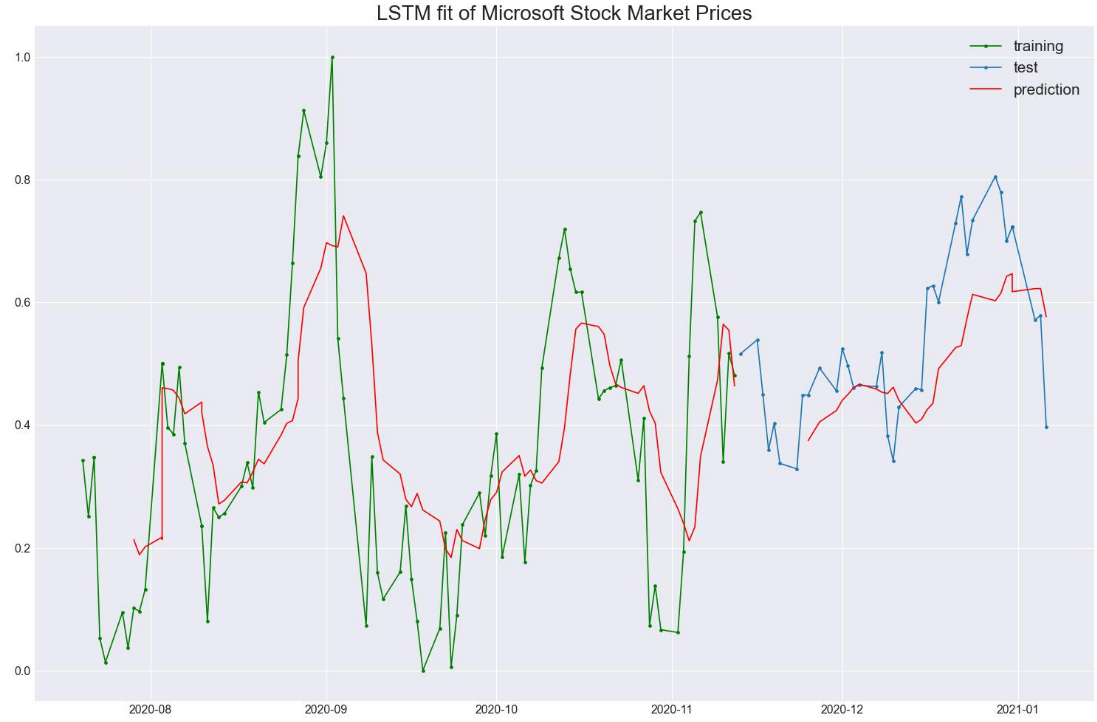
1. Original LSTM model

```
model = Sequential()
model.add(LSTM(200, input_shape=(X_train_features.shape[1],
X_train_features.shape[2])))
model.add(Dropout(0.20))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')

history = model.fit(X_train_features, y_train, epochs=300, batch_size=25,
validation_data=(X_test_features, y_test),
                    callbacks=[EarlyStopping(monitor='val_loss', patience=10)],
verbose=0, shuffle=False)
```

Result



Evaluation

Train Mean Absolute Error: 0.12637061

Train Root Mean Squared Error: 0.16235372

Test Mean Absolute Error: 0.08166171

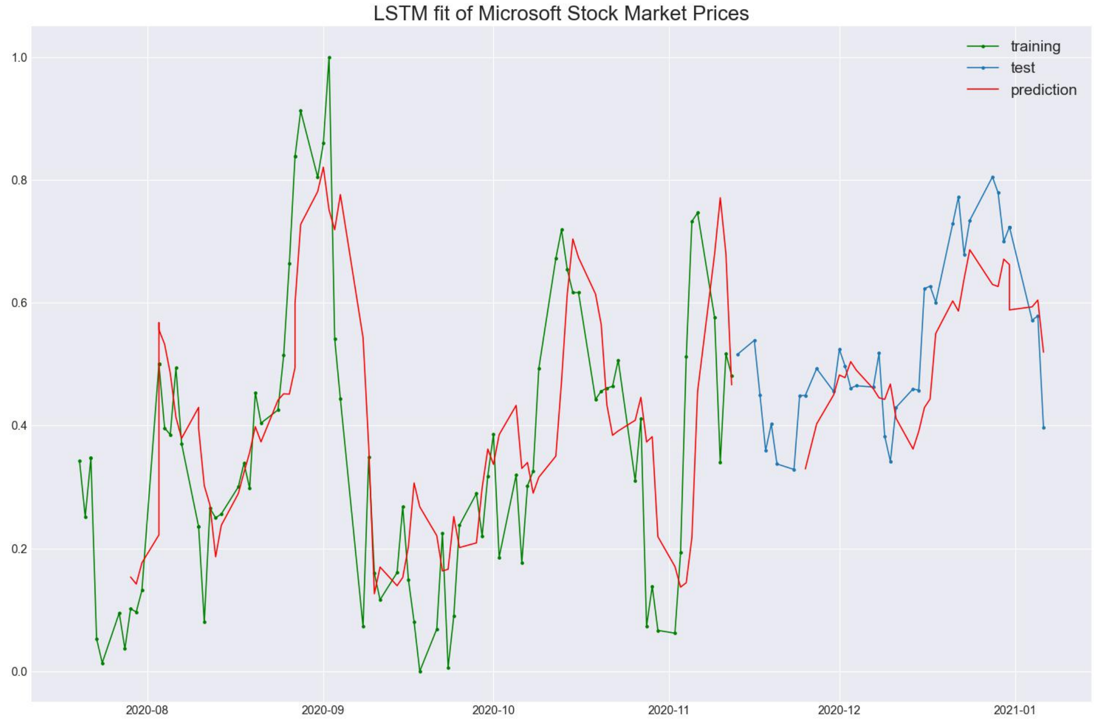
Test Root Mean Squared Error: 0.09914234

2. Model after parameter chosen

```
model = Sequential()
model.add(LSTM(200, input_shape=(X_train_features.shape[1],
X_train_features.shape[2])))
model.add(Dropout(0.20))
model.add(Dense(1))
```

```
history = model.fit(X_train_features, y_train, epochs=300, batch_size=25,
validation_data=(X_test_features, y_test),
                   callbacks=[EarlyStopping(monitor='val_loss', patience=10)],
verbose=0, shuffle=False)
```

Result



Evaluation

Train Mean Absolute Error: 0.10498382

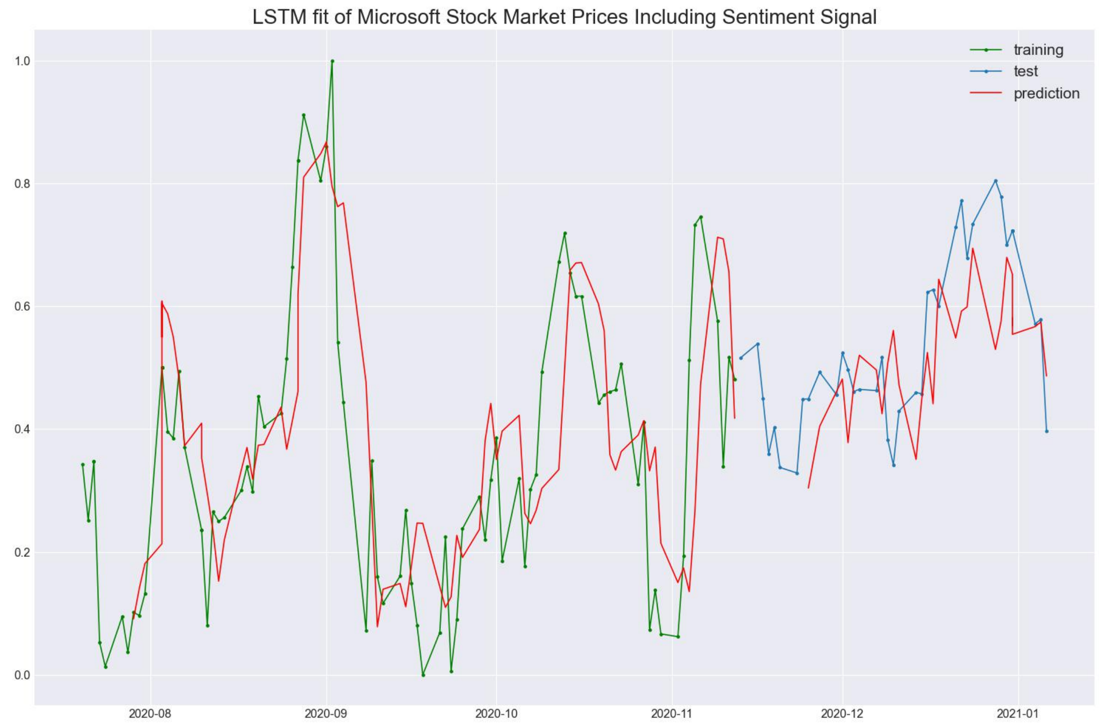
Train Root Mean Squared Error: 0.13535567

Test Mean Absolute Error: 0.07605335

Test Root Mean Squared Error: 0.09149347

3. Add sentiment signals

result



Evaluation

Train Mean Absolute Error: 0.1070031

Train Root Mean Squared Error: 0.13219422

Test Mean Absolute Error: 0.077997595

Test Root Mean Squared Error: 0.095276415