1. **Experiment 0**

* Input

**intputFeatureList = ['open', 'high', 'low', 'close', 'volume']**

**(prevMin, predMin) = (60, 15)**

**batch\_size = 32**

**epochs = 100**

* Model

model = Sequential()

model.add(LSTM(units=100, return\_sequences=True, input\_shape=(train\_X.shape[1], 1)))

model.add(Dropout(0.2))

model.add(LSTM(units=50, return\_sequences=True))

model.add(Dropout(0.2))

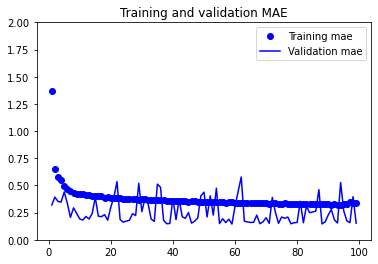
model.add(Flatten())

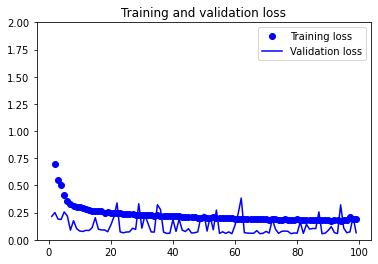
model.add(Dense(units=15))

model.compile(optimizer='adam', loss='MeanSquaredError', metrics=\

['mean\_absolute\_error'])

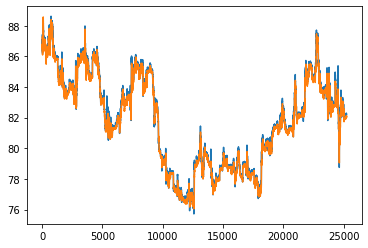
* Training Callback

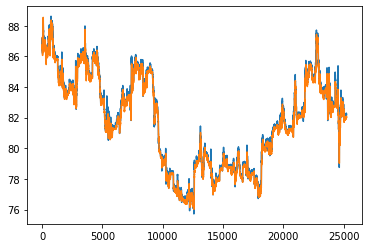


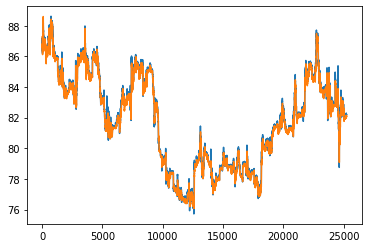


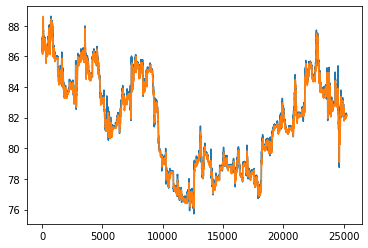
* Output

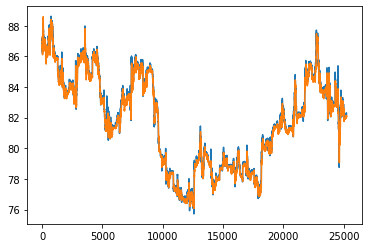
0.0972965657711029 0.23238684237003326

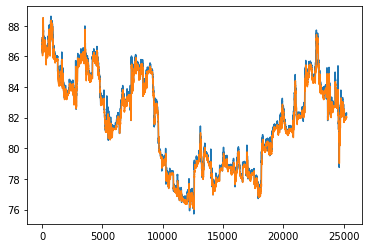


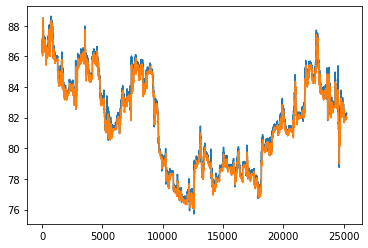


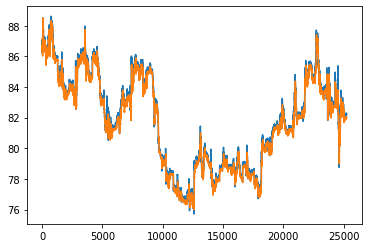


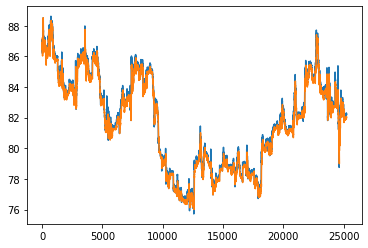


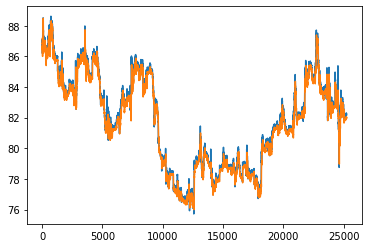


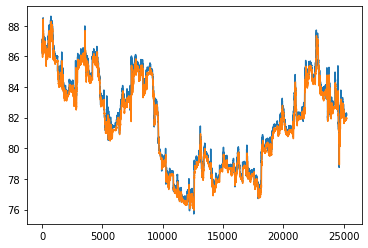


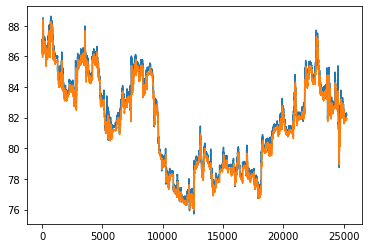


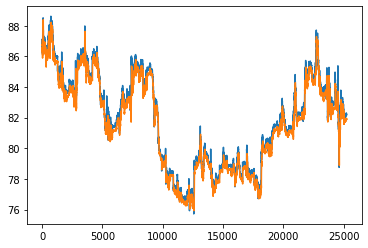












1. **Experiment 1**

* Input

**intputFeatureList = ['open', 'high', 'low', 'close']**

**(prevMin, predMin) = (60, 15)**

**batch\_size = 64**

**epochs = 150**

* Model

# Build the model

model = Sequential()

model.add(LSTM(units=100, return\_sequences=True, input\_shape=(train\_X.shape[1], 1)))

model.add(Dropout(0.2))

model.add(LSTM(units=50, return\_sequences=True))

model.add(Dropout(0.2))

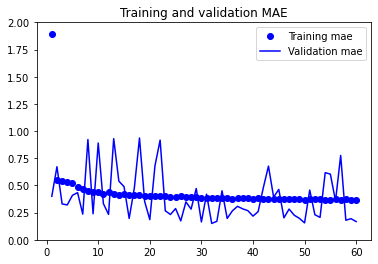
model.add(Flatten())

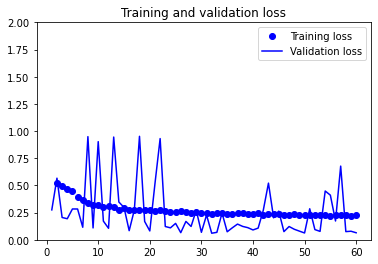
model.add(Dense(units=15))

model.compile(optimizer='adam', loss='MeanSquaredError', metrics=\

['mean\_absolute\_error'])

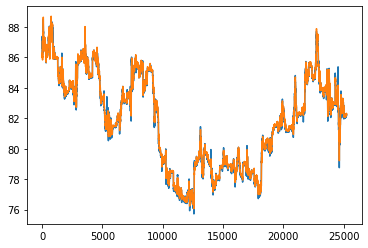
* Training Callback

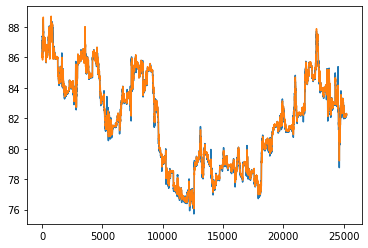


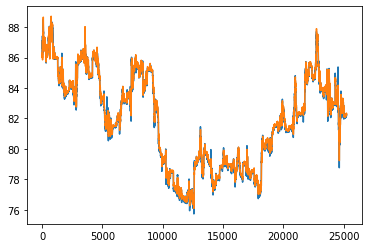


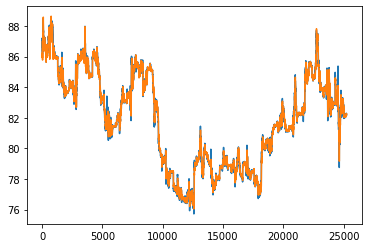
* Output

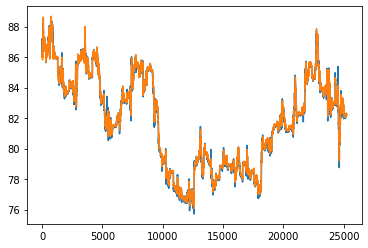
0.0708729699254036 0.179587259888649

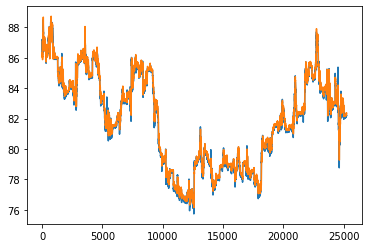


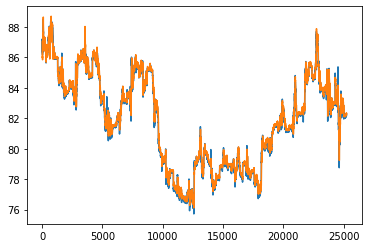


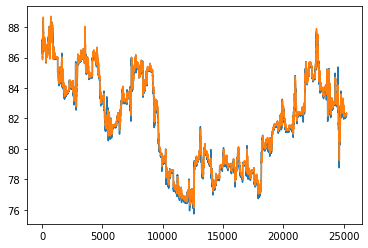


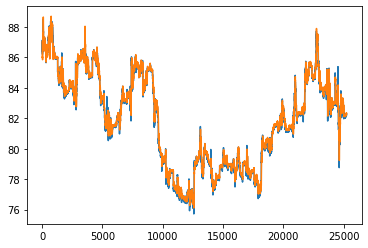


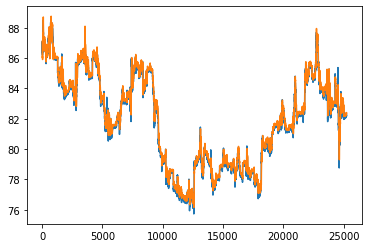


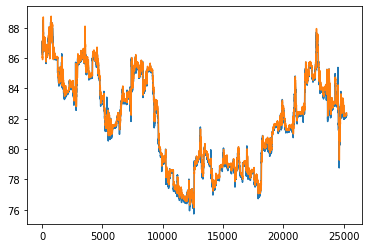


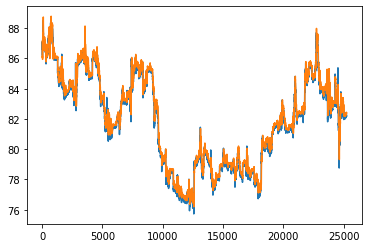


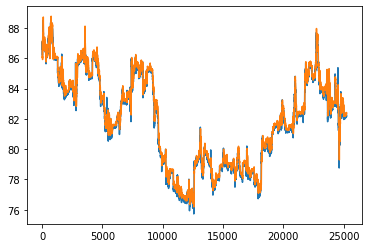


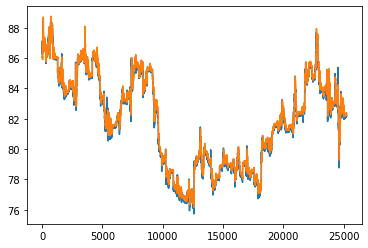


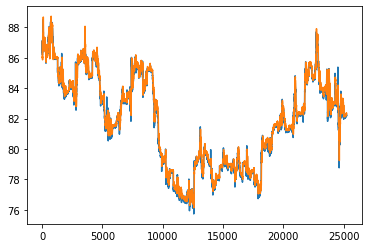












1. **Experiment 2**

* Input

**intputFeatureList = ['open', 'high', 'low', 'close', 'volume']**

**(prevMin, predMin) = (60, 15)**

**batch\_size = 32**

**epochs = 150**

* Model

# Build the model

model = Sequential()

model.add(LSTM(units=100, return\_sequences=True, input\_shape=(train\_X.shape[1], 1)))

model.add(Dropout(0.4))

model.add(LSTM(units=100, return\_sequences=True, input\_shape=(train\_X.shape[1], 1)))

model.add(Dropout(0.4))

model.add(LSTM(units=50, return\_sequences=True))

model.add(Dropout(0.4))

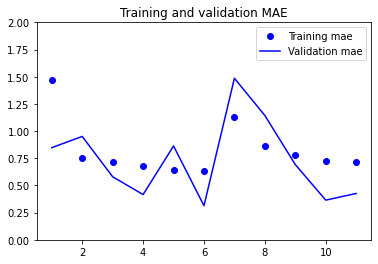
model.add(Flatten())

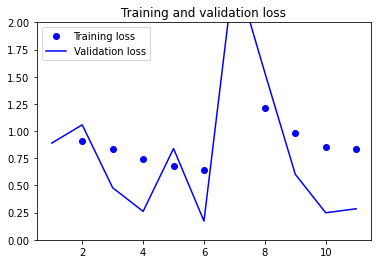
model.add(Dense(units=15))

model.compile(optimizer='adam', loss='MeanSquaredError', metrics=\

['mean\_absolute\_error'])

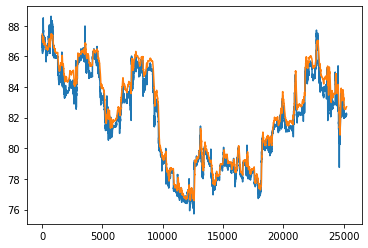
* Training Callback

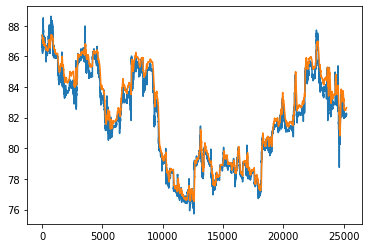


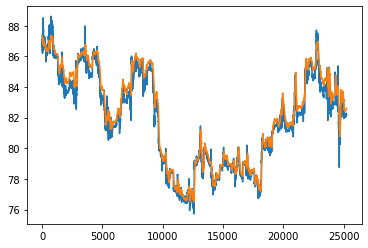


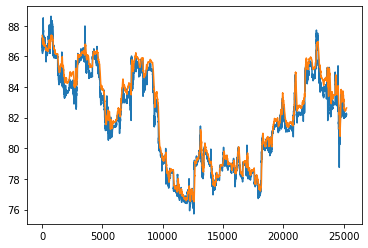
* Output

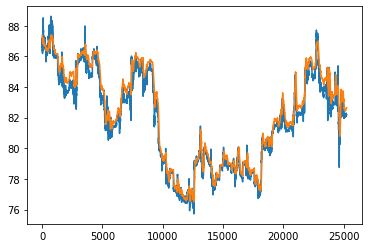
0.33037087321281433 0.4517090916633606

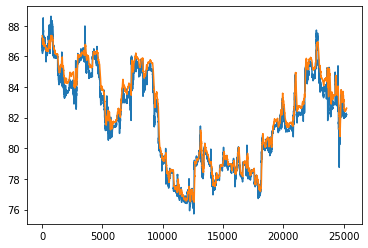


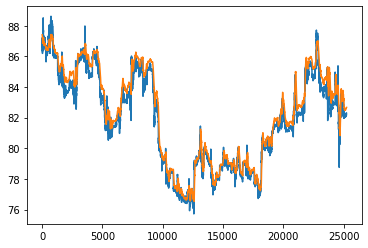


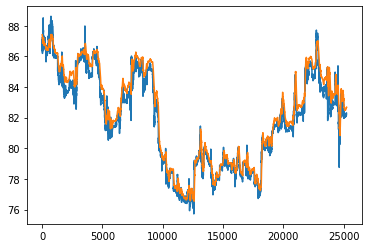


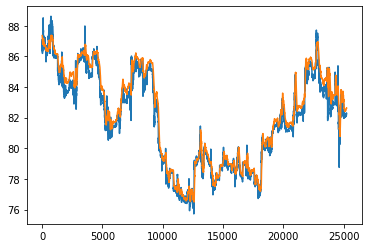


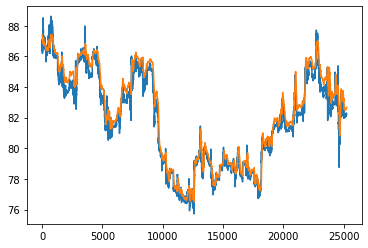


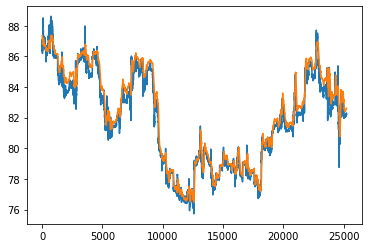


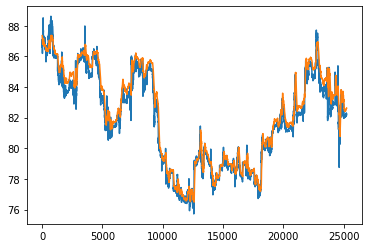


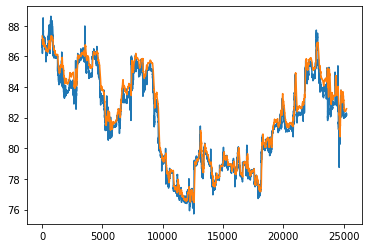


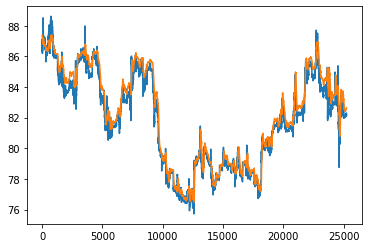


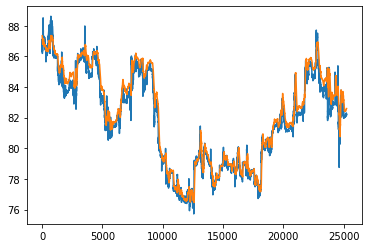












1. **Experiment 3**

* Input

**intputFeatureList = ['open', 'high', 'low', 'close', 'volume']**

**(prevMin, predMin) = (120, 15)**

**batch\_size = 32**

**epochs = 150**

* Model

model = Sequential()

model.add(LSTM(units=100, return\_sequences=True, input\_shape=(train\_X.shape[1], 1)))

model.add(Dropout(0.2))

model.add(LSTM(units=50, return\_sequences=True))

model.add(Dropout(0.2))

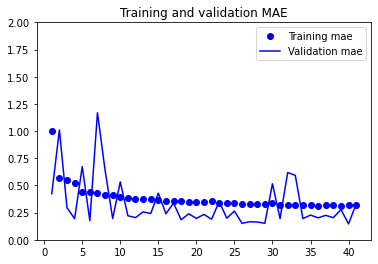
model.add(Flatten())

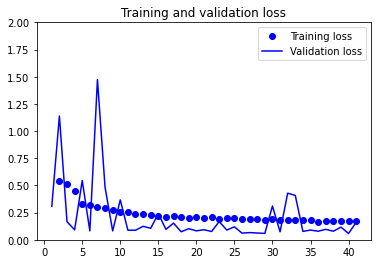
model.add(Dense(units=15))

model.compile(optimizer='adam', loss='MeanSquaredError', metrics=\

['mean\_absolute\_error'])

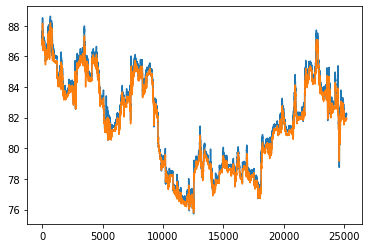
* Training Callback

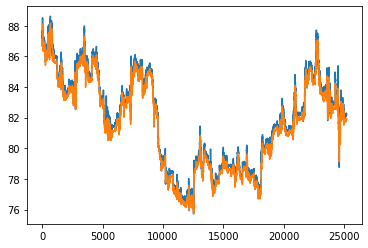


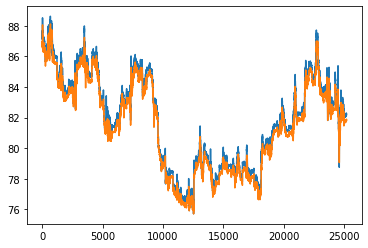


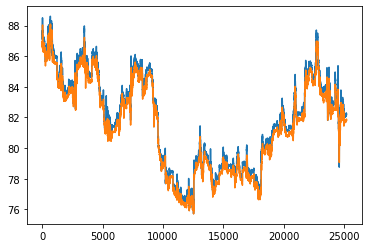
* Output

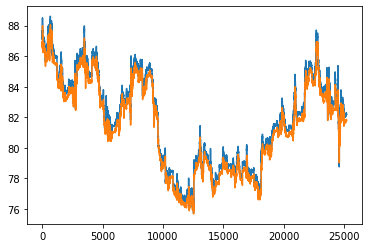
0.2070915549993515 0.39458373188972473

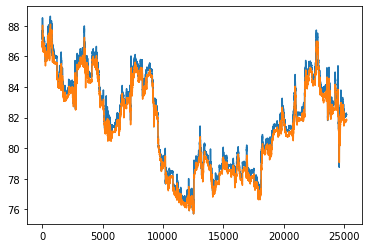


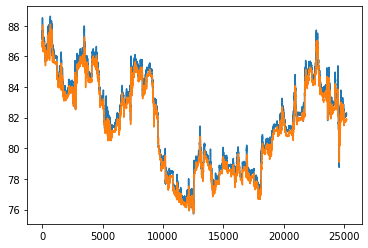


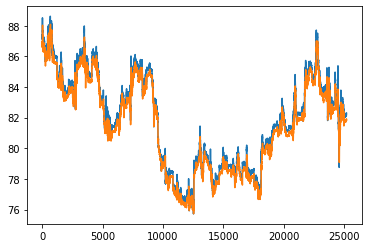


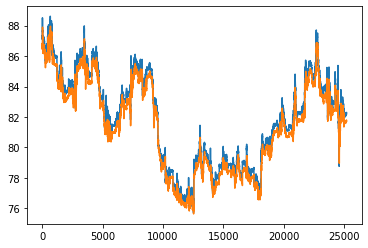


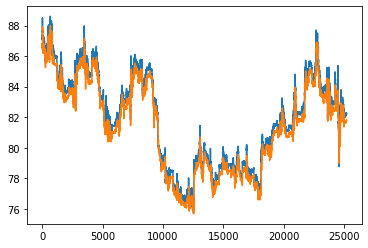


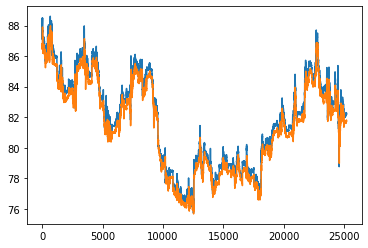


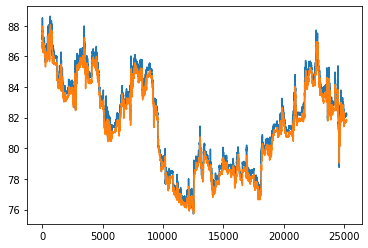


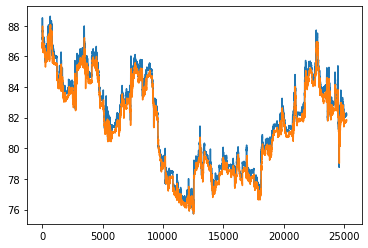


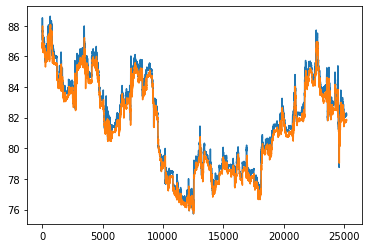


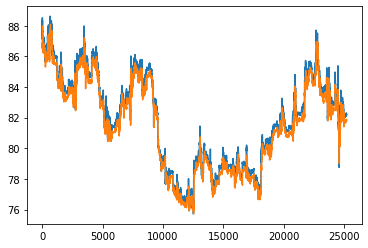












1. **Experiment 0**

* Input

**intputFeatureList = ['open', 'high', 'low', 'close', 'volume']**

**(prevMin, predMin) = (120, 15)**

**batch\_size = 32**

**epochs = 150**

* Model

model = Sequential()

model.add(LSTM(units=100, return\_sequences=True, input\_shape=(train\_X.shape[1], 1)))

model.add(Dropout(0.4))

model.add(LSTM(units=100, return\_sequences=True, input\_shape=(train\_X.shape[1], 1)))

model.add(Dropout(0.4))

model.add(LSTM(units=50, return\_sequences=True))

model.add(Dropout(0.4))

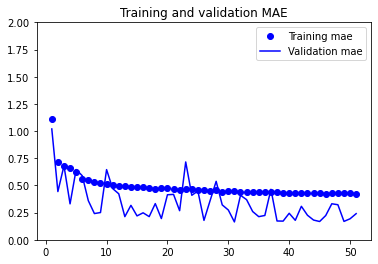
model.add(Flatten())

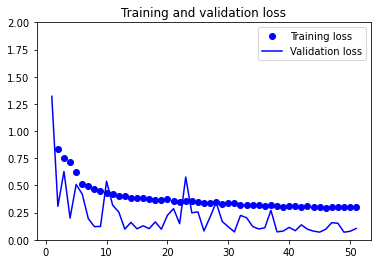
model.add(Dense(units=15))

model.compile(optimizer='adam', loss='MeanSquaredError', metrics=\

['mean\_absolute\_error'])

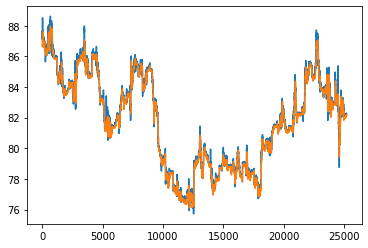
* Training Callback

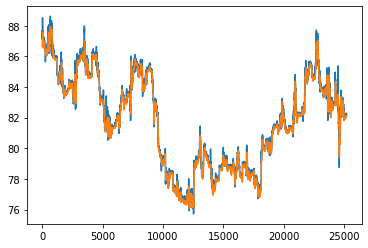


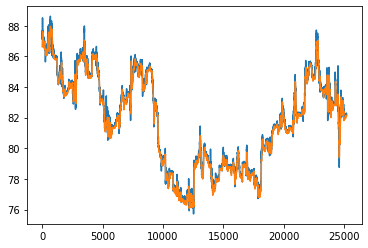


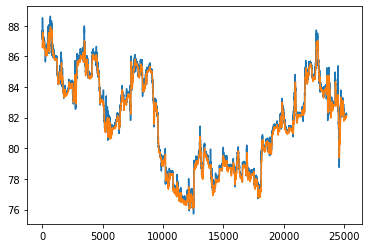
* Output

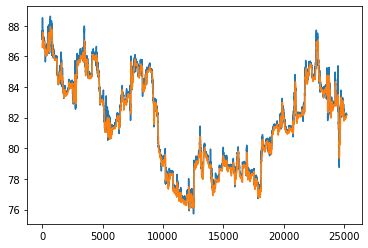
0.10259725153446198 0.22807930409908295

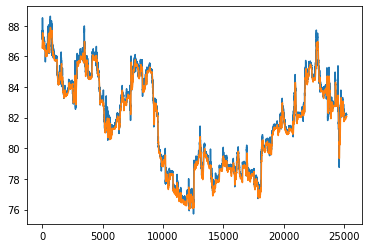


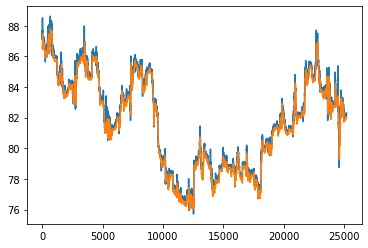


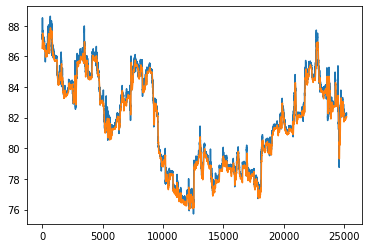


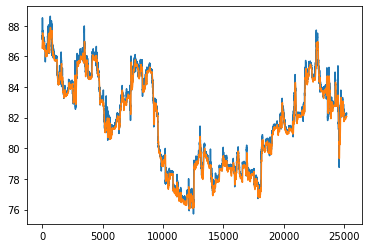


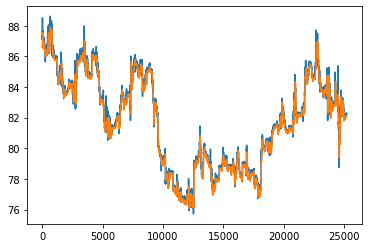


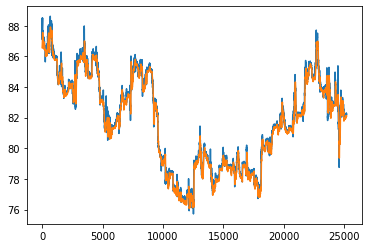


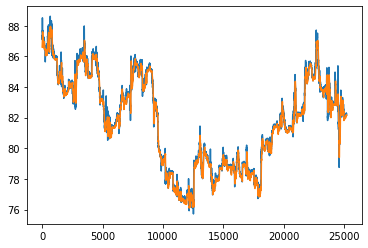


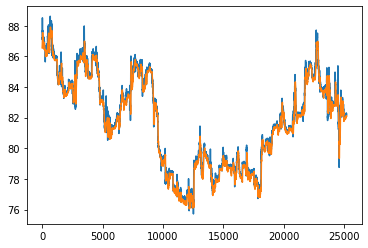


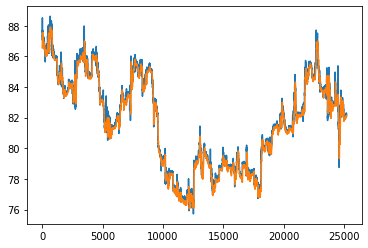


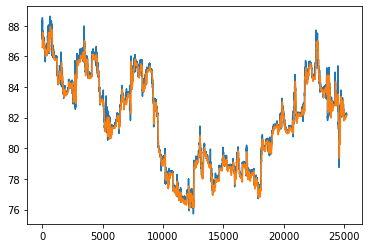












**RESULT**

| **Input Parameter** | **Model** | **Output** |
| --- | --- | --- |
| **features=[open, high, low, close, volume]**  **(PrevMin, PredMin)=(60, 15)**  **batch\_size=32**  **epochs=100** | **LSTM(100), Dropout(0.2), LSTM(50), Dropout(0.2), Dense(15)** | **loss=0.0972965657711029**  **mae=0.23238684237003326** |
| **features=[open, high, low, close]**  **(PrevMin, PredMin)=(60, 15)**  **batch\_size=64**  **epochs=150** | **LSTM(100), Dropout(0.2), LSTM(50), Dropout(0.2), Dense(15)** | **loss=0.0708729699254036 mae=0.179587259888649** |
| **features=[open, high, low, close, volume]**  **(PrevMin, PredMin)=(60, 15)**  **batch\_size=32**  **epochs=150** | **LSTM(100), Dropout(0.4), LSTM(100), Dropout(0.4), LSTM(50), Dropout(0.4), Dense(15)** | **loss=0.33037087321281433**  **mae=0.4517090916633606** |
| **features=[open, high, low, close, volume]**  **(PrevMin, PredMin)=(120, 15)**  **batch\_size=32**  **epochs=150** | **LSTM(100), Dropout(0.2), LSTM(50), Dropout(0.2), Dense(15)** | **loss=0.2070915549993515**  **mae=0.39458373188972473** |
| **features=[open, high, low, close, volume]**  **(PrevMin, PredMin)=(120, 15)**  **batch\_size=32**  **epochs=150** | **LSTM(100), Dropout(0.4), LSTM(100), Dropout(0.4), LSTM(50), Dropout(0.4), Dense(15)** | **loss=0.10259725153446198**  **mae=0.22807930409908295** |