Configuration Manual

MSc Research Project
Programme Name

Swapnil Pawar
Student ID: x23202530

School of Computing
National College of Ireland

Supervisor: Dr. Anu Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Swapnil Pawar
Student ID:	23202530
Programme:	MSc Data Analytics **Year:** 2024-25
Module:	Research Project
Lecturer:	Dr. Anu Sahni
Submission Due Date:	24/04/2025
Project Title:	Advanced Strategies for Enhancing Tesla Stock Price Prediction
Word Count:799	**Page Count:** 7

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: SP

Date: 24/04/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Swapnil Pawar
Student ID: x23202530

1 Hardware Requirement

The system used for development was a personal computer with the following specifications:

- Processor: Intel Core i5
- RAM: 8 GB
- Operating System: Windows 10
- GPU: Integrated Intel Iris Xe Graphics (local execution)

2 Software Requirement

The research was conducted using Jupyter Notebook on Anaconda (Fig 2.1). Python 3.x was the primary programming language. The following libraries and tools (Fig 2.2) were used during development:

- pandas, numpy, matplotlib, seaborn
- sklearn (MinMaxScaler, train_test_split, evaluation metrics)
- tensorflow and keras (Conv1D, LSTM, Bidirectional, Dense, Dropout)
- vaderSentiment (VADER)
- shap and lime for explainability

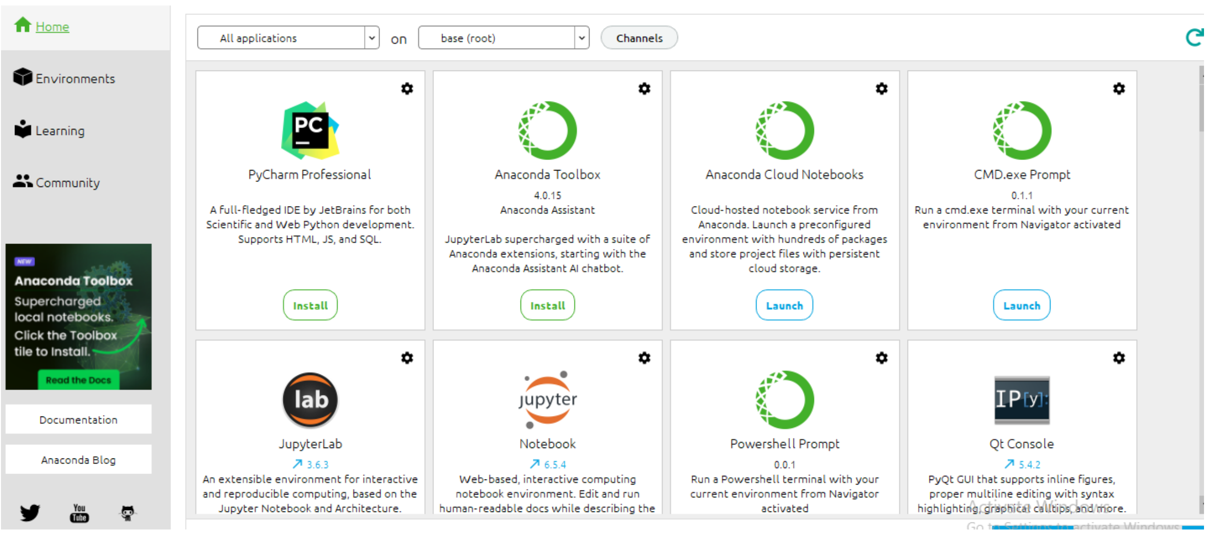

Fig 2.1: Software Requirement

```python
import pandas as pd
import numpy as np
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, LSTM, Bidirectional, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from sklearn.metrics import mean_squared_error, mean_absolute_error
import matplotlib.pyplot as plt
import shap
import lime
import lime.lime_tabular
import datetime
import warnings
warnings.filterwarnings("ignore")
```

Fig 2.2: Libraries and Tools

3 Implementation Overview

The implementation was carried out in a single Jupyter notebook. Data from Kaggle, Twitter, and Reddit were preprocessed and merged using pandas. Sentiment scores were extracted using VADER and aggregated daily. Stock data was normalized using MinMaxScaler and transformed into 15-day input sequences to predict the next day's closing price. SHAP and LIME were used to interpret model predictions.

4 Dataset Description

- Stock Data: Tesla stock price data obtained from Kaggle. (fig: 4.1)
- Twitter Data: Pre-obtained dataset collected via Twitter API containing tweets about Tesla. (fig:4.2)
- Reddit Data: Posts collected from subreddits like r/TSLA, r/WallStreetBets, and r/investing.(fig:4.3)

- Sentiment scores were computed using VADER and grouped by daily averages.

	Date	Close	High	Low	Open	Volume
0	2019-01-02	20.674667	21.008667	19.920000	20.406668	174879000.0
1	2019-01-03	20.024000	20.626667	19.825333	20.466667	104478000.0
2	2019-01-04	21.179333	21.200001	20.181999	20.400000	110911500.0
3	2019-01-07	22.330668	22.449333	21.183332	21.448000	113268000.0
4	2019-01-08	22.356667	22.934000	21.801332	22.797333	105127500.0

Fig 4.1: Tesla stock price

```
        Date                                              Tweet
0 2022-09-30  The most powerful word in my English vocabular...
1 2022-09-30  Tesla China Says Rumor of Model Y Price Cut Is...
2 2022-09-30  +$10,460 day trading $TSLA $SPYNew video / rec...
3 2022-09-30  The strong push for quarter end is happening. ...
4 2022-09-30  Tesla team is awesome, such an honor to work w...
```

Fig 4.2: Twitter API containing tweets

```
                 Date                                               text
0 2022-04-06 09:14:16  Wash sale Hi guys I bought uvxy beginning of t...
1 2022-04-04 22:17:42  Let's say you made 1 Million dollars and had t...
2 2022-04-02 00:00:07  R/place I know you degenerates aren't trading ...
3 2022-03-31 22:12:12  AST SpaceMobile inc. BlueWalker 3 satellite la...
4 2022-03-31 22:09:13  Recent trading performance for those who are i...
```

Fig 4.3: Posts collected from subreddits

5 Data Preprocessing

- Merged stock and sentiment data on common date field.(Fig 5.1)

	Date	Close	High	Low	Open	Volume	twitter_sentiment	reddit_sentiment
944	2022-09-30	265.250000	275.570007	262.470001	266.149994	67726600.0	0.173933	0.0
945	2022-10-03	242.399994	255.160004	241.009995	254.500000	98363500.0	0.043345	0.0
946	2022-10-04	249.440002	257.500000	242.009995	250.520004	109578500.0	0.125210	0.0
947	2022-10-05	240.809998	246.669998	233.270004	245.009995	86982700.0	0.136154	0.0
948	2022-10-06	238.130005	244.580002	235.350006	239.440002	69298400.0	0.147633	0.0

Fig 5.1: Merged Data

- Filled missing sentiment values with 0. (Fig:5.2)

```
#  Merge all
df = df_stock.merge(twitter_daily, on='Date', how='left')
df = df.merge(reddit_daily, on='Date', how='left')
df[['twitter_sentiment', 'reddit_sentiment']] = df[['twitter_sentiment', 'reddit_sentiment']].fillna(0)
```

Fig 5.2

- Normalized all features using MinMaxScaler. (Fig:5.3)

```
# === Scaling ===
scaler = MinMaxScaler()
df_scaled = df.copy()
df_scaled[features] = scaler.fit_transform(df_scaled[features])
```

Fig:5.3: MinMaxScaler

- Converted data to supervised time-series format using a 15-day sliding window.
- Twitter sentiment had more variance and clarity; Reddit sentiment was sparse and mostly neutral.

6 Model Implementation

- Model: Hybrid deep learning architecture using Conv1D + Bi-LSTM + Transformer layers.(Fig 6.1)
- Activation: ReLU for hidden layers, Linear for output.
- Loss Function: Mean Squared Error
- Optimizer: Adam
- Metrics: MAE, RMSE
- The best configuration was selected using Bayesian Optimization over 5 trials with 5 epochs each.(Fig 6.2)

```
def build_combined_model(hp):
    cnn_filters = hp.Int('cnn_filters', min_value=32, max_value=128, step=32)
    lstm_units = hp.Int('lstm_units', min_value=32, max_value=128, step=32)
    attention_heads = hp.Int('attention_heads', min_value=2, max_value=8, step=2)

    input_layer = Input(shape=(seq_len, len(features)))

    # CNN-BiLSTM Branch
    x1 = Conv1D(filters=cnn_filters, kernel_size=2, activation='relu')(input_layer)
    x1 = Bidirectional(LSTM(lstm_units, return_sequences=True))(x1)
    x1 = GlobalAveragePooling1D()(x1)

    # Transformer Branch
    x2 = LayerNormalization(epsilon=1e-6)(input_layer)
    x2 = MultiHeadAttention(num_heads=attention_heads, key_dim=64)(x2, x2)
    x2 = GlobalAveragePooling1D()(x2)

    merged = Concatenate()([x1, x2])
    merged = Dense(64, activation='relu')(merged)
    merged = Dropout(0.2)(merged)
    output = Dense(1)(merged)

    model = tf.keras.Model(inputs=input_layer, outputs=output)
    model.compile(optimizer=Adam(0.001), loss='mse')
    return model
```

Fig 6.1: Combined Model

```
bayes_tuner.search(X_train, y_train, epochs=5, validation_data=(X_test, y_test))
best_model = bayes_tuner.get_best_models(1)[0]
```

```
Trial 5 Complete [00h 00m 18s]
val_loss: 0.0035380320623517036

Best val_loss So Far: 0.0032350802794098854
Total elapsed time: 00h 01m 43s
```

Fig 6.2: Bayesian Optimization

7 Explainability Frameworks

SHAP and LIME were used to interpret the model's predictions. SHAP's default deep learning-compatible Explainer was used to visualize global feature importance, while LIME's LimeTabularExplainer was applied to provide local interpretability of individual predictions.

```
# === SHAP Explainability: Combined Model ===
def combined_predict(x):
    return best_model.predict(x.reshape((x.shape[0], seq_len, len(features))))

explainer_c = shap.KernelExplainer(combined_predict, X_train_flat[:100])
shap_values_c = explainer_c.shap_values(X_test_flat[:1])
shap.plots.waterfall(shap.Explanation(
    values=shap_values_c[0].flatten(),  # flatten to 1D
    base_values=explainer_c.expected_value[0],  # get scalar
    data=X_test_flat[0],
    feature_names=[f'{f}_{i}' for i in range(seq_len) for f in features]
))
```

Fig 7.1: SHAP Explainability

```
# === LIME Explainability: Combined Model ===
lime_explainer_c = lime.lime_tabular.LimeTabularExplainer(
    X_train_flat,
    feature_names=[f'{f}_{i}' for i in range(seq_len) for f in features],
    mode='regression',  # Important!
    verbose=True
)
exp_c = lime_explainer_c.explain_instance(X_test_flat[0], combined_predict, num_features=10)
exp_c.show_in_notebook()
```

Fig 7.2: Lime Explainability

References

Chollet, F. (2015). Keras. Available at: https://keras.io

Hutto, C.J. and Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. In: Proceedings of the Eighth International Conference on Weblogs and Social Media (ICWSM-14), pp. 216–225.

Lundberg, S.M. and Lee, S.I. (2017). A unified approach to interpreting model predictions. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), pp. 4765–4774.

Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, pp.2825–2830.

Ribeiro, M.T., Singh, S. and Guestrin, C. (2016). “Why should I trust you?”: Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144.

Twitter Developer Platform. (n.d.). Twitter API Documentation. Available at: https://developer.twitter.com/en/docs