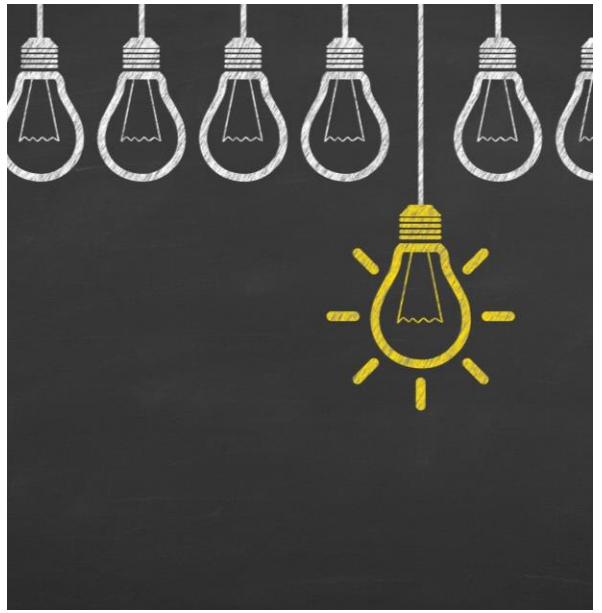
Anomaly Detection in Stocks

Machine Learning and Predictive Analytics

Deep Learning Class Project

Roselyn Rozario

Table of Contents



Problem Statement

Stock trading is a highly volatile and lucrative business, that can be influenced by anomalies. These anomalies are caused by factors (i.e., real-life events, the economy, etc.), which in turn can have impacts around the world. Therefore, understanding anomalies can prove to be beneficial in understanding the market, detecting any illicit activities and more.

In the real-world, anomalies can go undetected to the naked eye, often resulting in impacts within the real-world. This is a problem that machine learning can address.

Stocks trading is an activity that many people participate in all over the world. An activity that companies rely given that their issuance of stocks is what raises capital. On the other hand, stock owners stand to profit from their gains.

Stock trading is highly volatile and extremely lucrative. There are instances when anomalies can occur due to external factors, such as real-life events or economic activities (i.e., rate changes, etc.) that can result in anomalies, or because of bad actors trying to manipulate the market.

Regardless of the cause of anomaly, understanding how they came about can help with understanding market behavior and detecting whether any illicit activities are occurring.

Overview of Dataset and Model

- A dataset with 5 years' worth of data, spanning from 2013 to 2018, will be used to examine stocks of S&P 500 companies.*
- The dataset is courtesy of Kaggle. The dataset source explains to have procured the stock data via The Investor's Exchange API.
- A deep learning model, specifically a reconstruction convolutional autoencoder model, is utilized to detect anomalies within this dataset, which leverages the Keras' API.**
    - Autoencoders are typically useful in detecting anomalies given that one can compare involves comparing the output with the input, after reconstruction. For cases where the output varies from the input, that is, where the reconstruction error is high, it means there is an anomaly, given that the model has a harder time in reconstructing an input that is different to the typical data that it saw during training.

*Dataset: https://www.kaggle.com/datasets/camnugent/sandp500/data

**Keras Example: https://keras.io/examples/timeseries/timeseries_anomaly_detection/

Assumptions/Hypotheses About Data and Model

One limitation with using stock data is that is has autocorrelation due to stock prices typically having correlation with preceding prices, which can affect overfitting and the model's performance.

Another limitation is that if the model is trained on seeing certain patterns, if there are new types of anomalies that arise, unlike what it has seen before, it can struggle to detect these new anomalies.

The data will likely have to undergo feature engineering, pre-processing, and any other preparations to ensure any factors, such as missing data, does not negatively influence the results.

Given that external factors can result in market anomalies, if the model finds any anomalies, those external factors will have to be considered to determine whether they inform on the cause of any anomalies that arise.

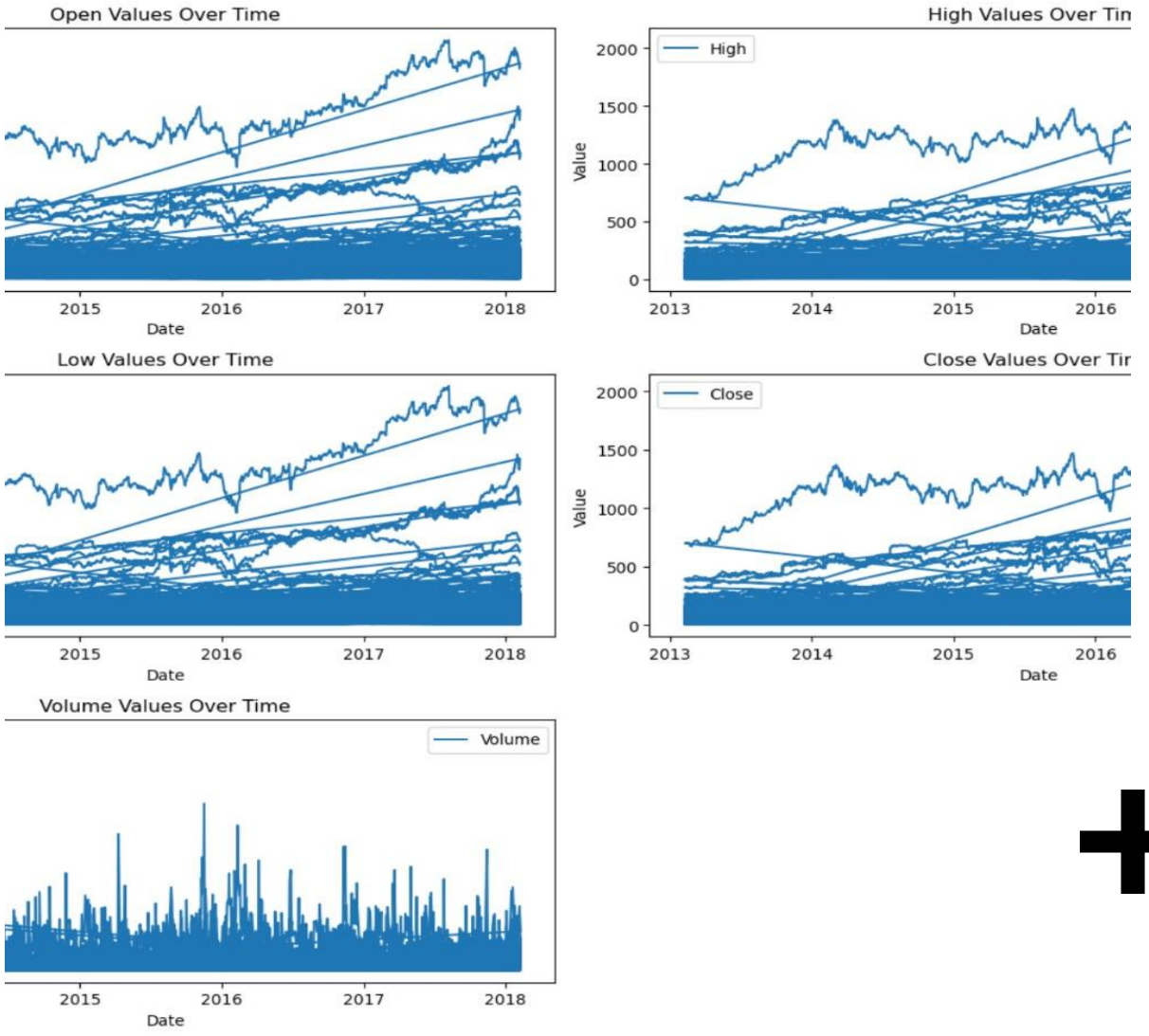
The convolutional autoencoder is expected to play a crucial role in detecting anomalies in that if the reconstruction error is high or put alternatively, if the output is different from the inputs, the model will report having found an anomaly. These differences would be patterns that the model did not learn during training.

Exploratory Data Analysis (EDA)*

- The Exploratory Data Analysis was conducted via a few approaches, which included steps that went beyond the typical explorations as part of this process.
- This section began with pre-processing the data (i.e., date column, etc.), checking for null values, viewing the statistics, and finally culminating in the part that is most relevant for this section, that is, using visuals to examine the data.
- The visuals demonstrated the patterns in data of the various features (i.e., "Open", "Close", "Volume", etc.) over the time-period of the dataset, whereas a deeper dive into the stocks occurred, with an examination of with each of the metrics (features) were plotted for each stock.

**See HTML for full EDA.*

Example of EDA – Stock Metrics Over Time



Feature Engineering & Transformation(s)*

Feature engineering and transformations play a role in the data and model.

- Feature Engineering: Two features were created that examined the spread between the high and low stock values (i.e., "High_Low_Spread") and the (i.e., "Open_Close_Change").
- Transformation(s): The selected features underwent a transformation in that they were standardized or rather, scaled.

The feature engineering and transformations were kept to a minimum so that the original features were mainly used.

- The rationale for this was to discourage overfitting by introducing too much noise via added features and to not add to the complexity that already exists from the stock data being noisy.

**See HTML for Feature Engineering and Transformation(s).*

Proposed Approaches (Model)

Types of Models

- There are various deep learning techniques that can be used to detect anomalies in stock data, including using autoencoders, RNNs and LSTMs.
- The main approach lied in creating a reconstruction convolutional autoencoder model because of it is effective with data that has a time component, such as with historical stock data.

Varying Approaches*

- The first model served as a "base" that includes an input convolutional, dropout, and convolutional transpose layers, as well as a loss function.
  - This model has an encoder and decoder that helps with reconstruction to determine how high the error is, which informs on whether there is an anomaly.
  - The convolutional layer is meant to focus on the features so that it can learn patterns.
  - Regularization is incorporated in the form of dropouts, to discourage overfitting and improve the model's ability to generalize.
- The second model is a culmination of several versions of trying to improve the base model.
  - The improvements predominantly factored in checking for overfitting/underfitting.

Overfitting/Underfitting Checks

- The first overfitting/underfitting check occurred after the model was trained, where the validation and training losses were evaluated for every version of the models that were ran.
  - This first check was the main factor in deciding which model to use given that early signs of overfitting/underfitting during training only meant that further checks, during later parts of the process, would see similar results.
- The second check involved using visuals to convey how the model was reconstructing each of the features, which informed on the model's performance and whether any considerations were needed when analyzing the results.

Addressing Overfitting/Underfitting

- The different components were adjusted to create the best version of an improved model that outperformed the base model.
  - The model's complexity was changed by tweaking the number of filters and layers used.
  - For regularization, dropout layers with varying dropout rates were experimented with, and L2 regularization was incorporated within the Conv1D layers.
  - The learning rate and batch size were also tweaked to determine what training process suited the model.
  - Early stopping and hyperparameter tuning (i.e., kernel size, etc.) were also factored in to improve the model.

**See HTML for the respective models.*

Proposed Solution (Model Selection)

- The selection criteria for the model selected mostly focused on whether the model was overfitting or underfitting during training.
- In the end, the second model is chosen because its training performance and handling of overfitting is better.
  - The second model experiences a validation and training losses that consistently reduce, and shows signs that the model's generalization abilities outperform the base model's. This is especially important, as a model's ability to generalize can help it better detect anomalies and not be "used to" to the dataset that it was trained upon.

Both models experience large training losses initially, before they somewhat stagnate. The difference in how overfitting is handled between these two models is better shown in the respective validation losses. For the second model, the decrease in validation loss is more consistent with each epoch, thus signifying the model better generalizes compared the first model.

Results (Accuracy)

Analysis/Findings

- To determine anomalies, the model set a threshold using the training MAE loss to compare against the test MAE loss. Per this test, no anomalies were detected by the model.
- There are a few analyses that can be made about this finding:
    - The EDA shows that between 2013 – 2018, the stocks saw values of all degrees. The model has no understanding of time, nor is it aware of the different stocks, which is why being trained on the whole dataset as one entity may have resulted in a higher tolerance for what an anomaly is, which may affect the findings.
    - From a wholistic perspective, the model may not have found anomalies, but from a more granular point-of-view, the model might have had different results had it had additional knowledge, such as from incorporating the information about the different stocks and the temporal features.

Learnings From the Methodology

There was a lot learned from the methodology used:

- In terms of the features, when time is included within the data, temporal features should be used in the model, in addition to any information that can help it understand patterns on a more granular level.
- As for the model itself, there are a few things to make a note of. For one, the layers, parameters and other components need to be tailored according to the data, so that it does not overfit or underfit and can perform well. The methods to use to incorporate regularization, address overfitting/underfitting, and creating a model structure or rather, architecture, that suits the type of complexity that is needed were all learned.
- It was also learned that additional actions could have been taken for the model to be able to reconstruct negative values, such as through the types of activation functions that are used.

Model Improvements

The model should incorporate temporal features and the "Name" column so that it can distinguish between stocks (i.e., via embeddings or some other method).

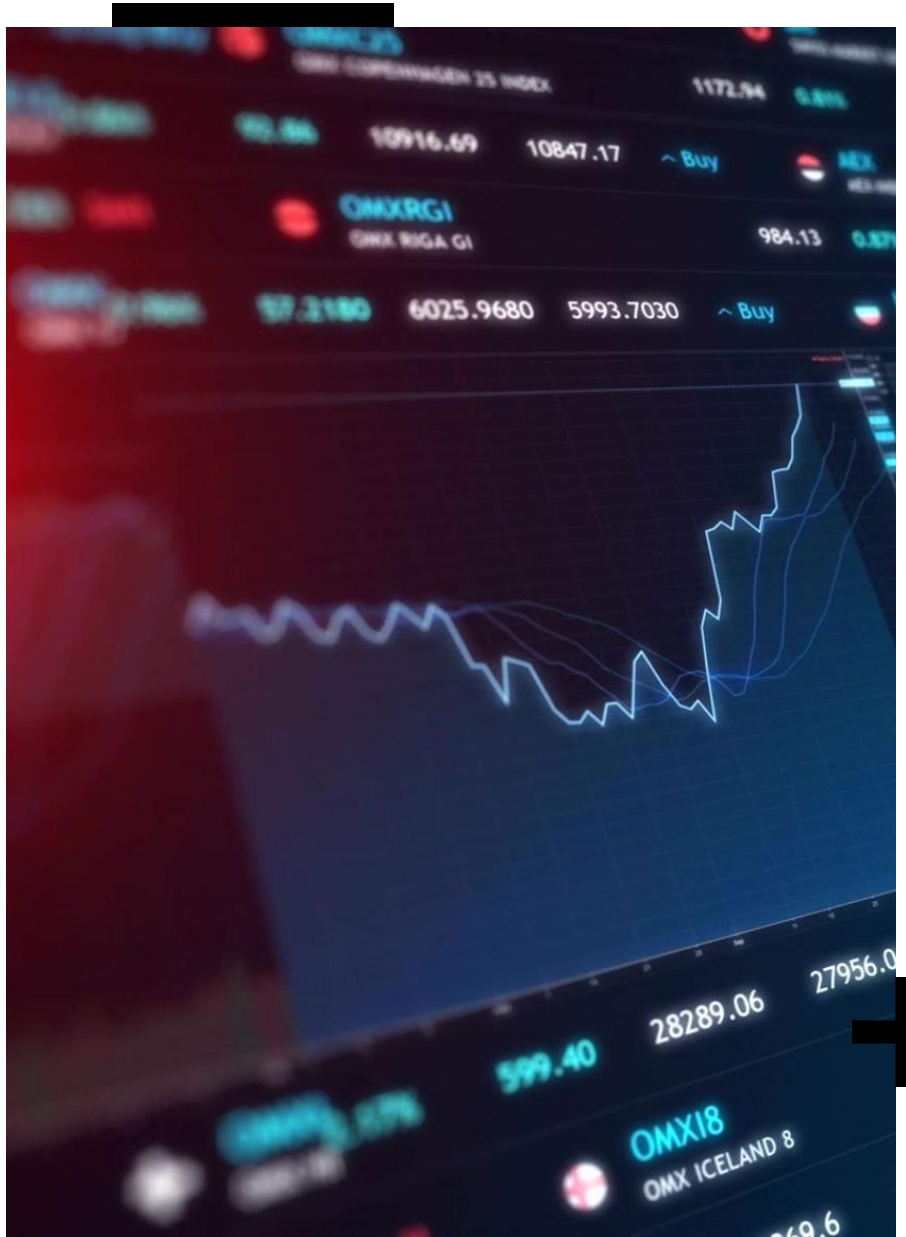
The model architecture should undergo a few changes. For starters, the layers can have tweaks made to them, components such as LSTM can be incorporated, the hyperparameters can be further tuned, and the activation functions can be set to account or allow for negative values. These changes should address some of the model's limitations, such as being unable to reconstruct negative values and address any overfitting.

The data can be better prepped, through augmentation, though this would have to be done carefully, adding additional feature, or by including any other additional information that can help the model better learn the data.

Future Work

- Any future work for evaluating anomalies within this dataset can involve testing out some of the ideas for how to improve the model.
- From an overall perspective, there are a few things to consider.
  - There is no standard for what an anomaly is, nor is there one for a model that can account for every scenario, which makes it challenging to detect anomalies (Raza, 2023).
  - AI can also mitigate many of the challenges that lie with anomaly detections, which can be incorporated in future works given that it is highly intelligent and probably can be trained better than a model can (Raza, 2023).

"Anomaly Detection in 2024: Opportunities & Challenges": https://www.splunk.com/en_us/blog/learn/anomaly-detection.html



References

- Pavithra, V. (2020, May 31). *Timeseries anomaly detection using an Autoencoder*. Keras. https://keras.io/examples/timeseries/timeseries_anomaly_detection/
- Raza, M. (2023, October 3). *Anomaly Detection in 2024: Opportunities & Challenges*. Splunk. https://www.splunk.com/en_us/blog/learn/anomaly-detection.html
- *S&P 500 Stock Data*. Kaggle. (n.d.). https://www.kaggle.com/datasets/camnugent/sandp500?select=all_stocks_5yr.csv