs output feature | p1,p2<br>f1 | | p1[i+1] - p1[i], p2[i+1] - p2[i]<br>f1 | | p1[i],p1[i-$\tau$],p1[i-2*$\tau$] p2[i],p2[i-$\tau$],p2[i-2*$\tau$]<br>f1 | |
|---|---|---|---|---|---|---|
| Model | DL model | RF | DL model | RF | DL model | RF |
| **"RMSE"** | ***0.081841*** | **0.075171** | ***0.055043*** | **0.060872** | ***0.047670*** | **0.060872** |
| Max error | ***0.575535*** | **0.918093** | ***0.798003*** | **0.923737** | ***0.800174*** | **0.94290** |
| Min error | ***4.17*10^-7*** | **5 * 10^-7** | ***1.192*10^-6*** | **2.42*10^-17** | ***5.96*10^-8*** | **1.98*10^-6** |
| Medin error | ***0.034598*** | **0.034598** | ***0.021189*** | **0.021189** | ***0.019451*** | **0.019451** |
| **Time taken by the model(in sec)** | ***0:00:27.6*** | **0:01:22.6** | ***0:01:15.9*** | **0:01:12.7** | ***0:01:15.0*** | **0:01:22.3** |

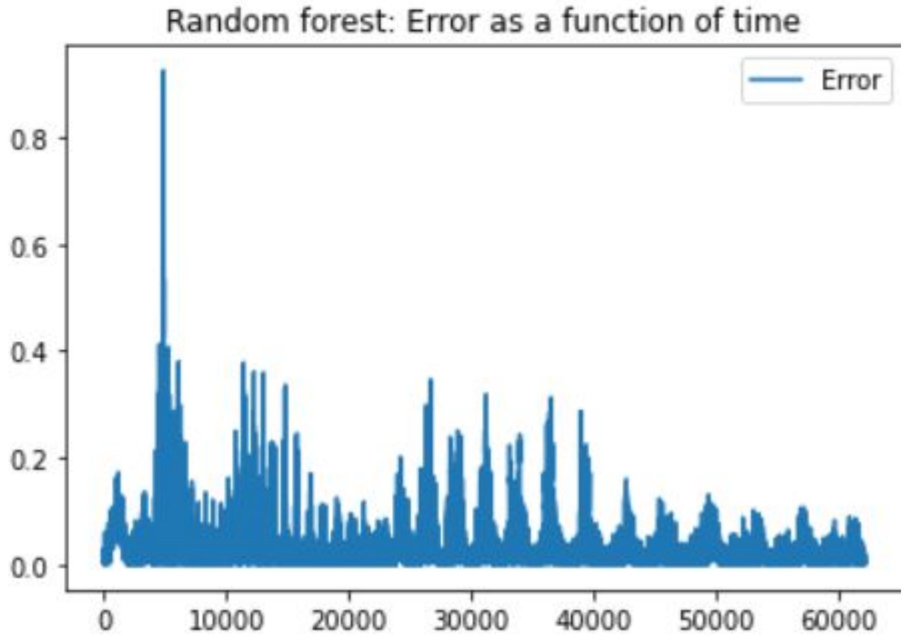
## Including Past Inputs

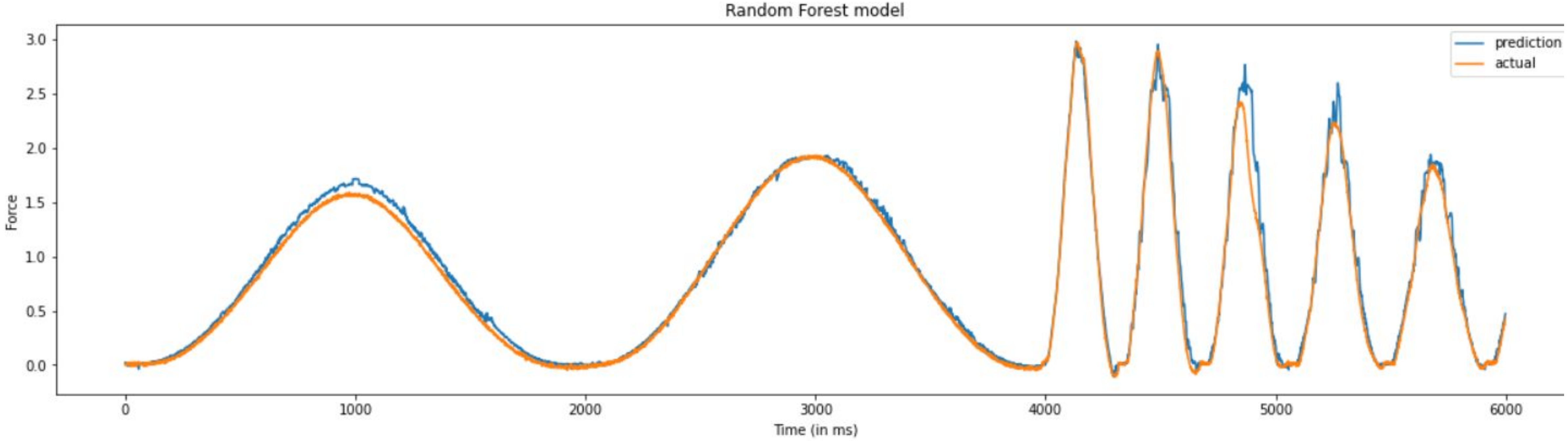
$x1(n), x1(n-t), x1(n-2t), x2(n), x2(n-t), x2(n-2t)$

$t = 1$

# Random Forest

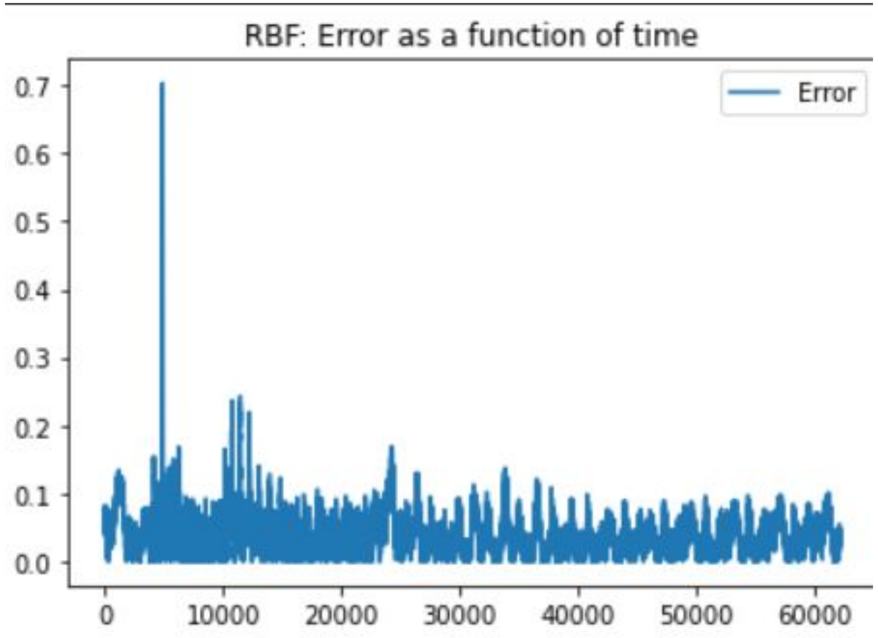
RMSE : `0.062569`

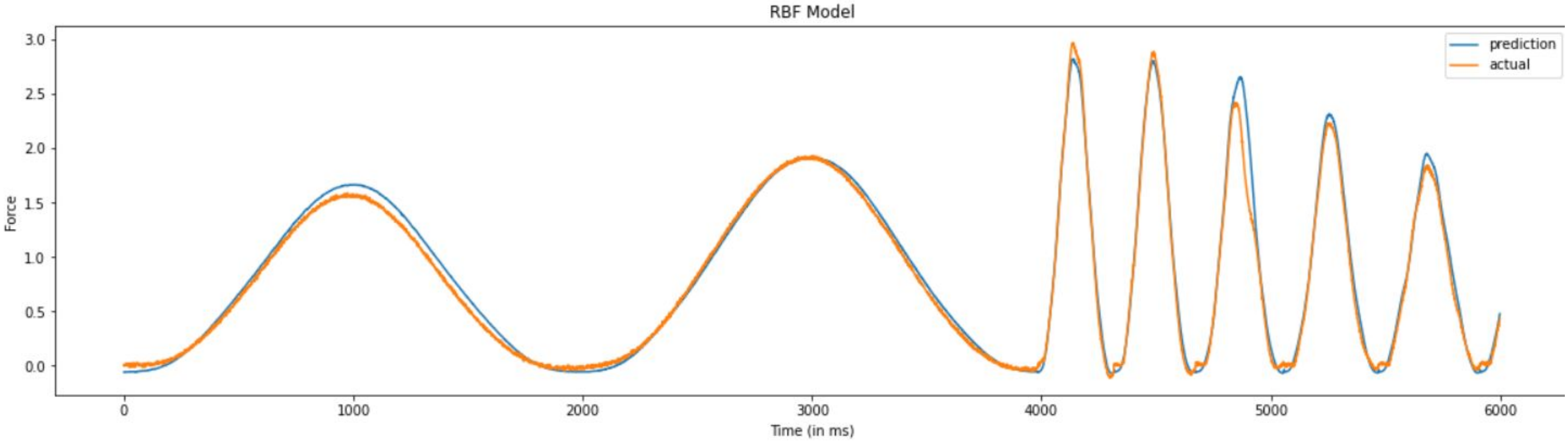


# RBF

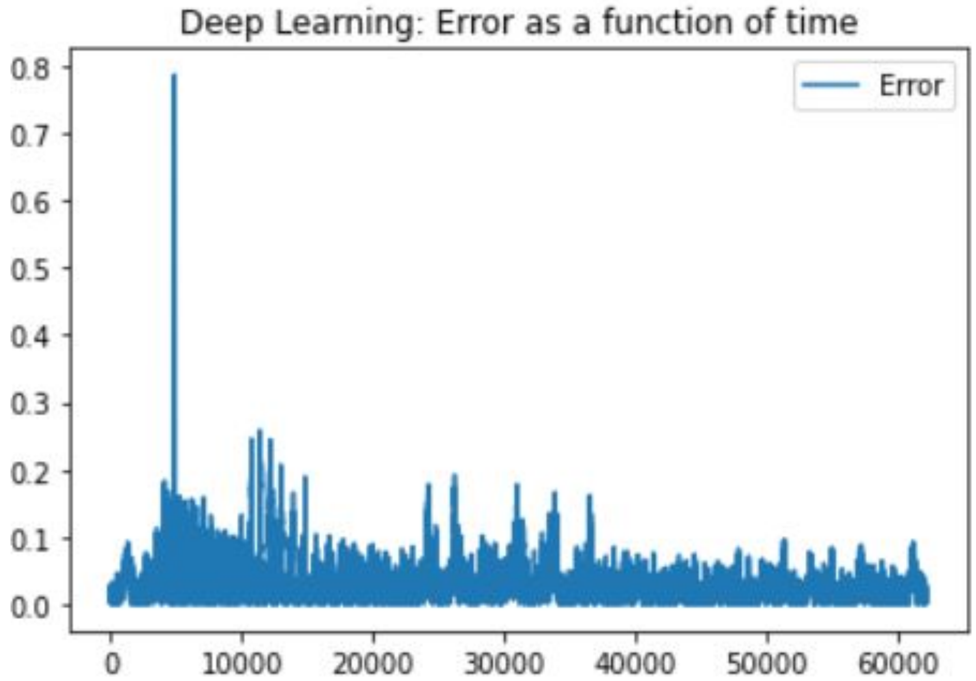
RMSE : 0.0519104

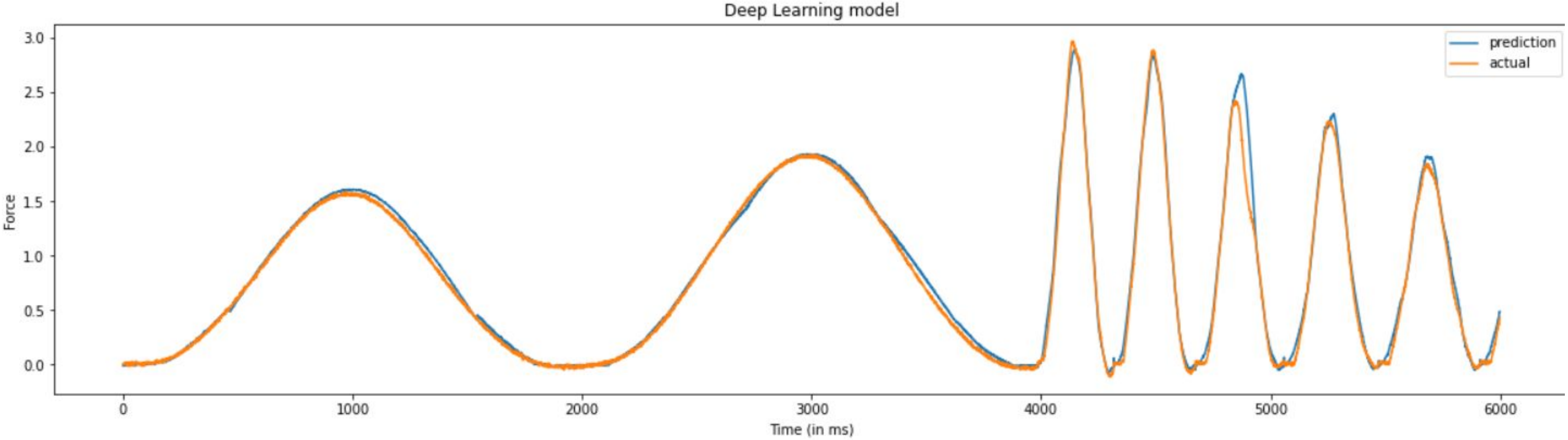


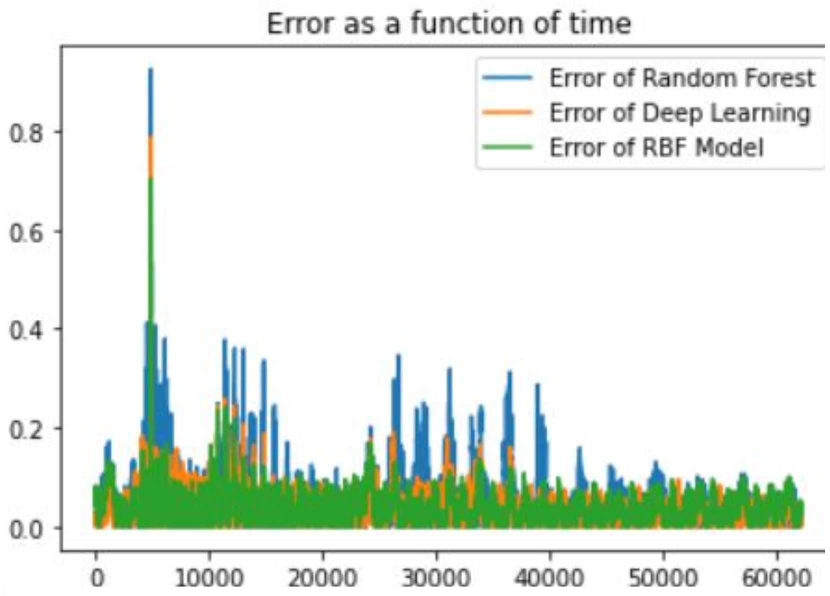
# Deep Learning

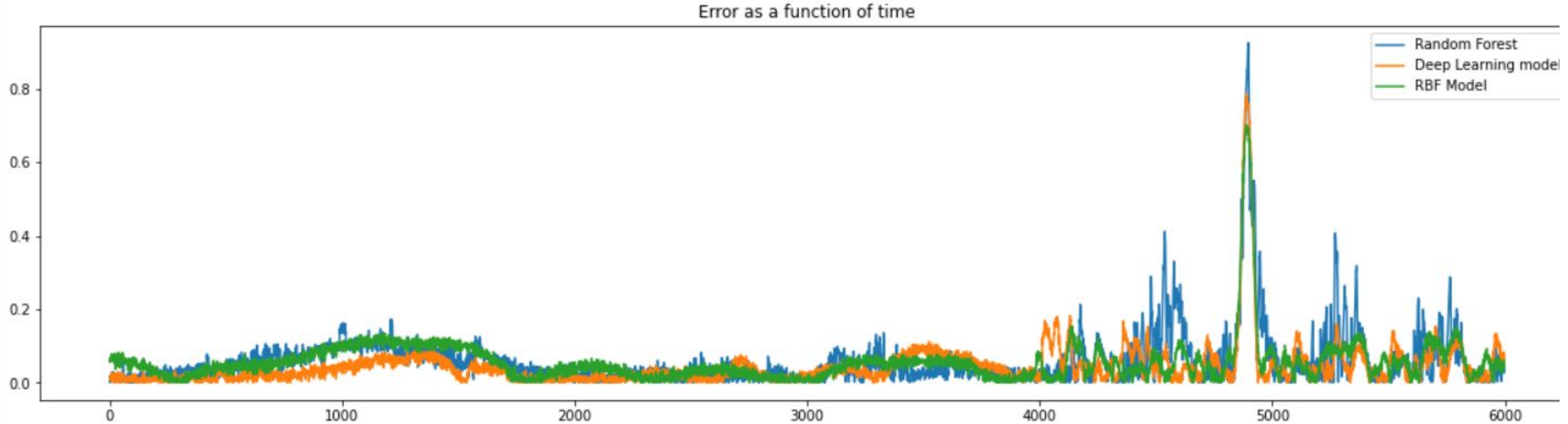
RMSE : `0.04770387`



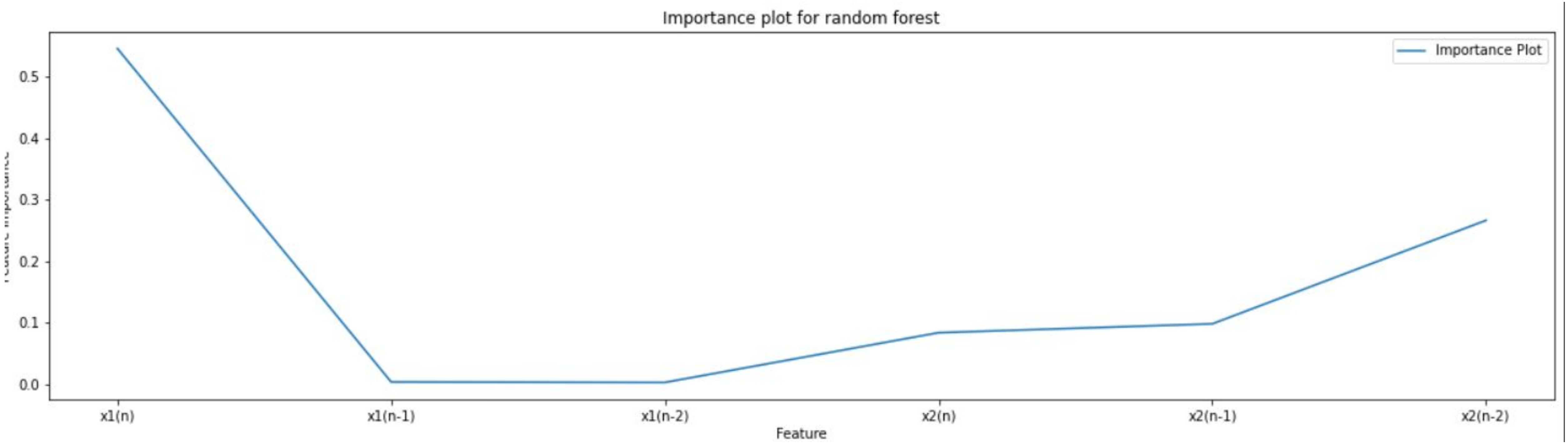
# Error comparison among the three models



# Feature Importance

x1(n), x1(n-1), x1(n-2), x2(n), x2(n-1), x2(n-2)



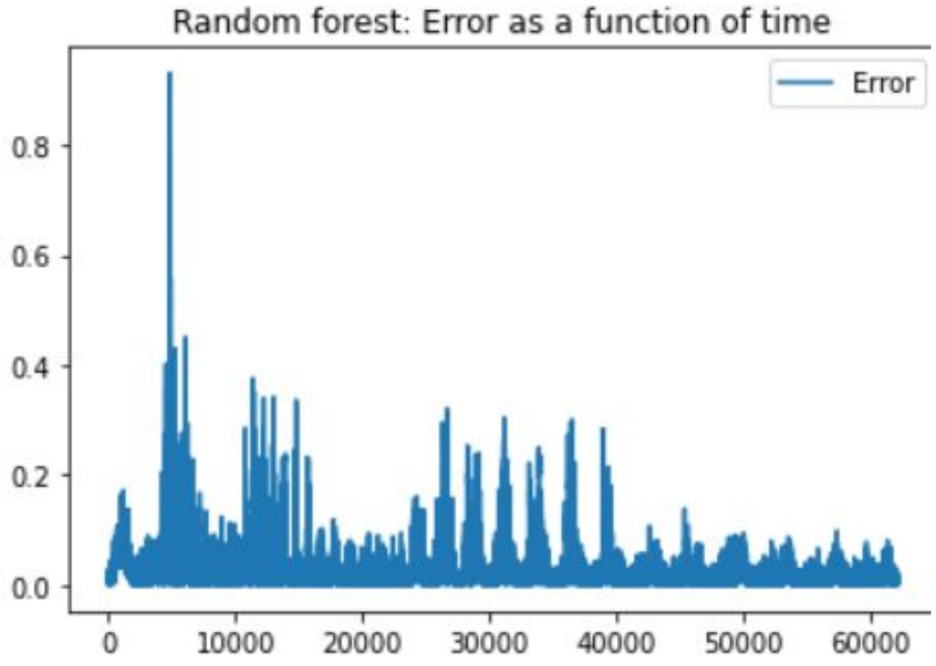
## Including Past Inputs

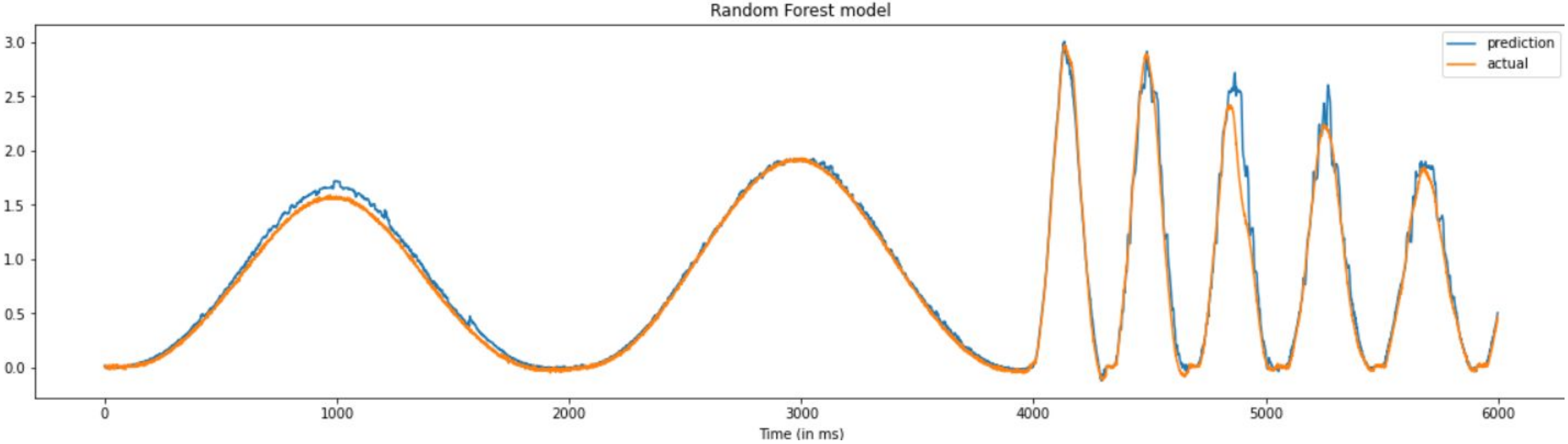
$x1(n), x1(n-t), x1(n-2t), x2(n), x2(n-t), x2(n-2t)$

$t = 2$

# Random Forest

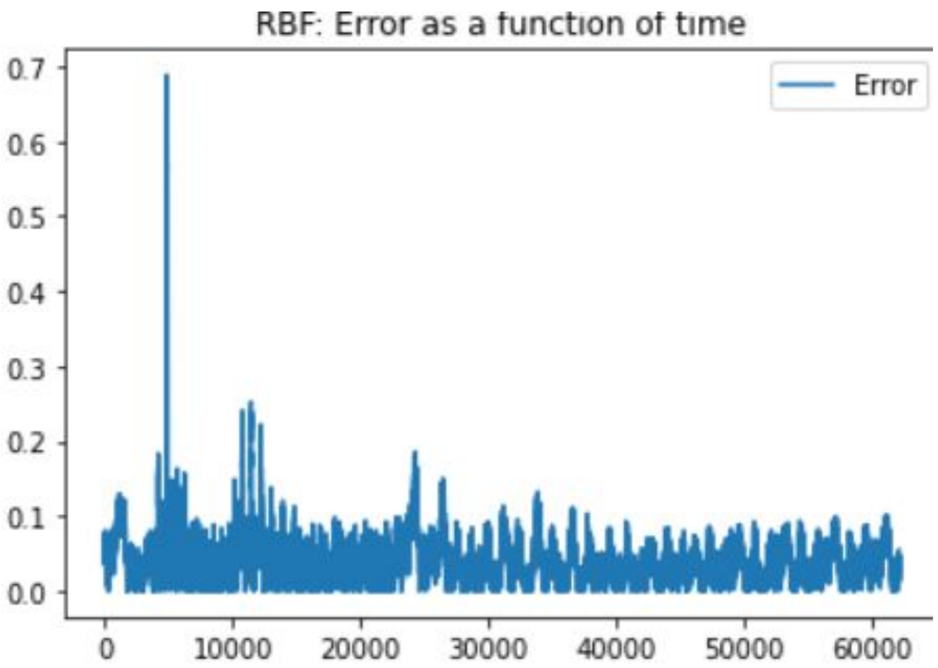
RMSE : `0.061248481`

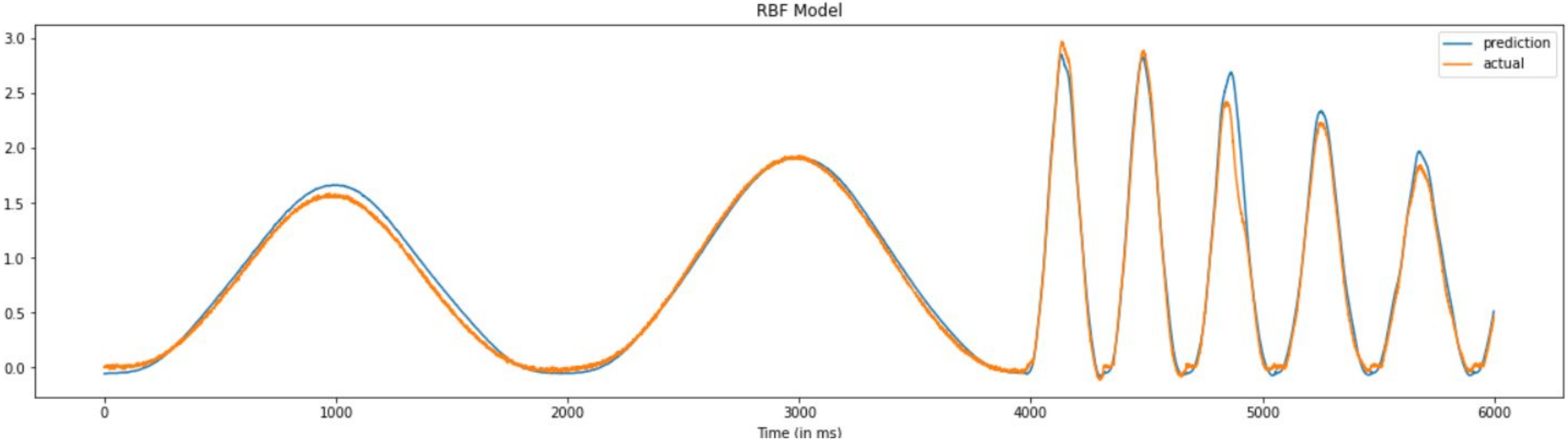


# RBF

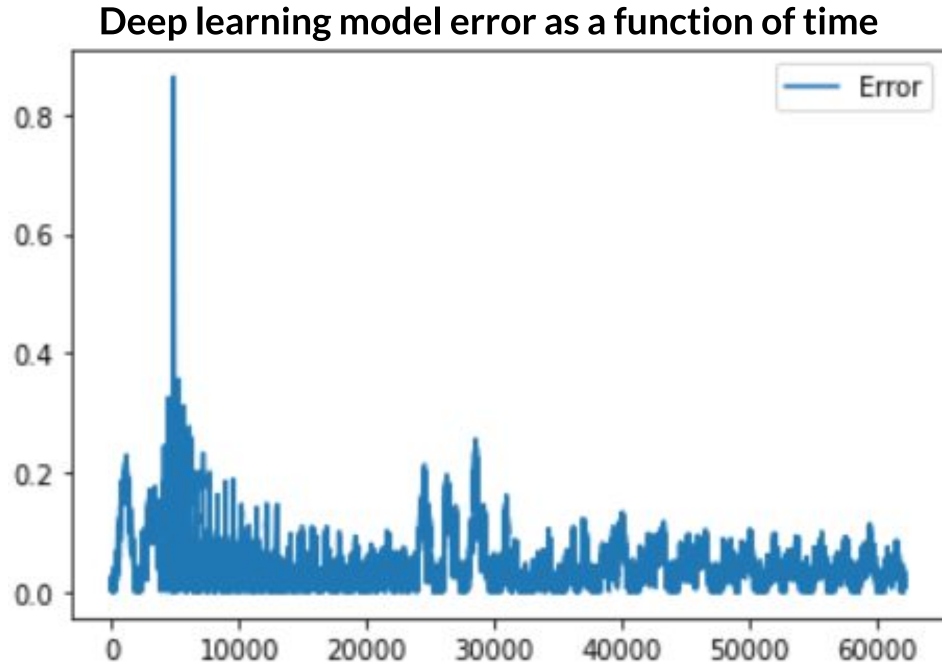
RMSE : 0.05222988

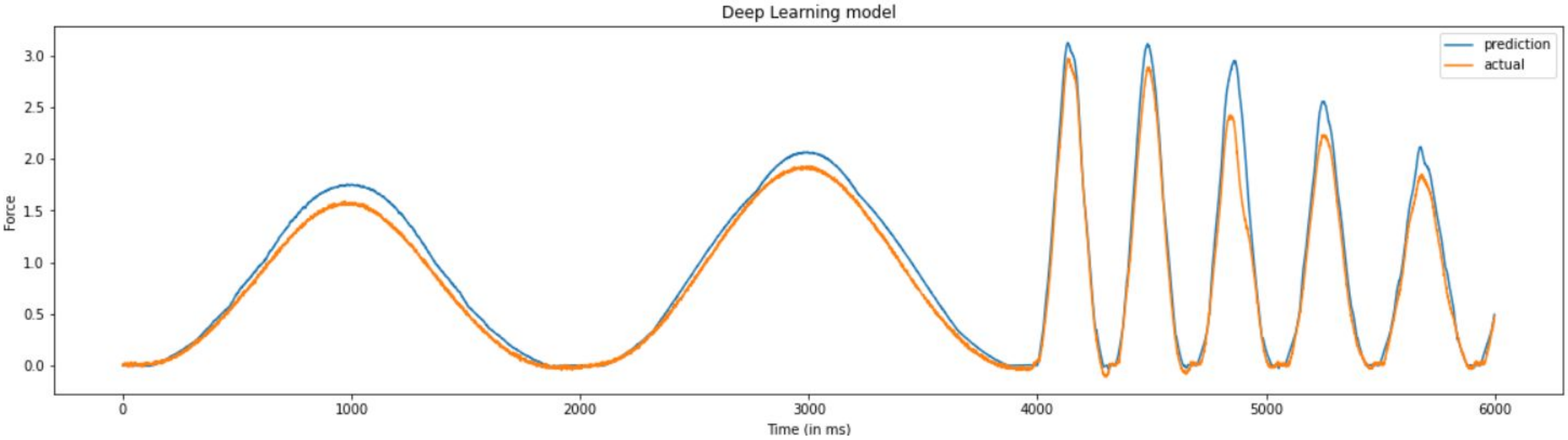


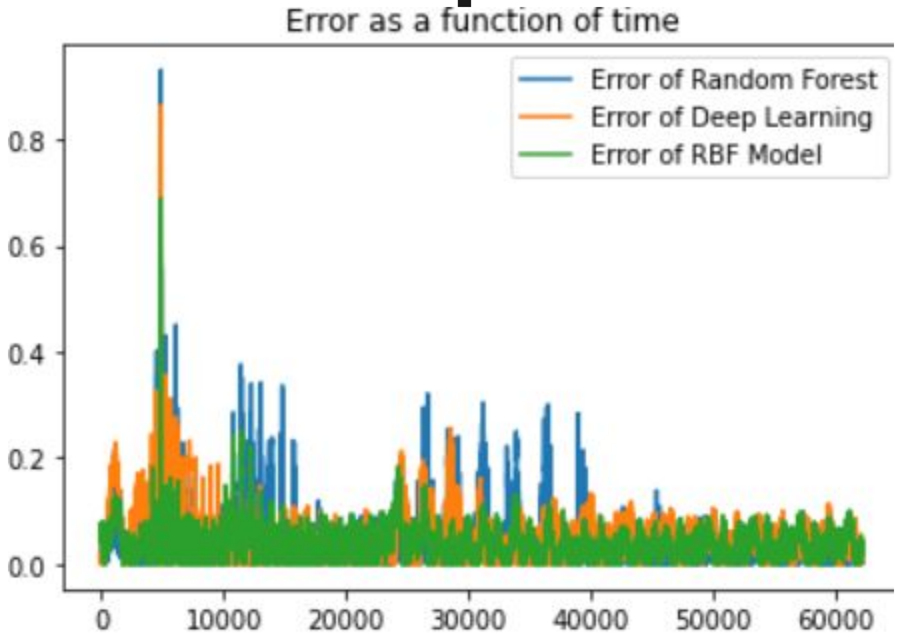
# Deep Learning

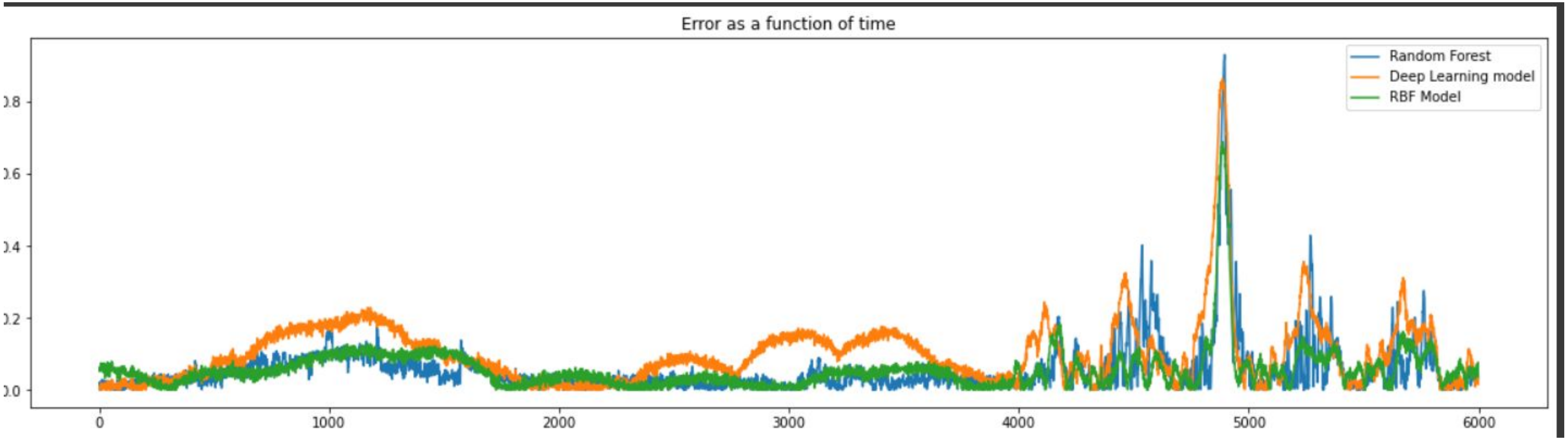
RMSE : 0.0698973



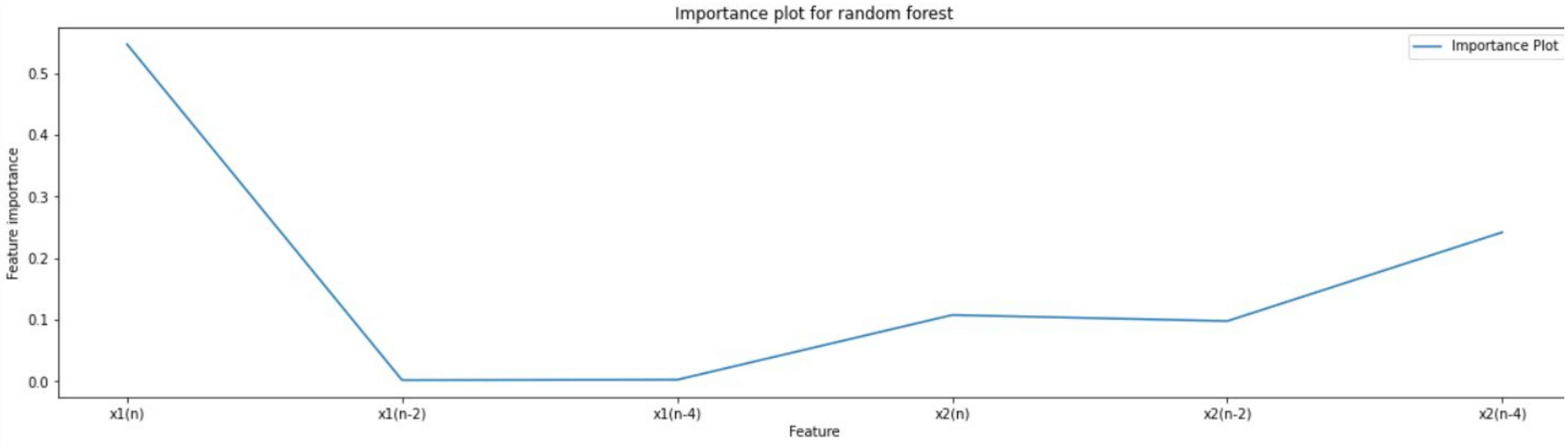
# Error comparison of the three models



# Feature importance

x1(n), x1(n-2), x1(n-4), x2(n), x2(n-2), x2(n-4)



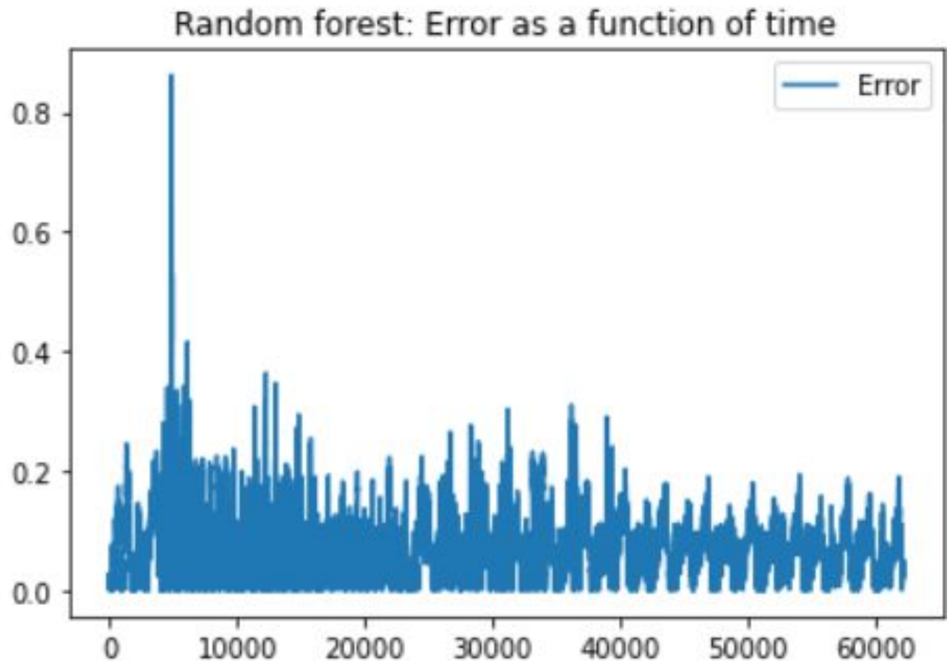
## 5 past inputs

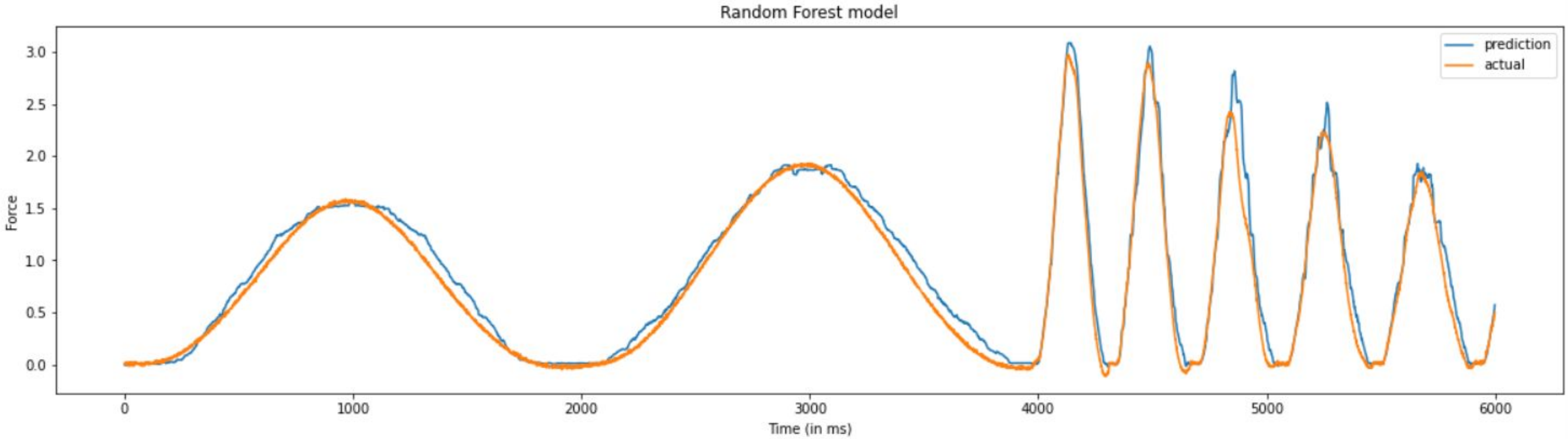
x1(n), x1(n-t), x1(n-2t), x1(n-3t) x1(n-4t), x2(n), x2(n-t), x2(n-2t), x2(n-3t), x2(n-4t)

t = 2

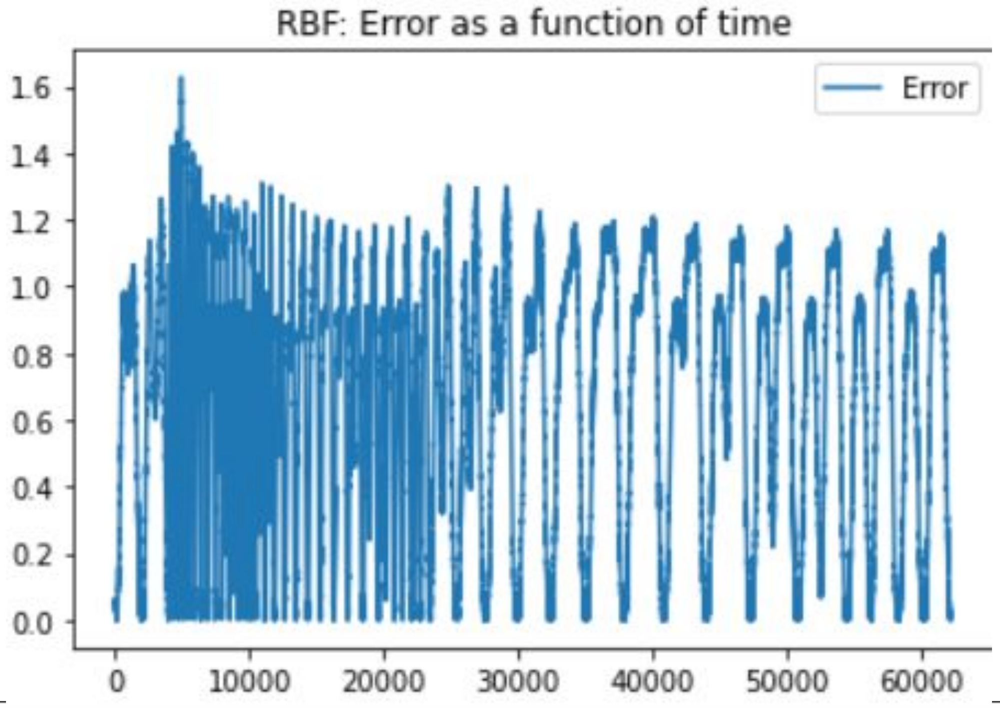
# Random Forest

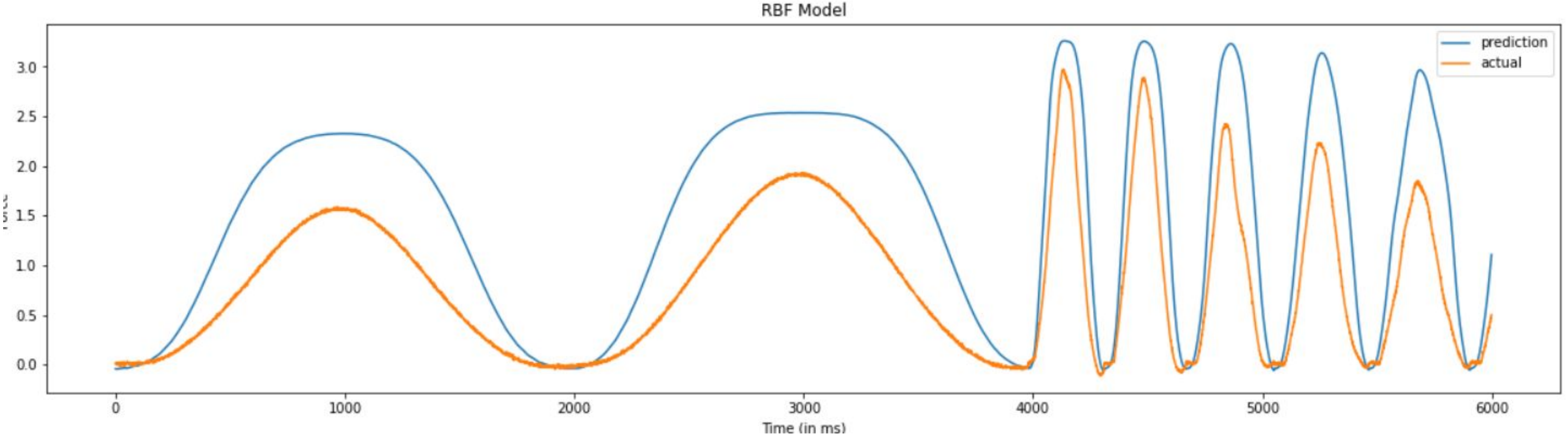
RMSE: 0.09221858



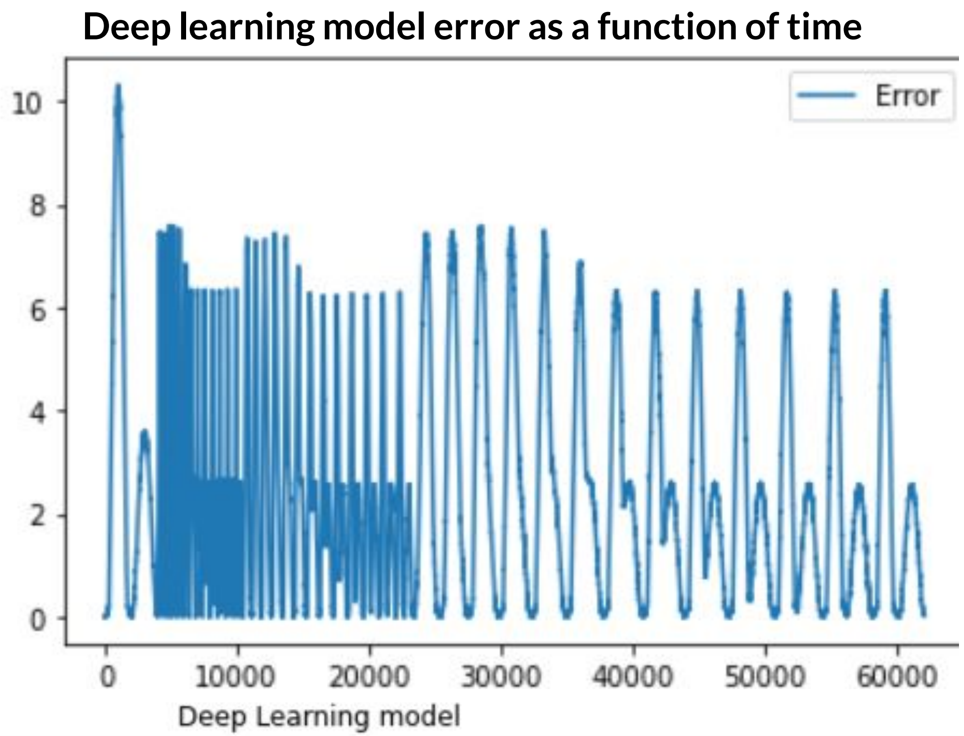
# RBF
RMSE : 0.77797065

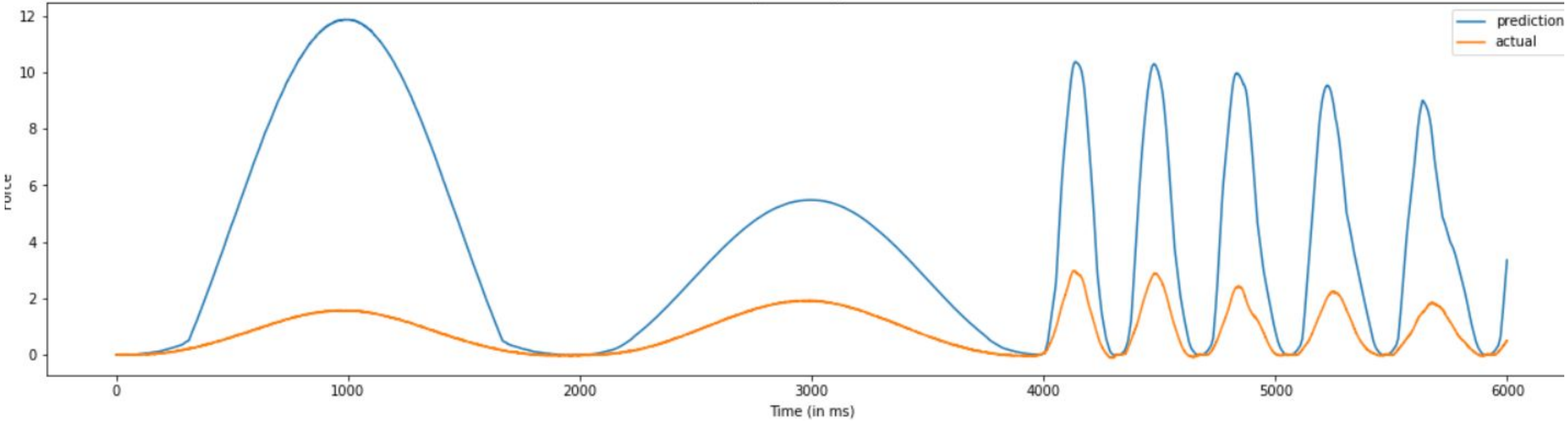


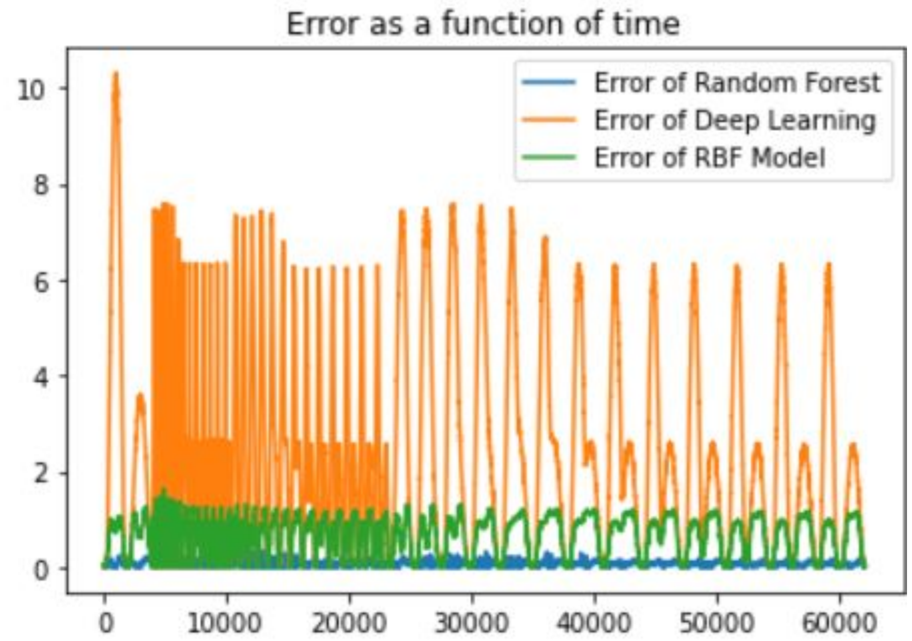
# Deep Learning

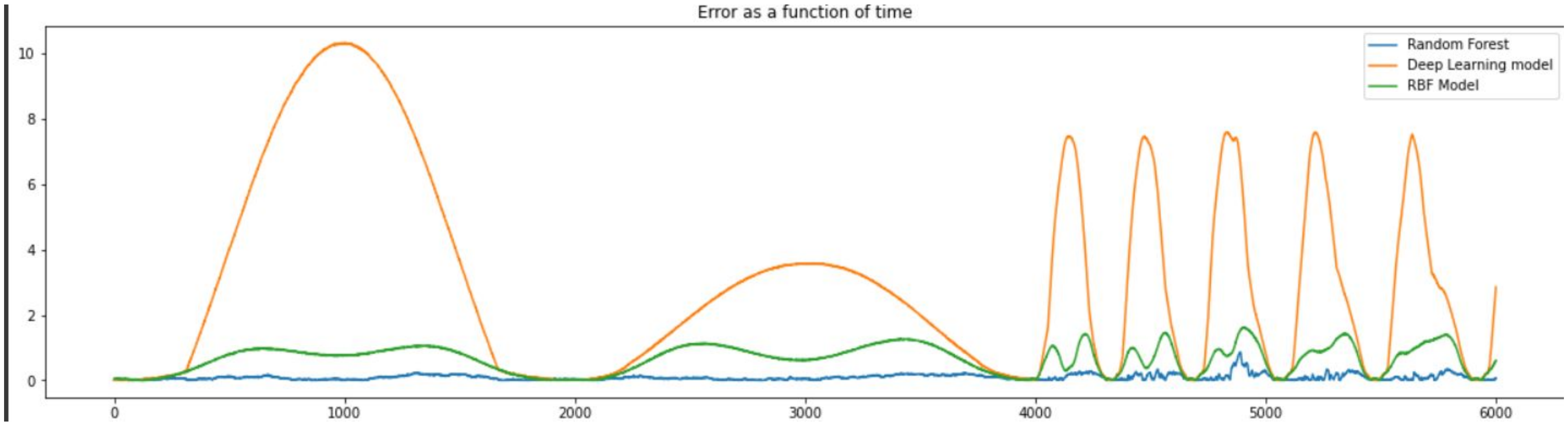
RMSE : `3.400057485`



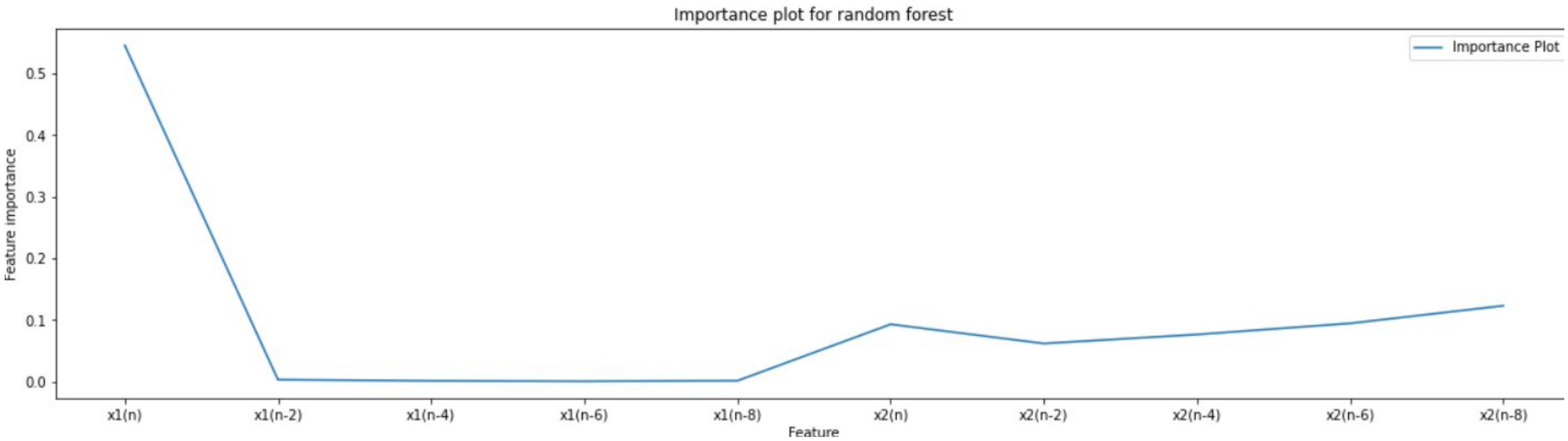



# Error comparison of the three models



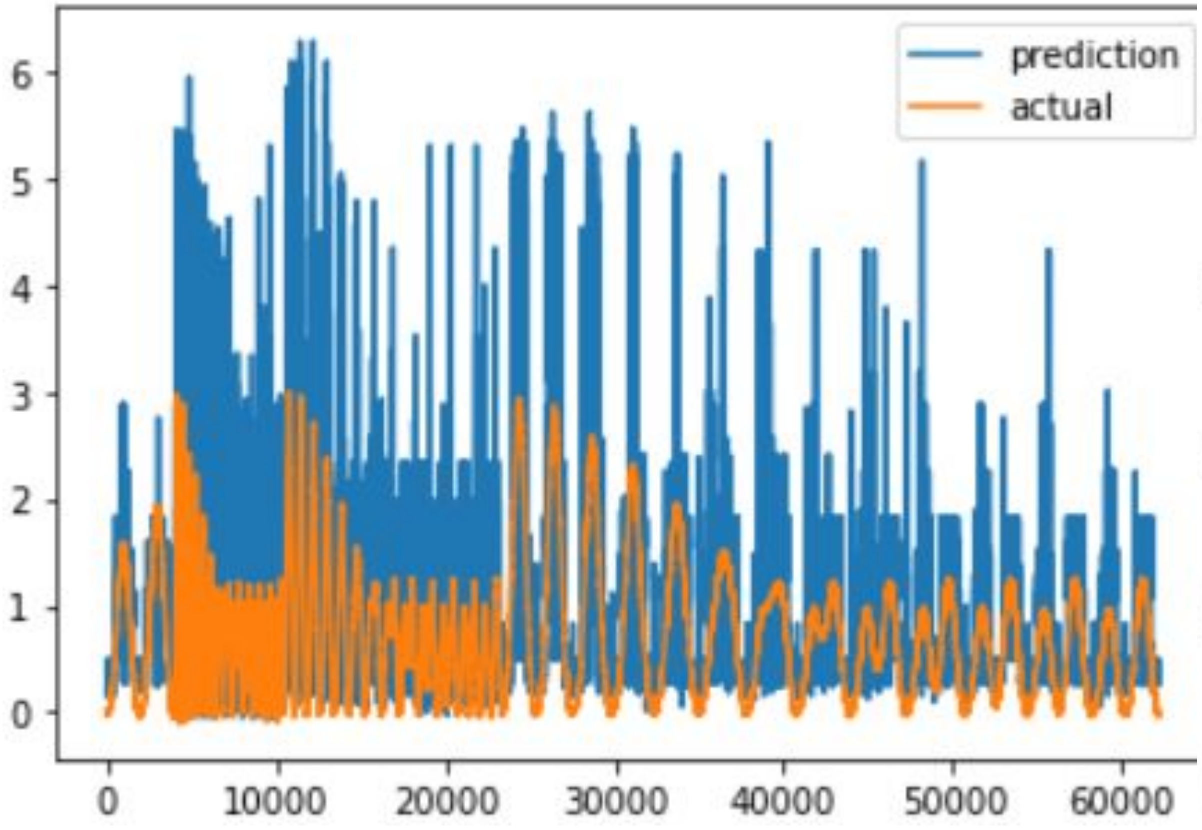
# Feature importance (given 5 past inputs)

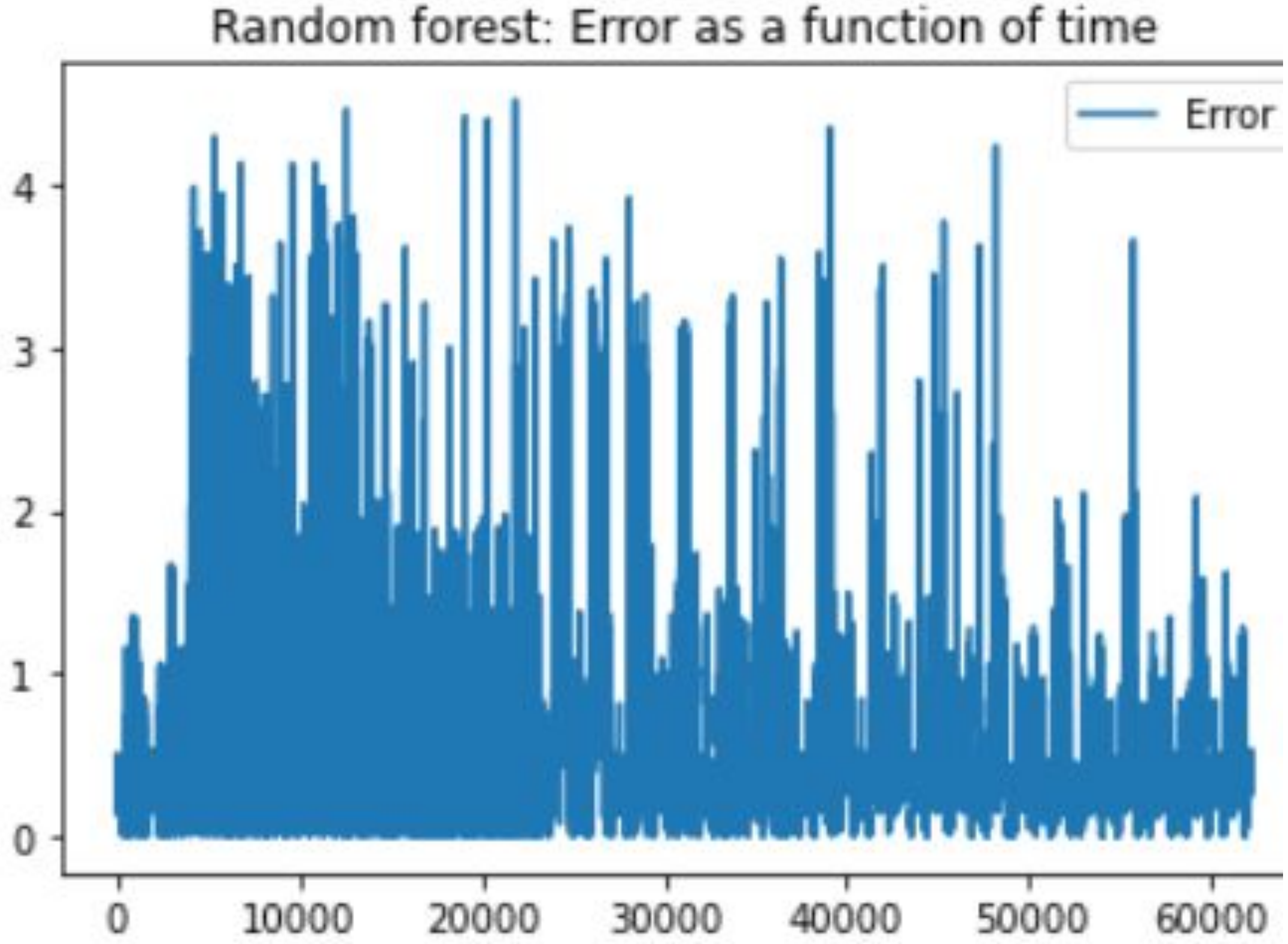


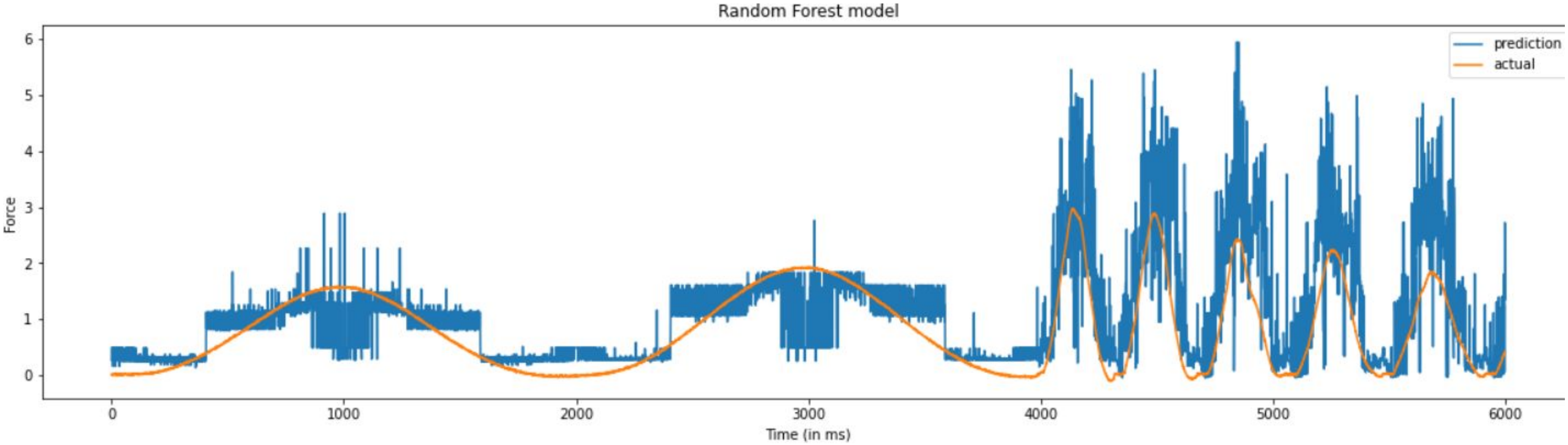
# Considering velocity as an input feature (dx/dt)

# Random Forest

RMSE : `0.5483032`



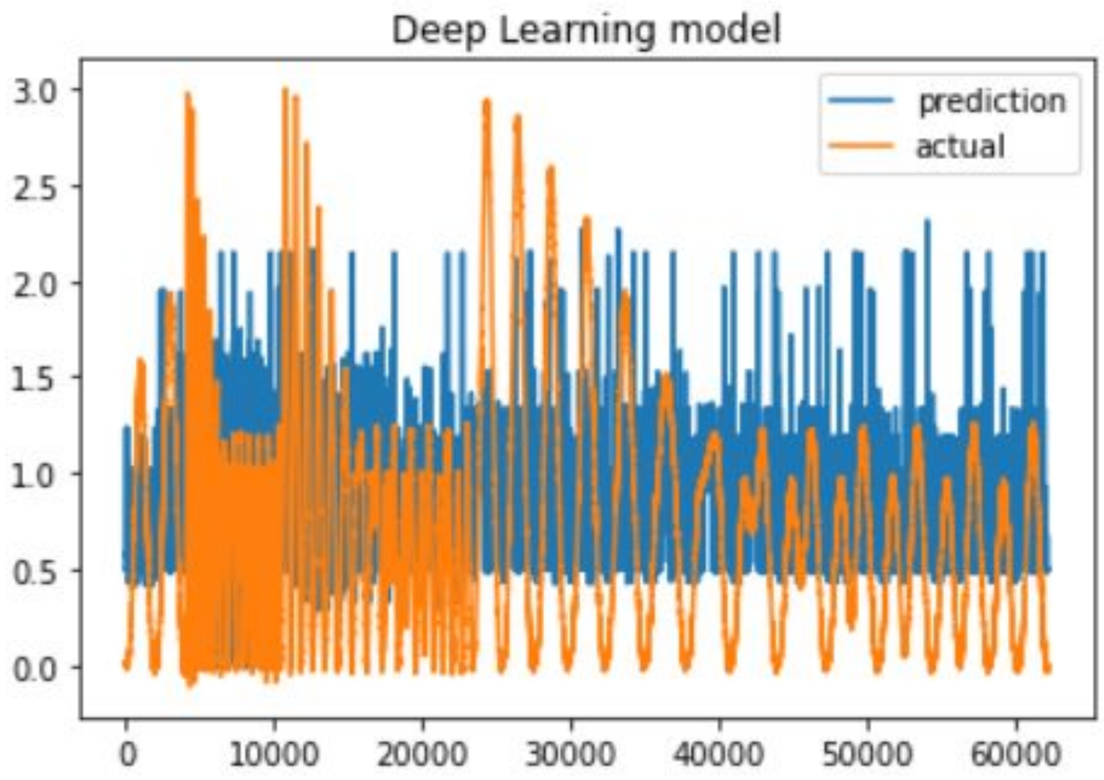



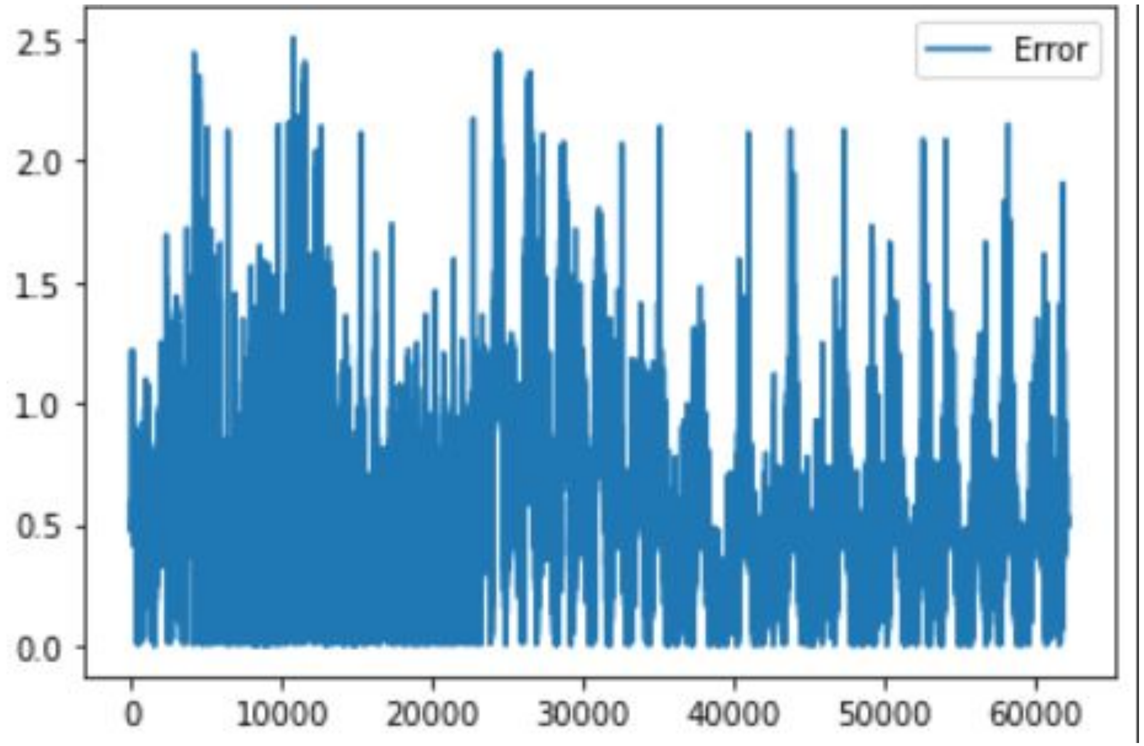


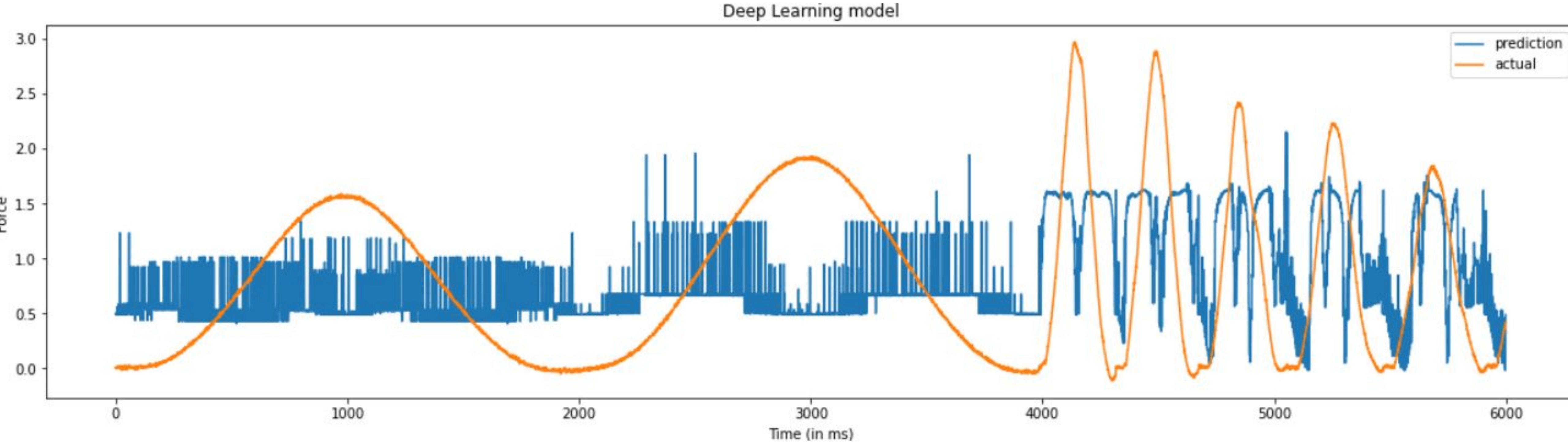
Random Forest model
prediction
actual
Force
Time (in ms)
0
1000
2000
3000
4000
5000
6000
1
2
3
4
5
6

# Deep Learning

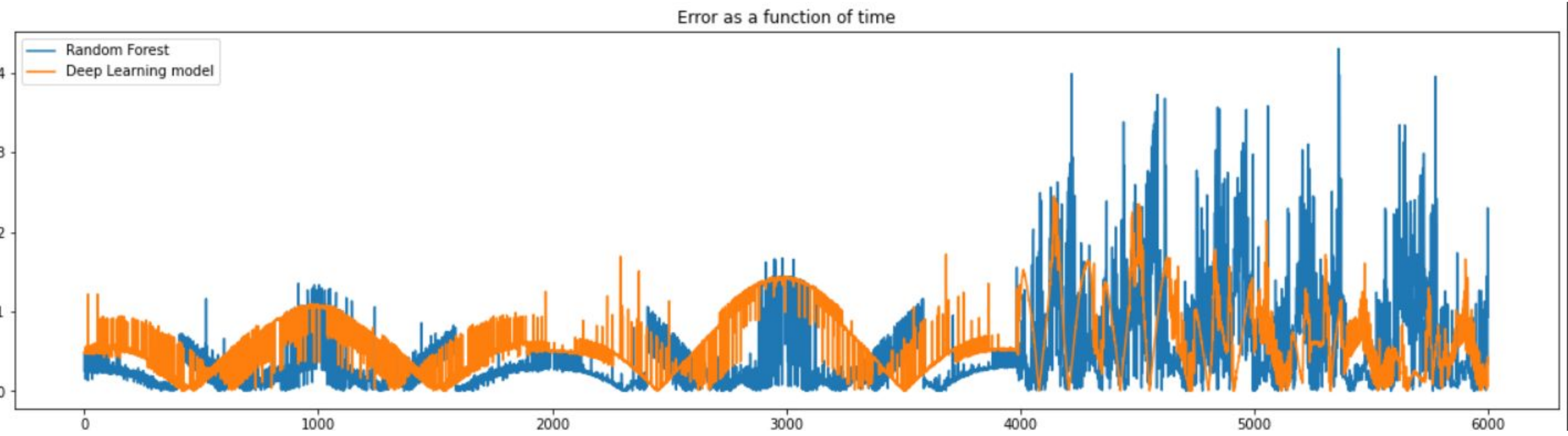
RMSE : `0.39842286`



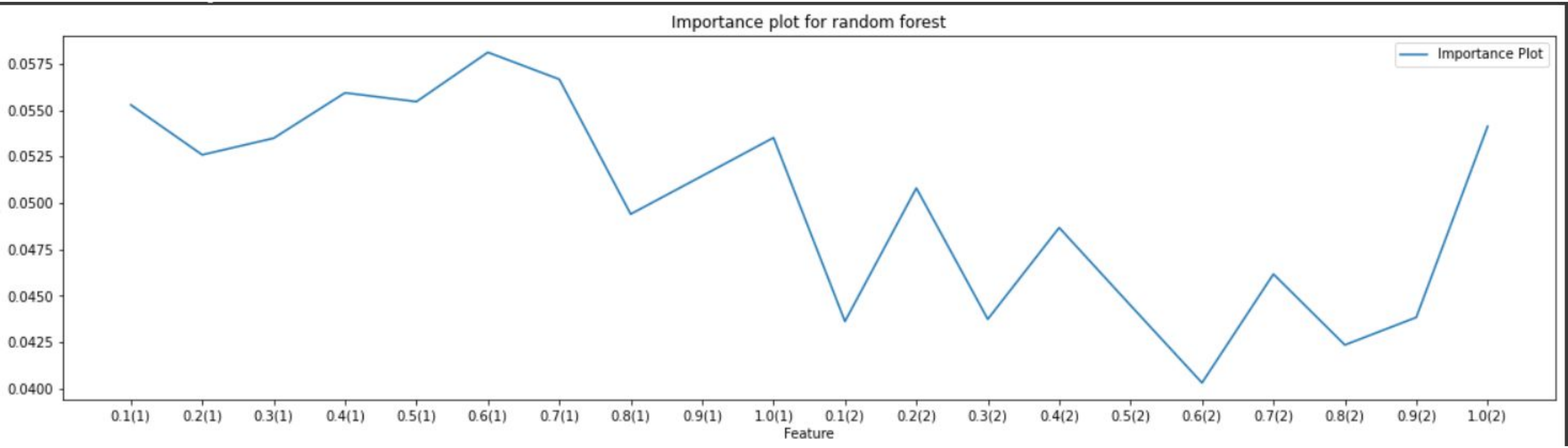




Deep Learning model
prediction
actual
Force
Time (in ms)
0
1000
2000
3000
4000
5000
6000
0.0
0.5
1.0
1.5
2.0
2.5
3.0

# Error comparison b/w Random Forest vs Deep learning model



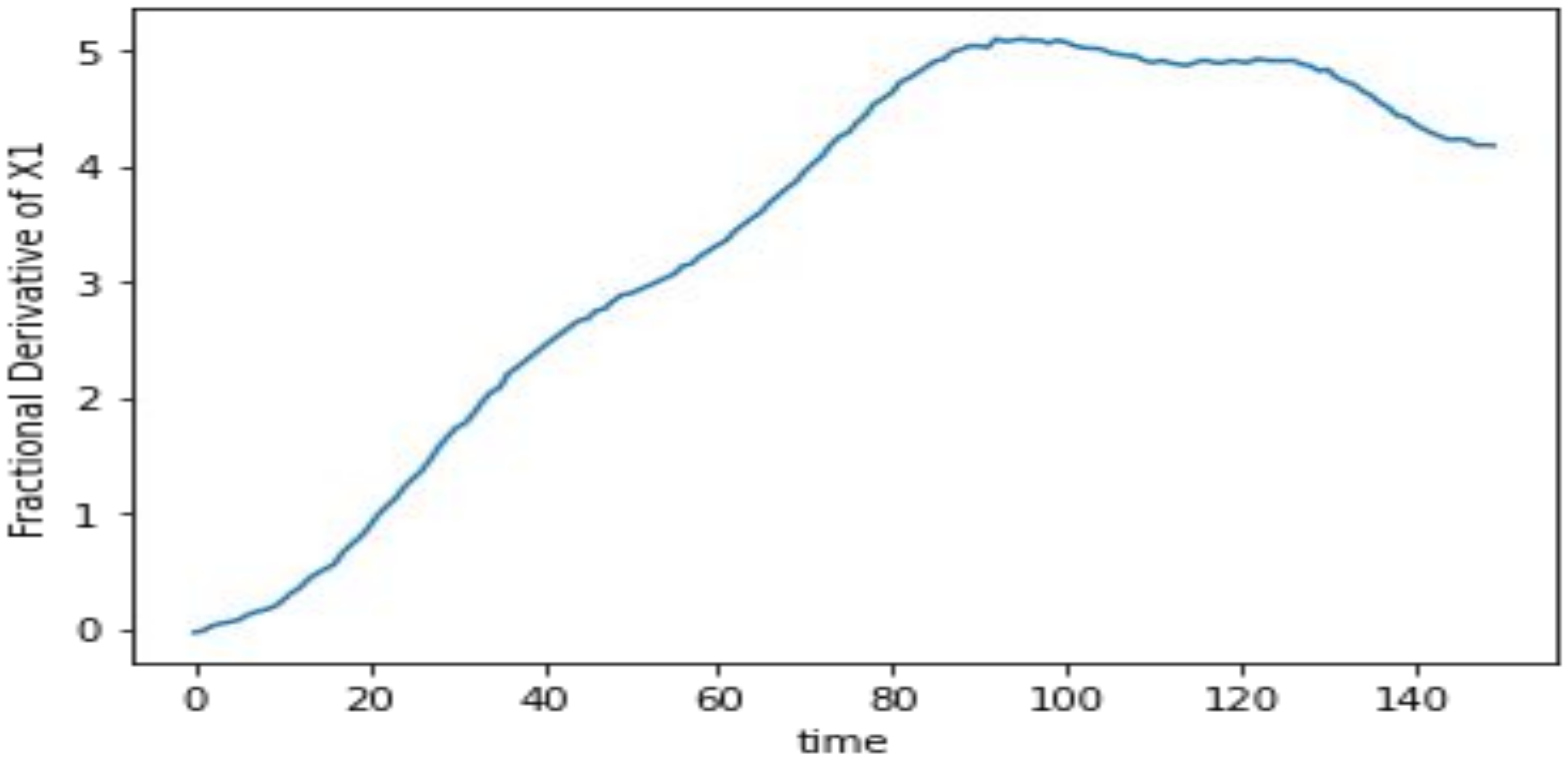
# Feature Importance for different values of α



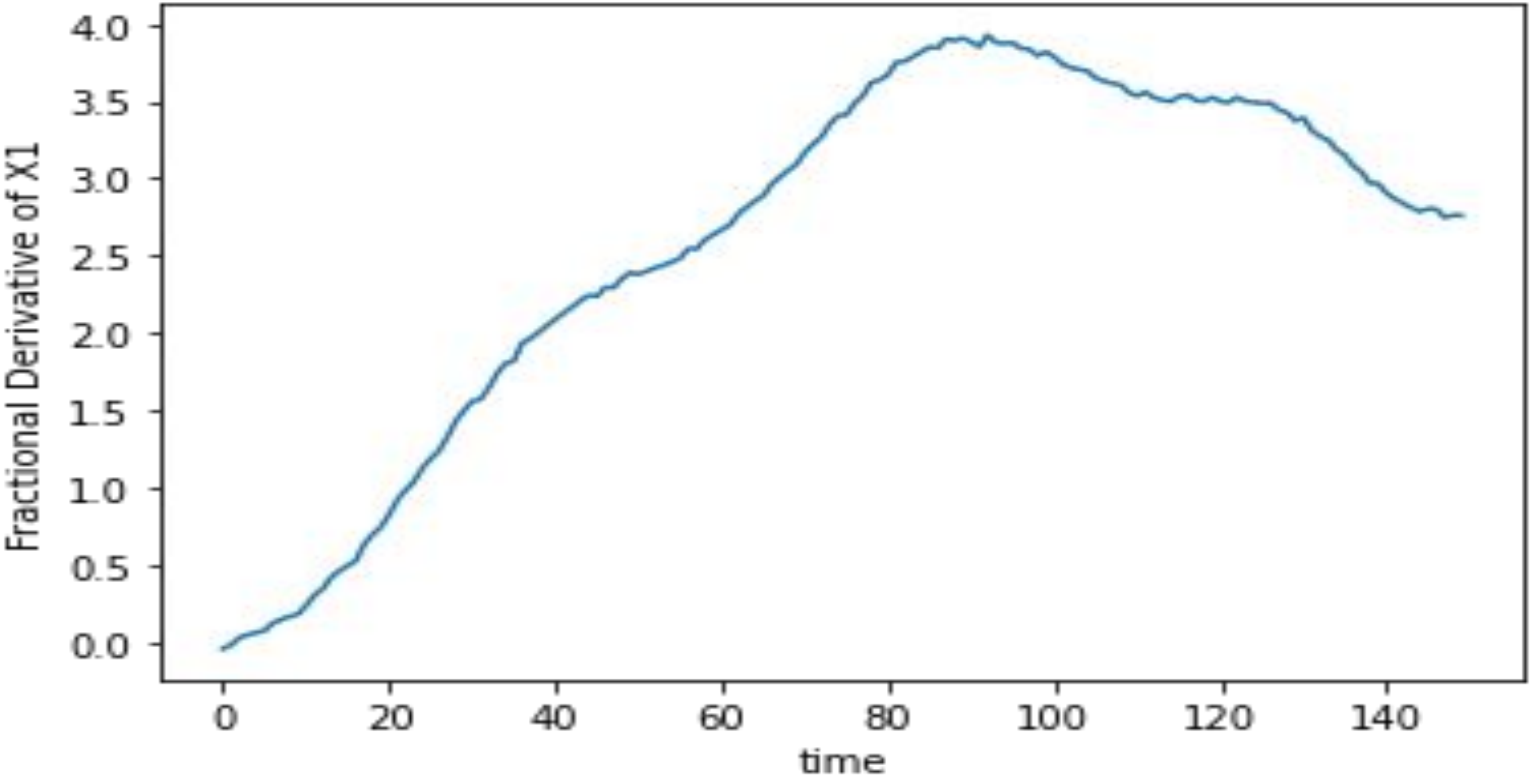
## Results:
## Using Fractional Derivatives of positions as input features with different values of α

**α ∈** [0.5, 0.6, 0.7, 0.8, 0.9]

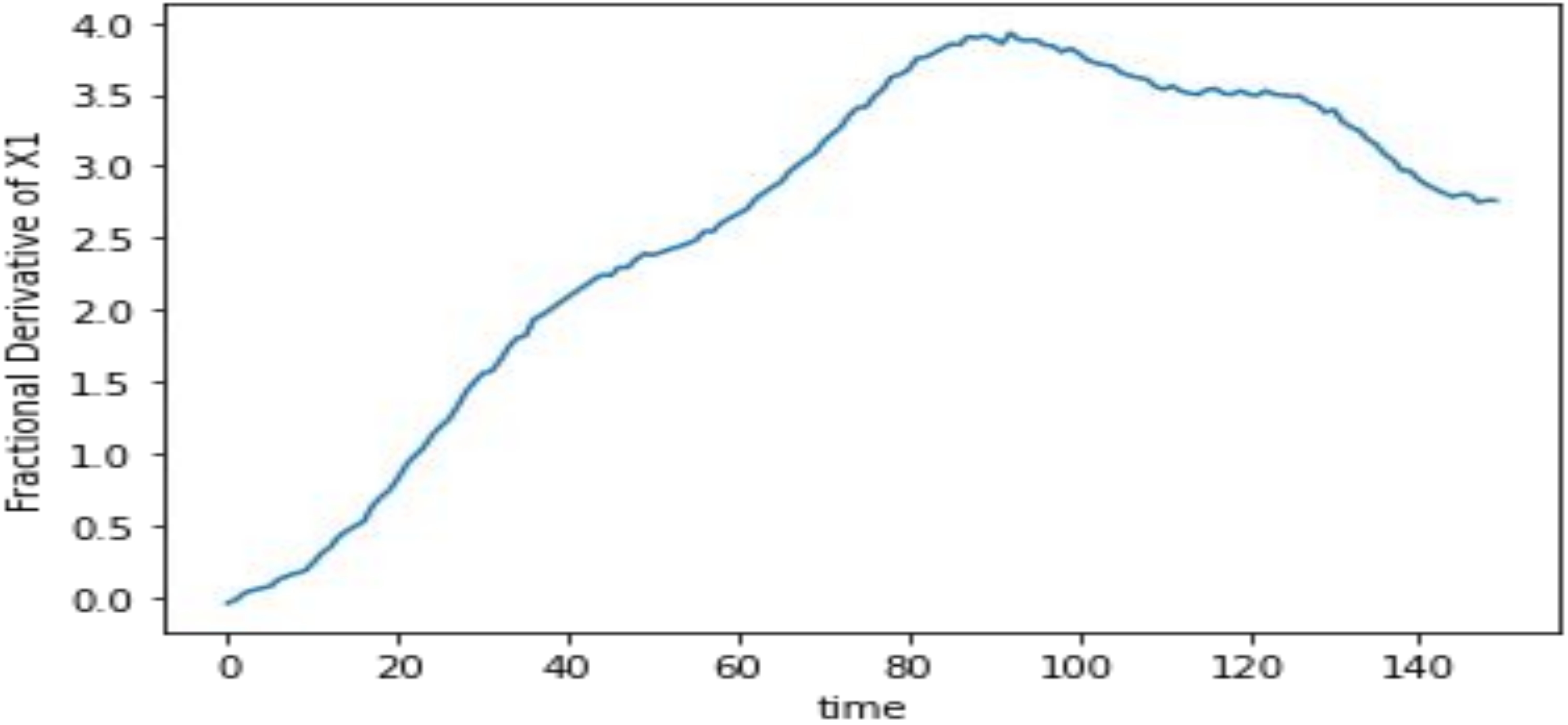
# *Fractional calculus of X1 at order = 0.5 using python*



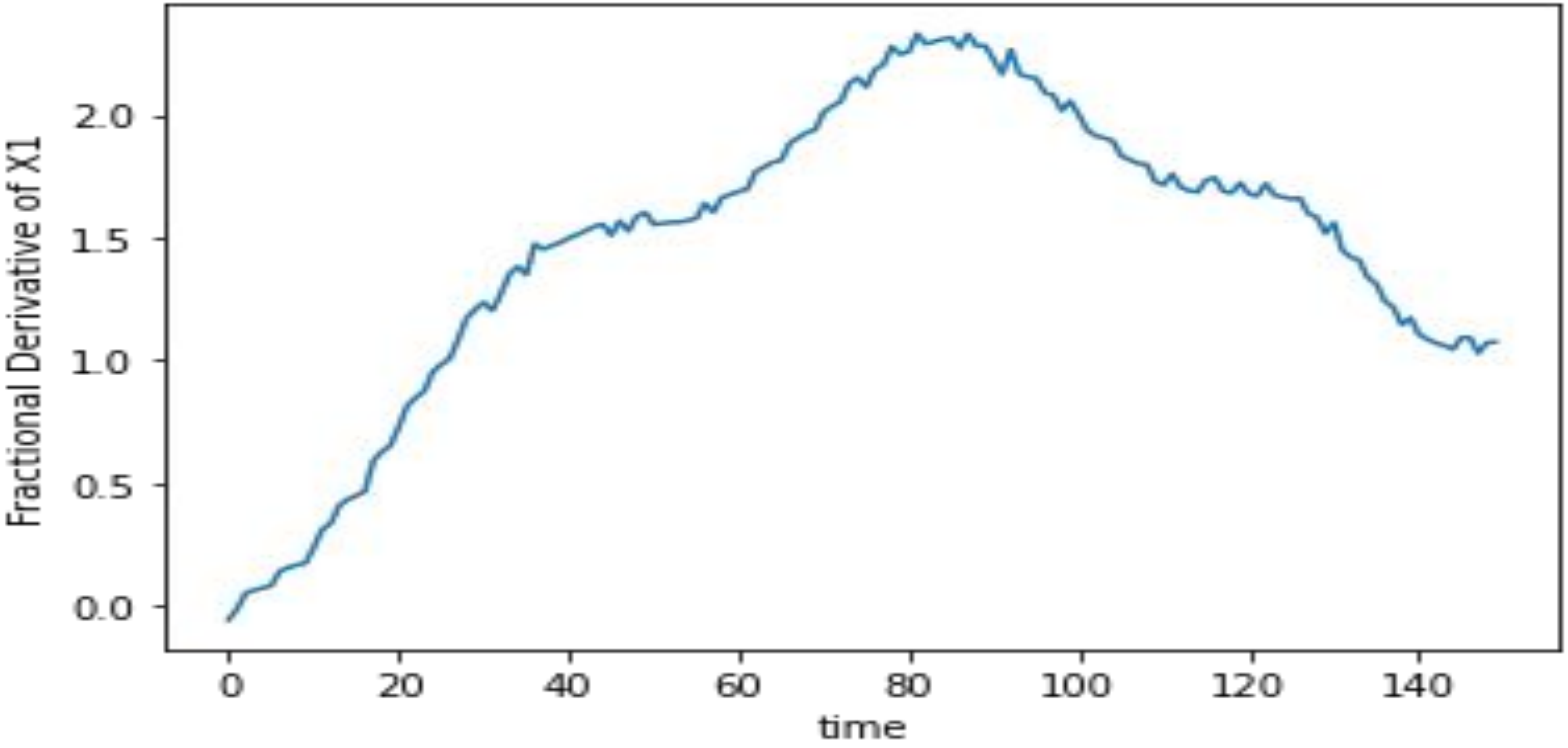
# *Fractional calculus of X1 at order = 0.6 using python*



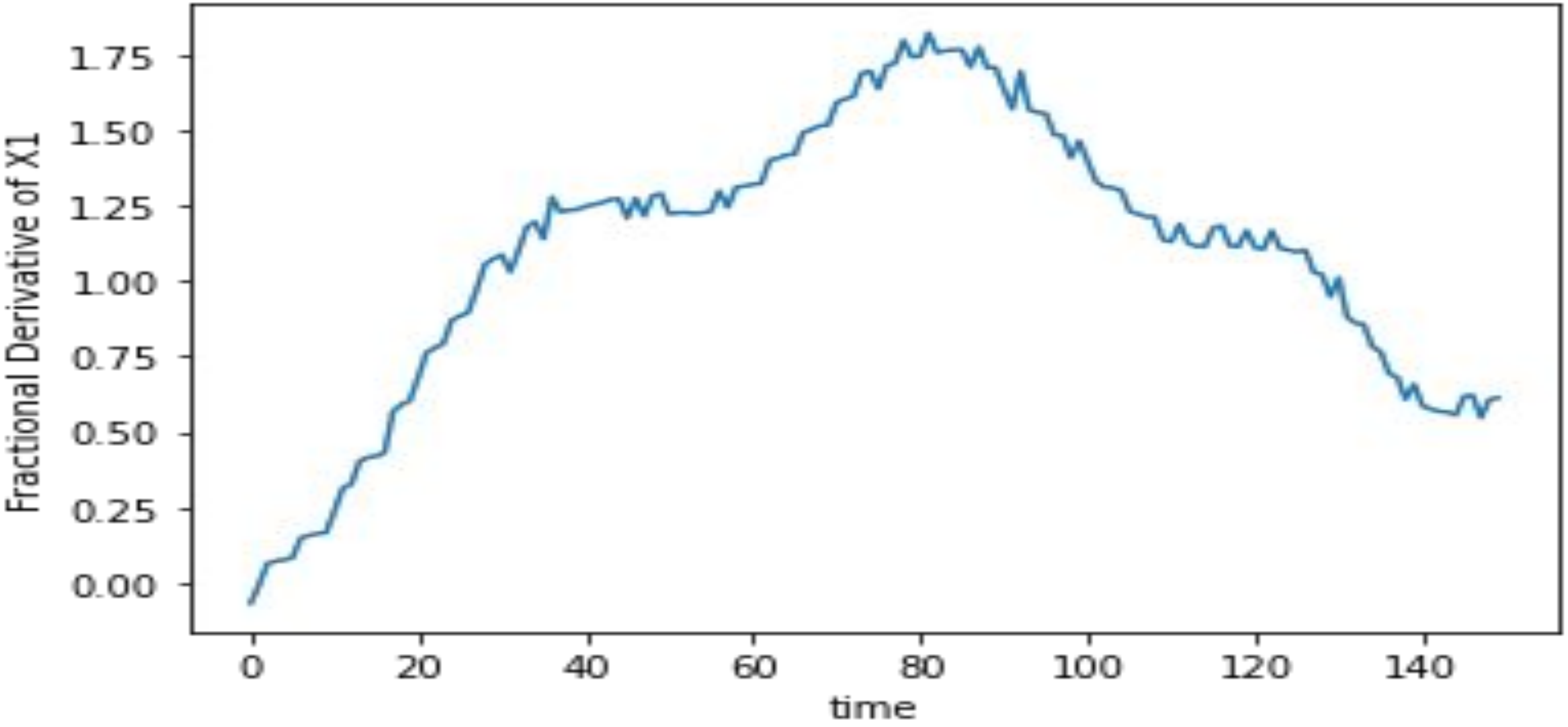
# *Fractional calculus of X1 at order = 0.7 using python*



# *Fractional calculus of X1 at order = 0.8 using python*



# *Fractional calculus of X1 at order = 0.9 using python*



# *Fractional calculus of X2 at order = 0.9 using python*