RAKATHAON 2022

ANOMALY DETECTION USING LSTM AUTOENCODER

TEAM : GEEKYSLOTHS

Overview

Finding occurrences and observations that differ from a dataset's typical pattern is known as anomaly detection.

People spend excruciating efforts hoping to detect them in advance and reduce even the slightest impact upon both businesses and users. Since classifying these events is very challenging, we use a deep learning model, namely the LSTM Autoencoder.

What is LSTM Autoencoder?

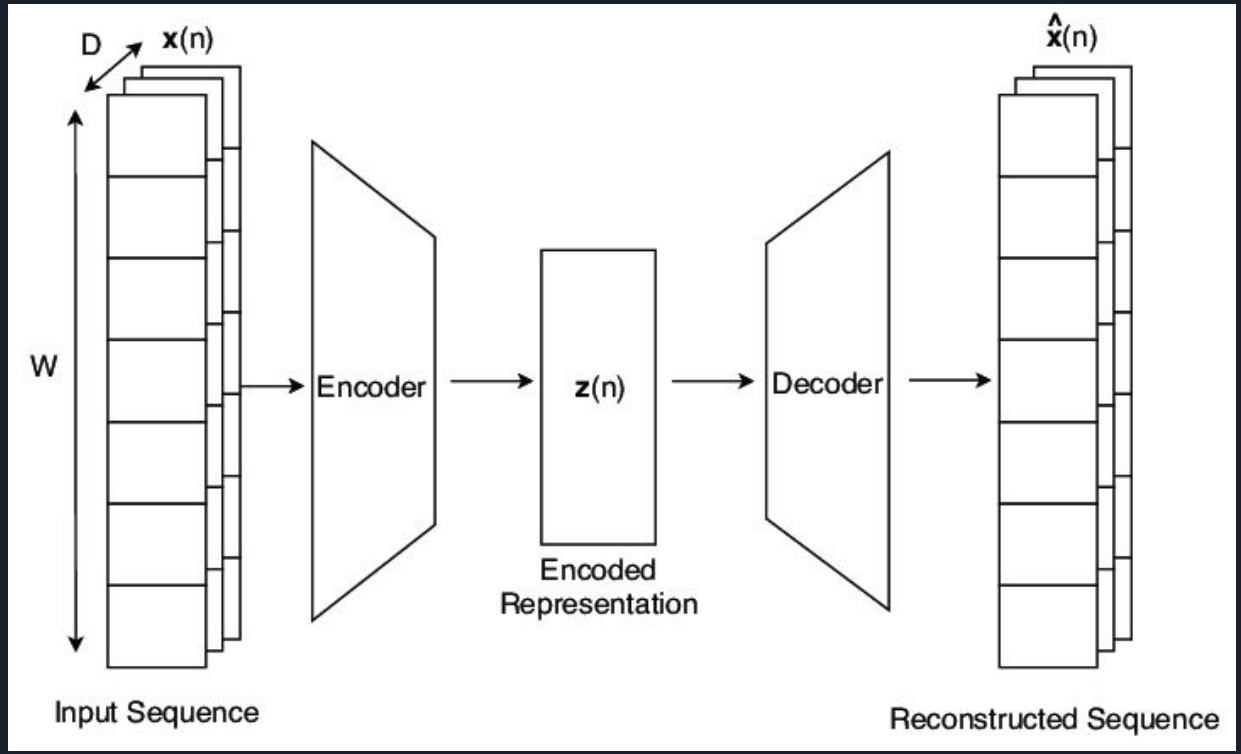
An LSTM Autoencoder is an implementation of an autoencoder for sequence data using an Encoder-Decoder LSTM architecture.

An encoder learns a vector representation of the input time-series and the decoder uses this representation to reconstruct the time-series. The LSTM-based encoder-decoder is trained to reconstruct instances of ‘normal’ time.

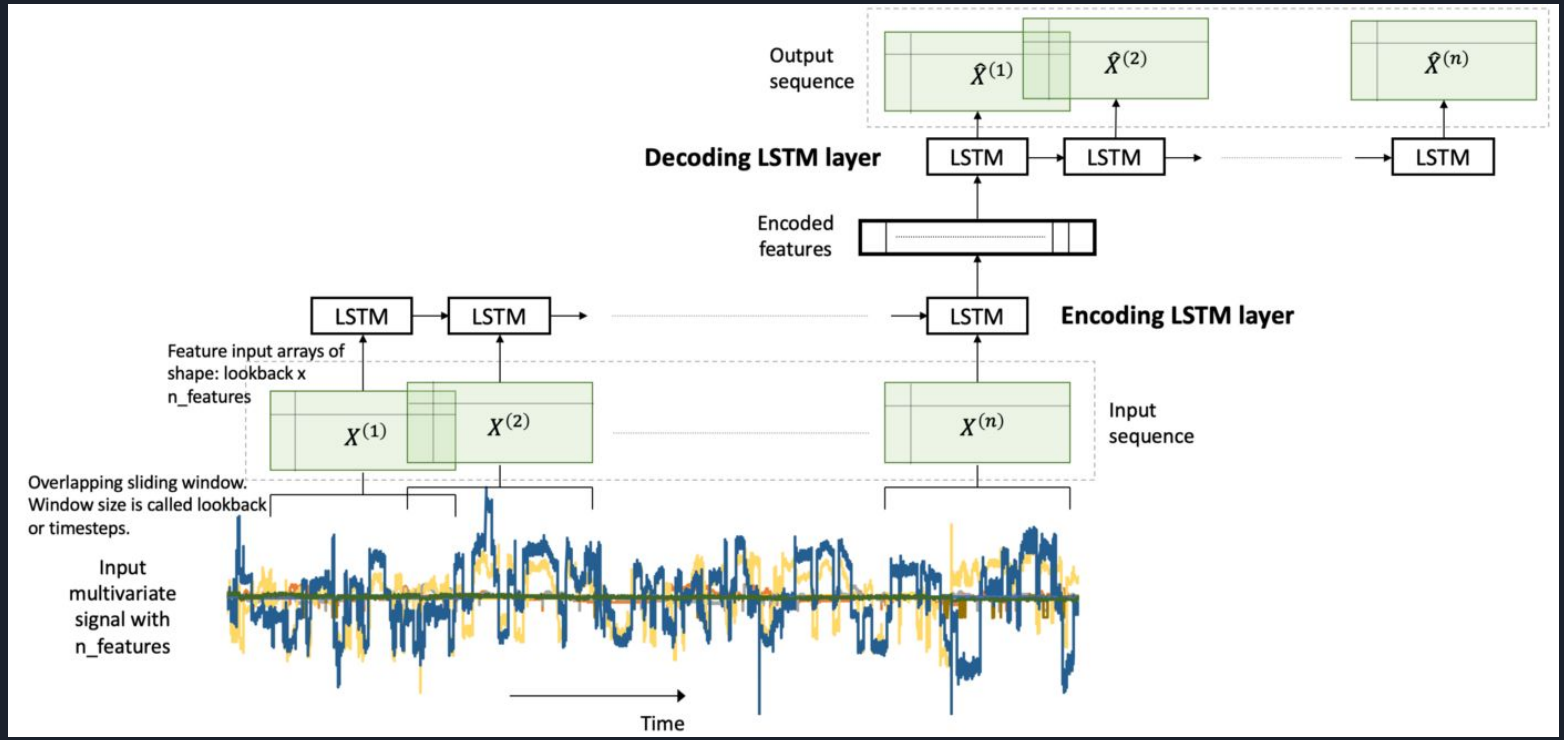
When given an anomalous sequence, it may not be able to reconstruct it well, and hence would lead to higher reconstruction errors compared to the reconstruction errors for the normal sequences.

It has the ability to extract both long and short-term effects of past event.

LSTM Autoencoder



LSTM Autoencoder



Dataset

We use the **NASDAQ Composite (^IXIC) GIDS** real time price currency in USD, our dataset being taken between 2017 to 2022 collected from Yahoo Finance.

Our dataset is a multivariate time-series dataset with 4 variables.

Since LSTM uses sigmoid and tanh functions that are sensitive to magnitude, we first normalize the data using the **StandardScalar()** function for **preprocessing**.

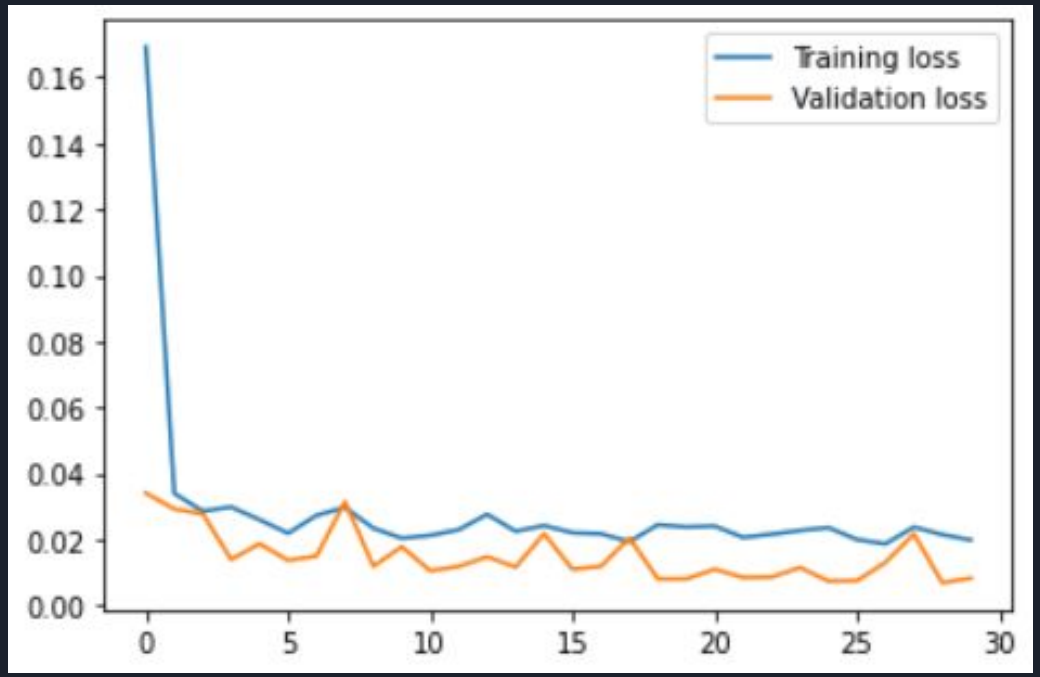
We then reformat the data into a shape of **(num_samples x timesteps x n_features)** numpy arrays that can then be used to fit to our Autoencoder model.

Model architecture

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 5, 64)             17664

 lstm_1 (LSTM)               (None, 32)                12416

 dropout (Dropout)           (None, 32)                0

 dense (Dense)               (None, 1)                 33

=================================================================
Total params: 30,113
Trainable params: 30,113
Non-trainable params: 0
_________________________________________________________________
```

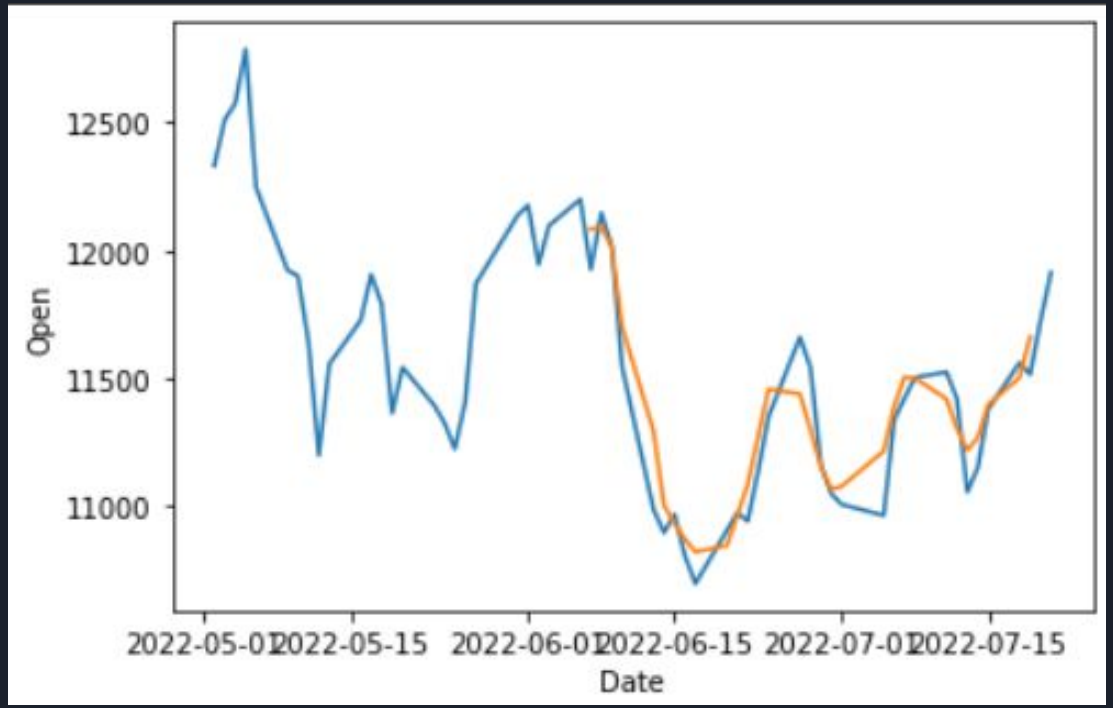
Training and validation loss

The model was trained for 30 epochs with a validation set split of 0.1, and batch size 16



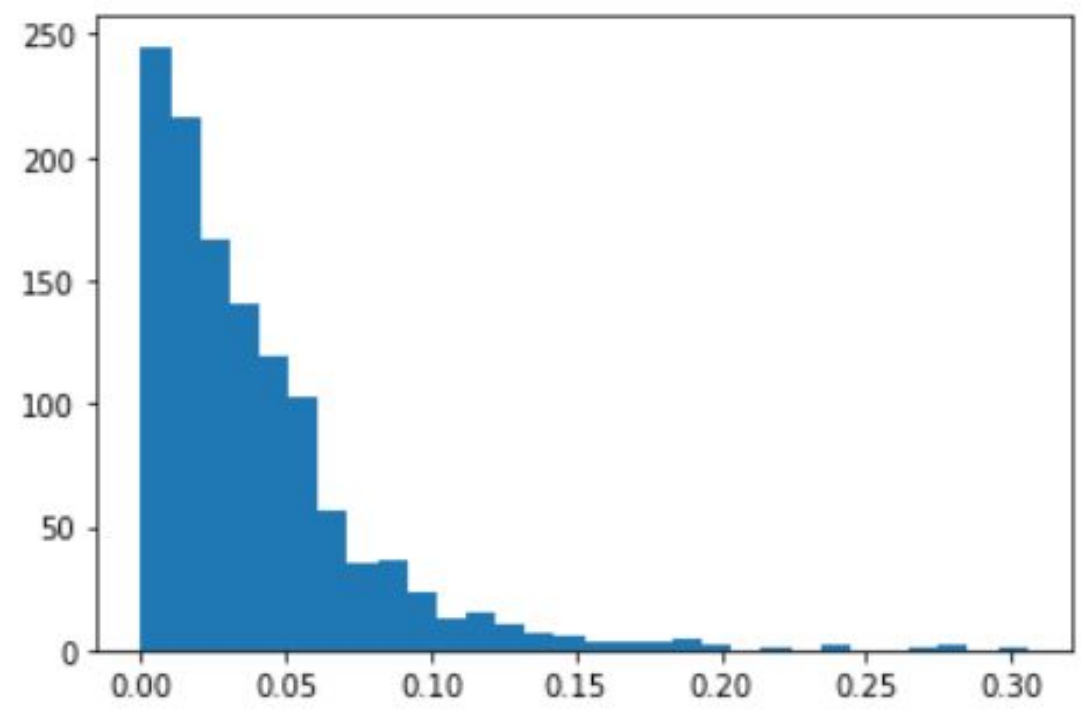
Prediction Plot

We plot our predictions for the last 30 days over the plot of original data,
Using the model trained over the data excluding the last 30 days (meant for predictions)



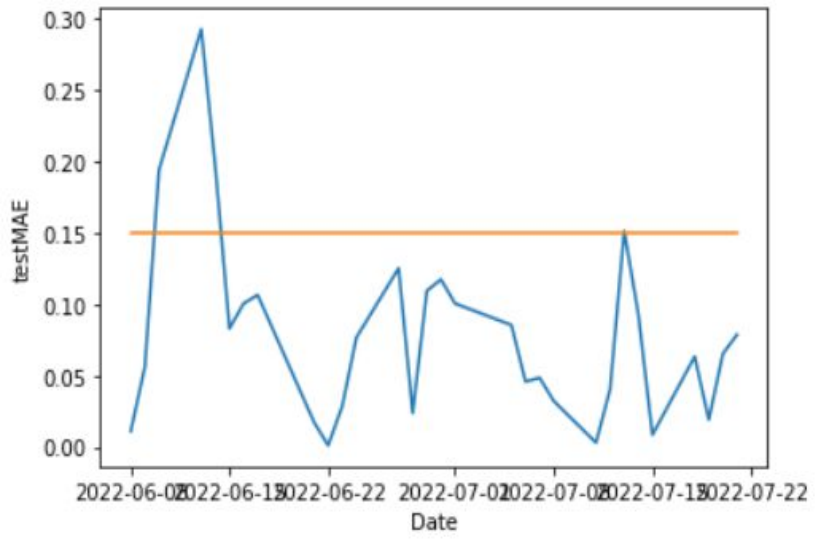
Anomaly detection

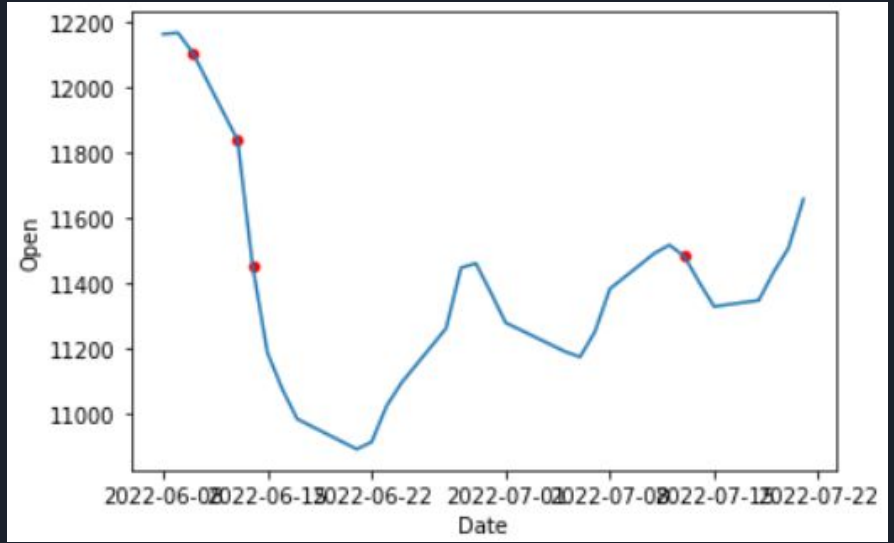
To decide the value of the threshold we first plot a histogram and plot the **mean absolute error** values between the prediction and original data of the 5 years, with observation we have decided to set a **threshold of 0.15**



Anomaly Detection

Once the threshold is set we can plot our anomaly predictions on our test set of last 30 days.
A total of 4 anomalies were detected on the course of 30 days (the threshold can further be decreased to detect more anomalies if deemed necessary)





THANK YOU