

Configuration Manual

MSc Research Practicum
M.Sc. in FinTech Information

Naga Lakshmi Divya Romala
Student ID: X23320133

School of Computing
National College of Ireland

Supervisor: Victor del Rosal

National College of Ireland
MSc Project Submission Sheet



School of Computing

Student Name: Romala Naga Lakshmi Divya
Student ID: X23320133
Programme: M.Sc. in FinTech Information **Year:** 2024-2025
Module: Practicum
Lecturer: Victor del Rosal
Submission Due Date: 15/09/2025
Project Title: Multi-Horizon Stock Price Volatility Forecasting Using Attention-Based Transformers
Word Count: 1003 **Page Count:** 6

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:Romala Naga Lakshmi Divya.....

Date:15/09/2025.....

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

Naga Lakshmi Divya Romala
Student ID: X23320133

This project implements an attention-based transformer model for multi-horizon financial volatility forecasting by integrating news sentiment analysis with market indicators. The system processes financial news data, performs ensemble sentiment analysis, creates market-derived labels using triple barrier method, and trains a transformer model to predict volatility at 1-day, 5-day, and 15-day horizons with comprehensive market regime and sector-specific analysis.

1 Google Colab Setup Guide - 5 Essential Steps

1.1 Step 1: Upload Notebook to Google Colab

1. Go to colab.research.google.com
2. Click **File** → **Upload notebook**
3. Upload
https://colab.research.google.com/drive/10SZuqvIa_8yivELjZ5RAAn9UISXVgaX8#scrollTo=CzoajmVj_iz7 file
4. Wait for upload to complete

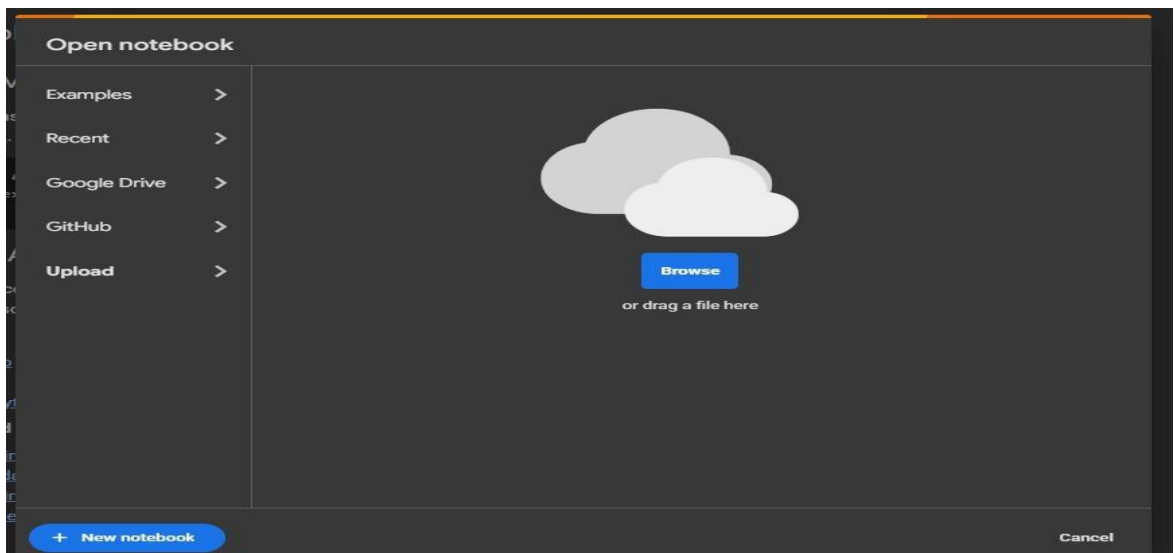


Figure 1 Upload the notebook

1.2 Step 2: Connect to T4 GPU

CRITICAL - Must do before running any cells:

1. Click **Runtime** → **Change runtime type**
2. Set:
 - Hardware accelerator: **GPU**

- GPU type: T4
- 3. Click **Save**
- 4. Click **Connect** button (top right)
- 5. Verify GPU: Run this test cell:

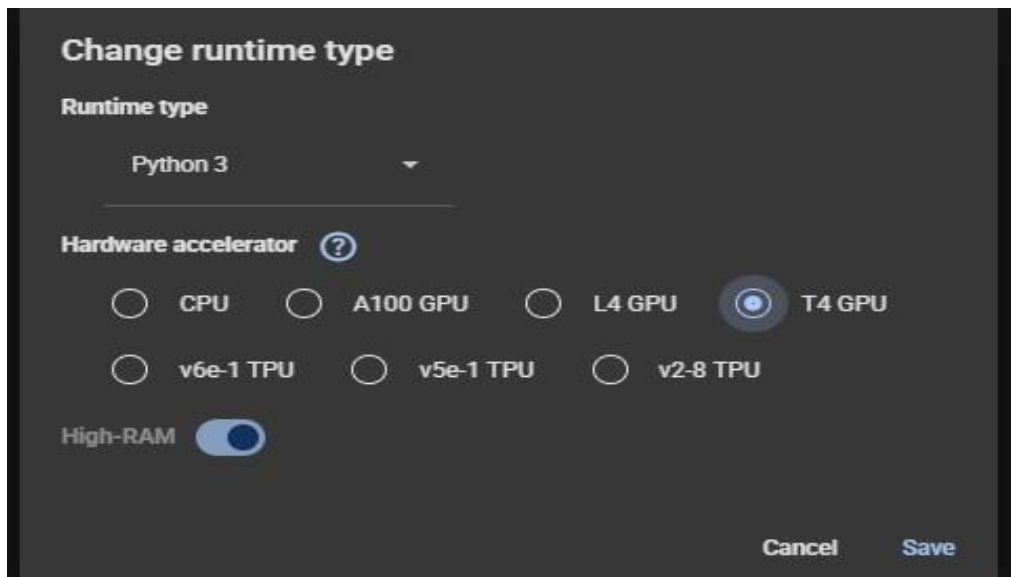


Figure 2 Connect to GPU

```
import torch
print(f'GPU Available: {torch.cuda.is_available()}')
print(f'GPU Name: {torch.cuda.get_device_name(0) if torch.cuda.is_available() else 'No GPU'}')
```

1.3 Install Dependencies & Mount Drive

Run the first few cells in the notebook that contain:

Dependencies installation cell

```
!pip install torch torchvision transformers datasets -q
!pip install pandas numpy matplotlib seaborn scikit-learn -q
!pip install ta textblob vaderSentiment tqdm -q
```

Google Drive mounting cell

```
from google.colab import drive
drive.mount('/content/drive')
```

1.4 Step 4: Configure Settings Before Running

Check these configuration points in the notebook:

1.4.1 Data Processing Configuration (Early in notebook):

```
▼ Data Preparation
1
2
3 import subprocess
4 import sys
5 import pandas as pd
6 import numpy as np
7 import matplotlib.pyplot as plt
8 import seaborn as sns
9 from datetime import datetime, timedelta
10 import re
11 import warnings
12 from datasets import load_dataset
13 from tqdm import tqdm
14 import os
15 import gc
16 import torch
17 from transformers import AutoTokenizer, AutoModelForSequenceClassification, pipeline
18 import ta
19 from sklearn.metrics.pairwise import cosine_similarity
20 from sklearn.feature_extraction.text import TfidfVectorizer
21 from sklearn.ensemble import VotingClassifier
22 from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
23 from sklearn.utils.class_weight import compute_class_weight
24 from scipy import stats
25 from scipy.stats import zscore
26 from textblob import TextBlob
27 from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
28 import logging
```

Figure 3 Data preparation part

Look for this section and adjust:

USE_SAMPLE = False # <- Set True for quick test (1000 rows), False for full dataset

SAMPLE_SIZE = 1000 # <- Only used if USE_SAMPLE = True

1.5 Model Training Configuration

Find the config dictionary and adjust if needed:

```
config = {
    'data_path': 'enhanced_financial_dataset_truly_fixed.csv',
    'batch_size': 32, # <- Reduce to 16 if GPU memory error
    'num_epochs': 20, # <- Reduce to 5 for quick testing
    'learning_rate': 1e-4,
    'device': 'cuda' if torch.cuda.is_available() else 'cpu'
}
```

```
1489
1490 def main():
1491     """Main execution function"""
1492
1493     # Configuration
1494     config = {
1495         'data_path': 'enhanced_financial_dataset_truly_fixed.csv',
1496         'batch_size': 32,
1497         'embed_dim': 256,
1498         'num_heads': 8,
1499         'num_epochs': 20,
1500         'learning_rate': 1e-4,
1501         'device': 'cuda' if torch.cuda.is_available() else 'cpu'
1502     }
1503
1504     logger.info(f" Configuration: {config}")
1505
1506     # Load and prepare data
1507     df, sentiment_cols, market_cols, target_cols, text_cols = prepare_data(config['data_path'])
1508
1509     # Initialize analyzers
1510     regime_analyzer = MarketRegimeAnalyzer()
1511     sector_analyzer = SectorAnalyzer()
1512
```

Figure 4 Configuration

1.6 Step 5: Run the Entire Notebook

1. Option A - Run All Cells:
 - Runtime → Run all
 - Total time: ~2-4 hours for full dataset
2. Option B - Run Section by Section:
 - Section 1: Data Preparation (~60-90 min)
 - Data loading from Hugging Face

- Triple barrier labeling
- Ensemble sentiment analysis
- Volatility target creation
- Saves: enhanced_financial_dataset_truly_fixed.csv
- **Section 2: Modelling** (~60-90 min)
 - Loads prepared dataset
 - Transformer model training
 - Market regime analysis
 - Sector-specific analysis
 - Saves: model weights, results, visualizations

1.7 Expected Output Files

After successful completion, check for these files:

1. **Enhanced_financial_dataset_truly_fixed.csv** (~XXX MB)
 - Processed dataset with all features
2. **Financial_volatility_transformer.pth** (~XX MB)
 - Trained model weights
3. **Comprehensive_training_results.json**
 - All metrics and analysis results
4. **Enhanced_volatility_forecasting_results.png**
 - Multi-panel visualization
5. **Regime_analysis_data.csv**
 - Market regime performance data
6. **Sector_analysis_data.csv**
 - Sector-wise performance data

1.8 Quick Test Configuration

For a 15-minute test run, modify these values:

In data processing section:

USE_SAMPLE = True

SAMPLE_SIZE = 500

In model config:

'num_epochs': 3,

'batch_size': 64 *# Can increase for small sample*

2 Troubleshooting

2.1.1 GPU Memory Error

Find config dictionary and change:

'batch_size': 16, *# or even 8*

2.1.1.1 Runtime Disconnected:

- Save outputs to Drive frequently
- Use checkpointing (model saves after each epoch)

2.1.1.2 Module Import Errors:

python

Re-run the dependency installation cell

```
!pip install transformers==4.30.0 torch==2.0.0
```

2.1.1.3 Results Interpretation

After completion, look for the summary section showing:

- **Overall R^2 scores** for 1d, 5d, 15d predictions
- **Market regime analysis** (Bull/Bear/Sideways performance)
- **Sector-specific performance** rankings
- **Company clustering** results

2.1.1.4 Saving Results to Drive

Add this cell at the end to save all outputs:

```
# Save all results to Drive
```

```
import shutil
```

```
SAVE_DIR = '/content/drive/MyDrive/thesis_results/'
```

```
os.makedirs(SAVE_DIR, exist_ok=True)
```

```
# Copy all output files
```

```
for file in ['*.csv', '*.pth', '*.json', '*.png']:
```

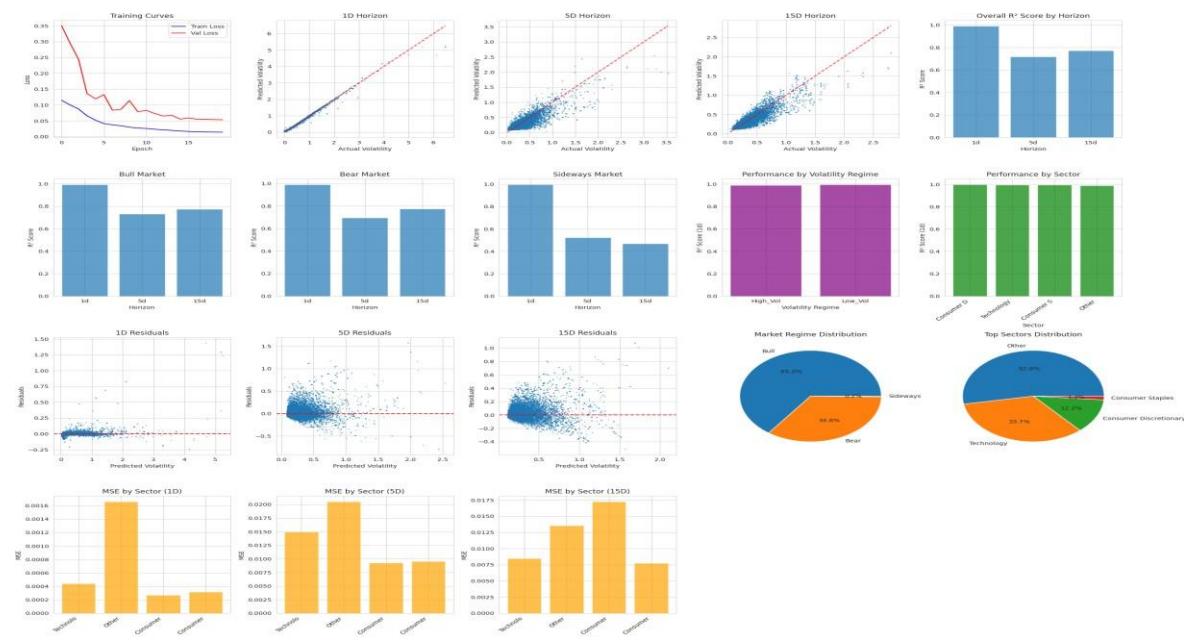
```
    !cp {file} {SAVE_DIR}
```

```
print(f'Results saved to: {SAVE_DIR}')
```

Important Notes:

- The notebook contains two main sections: **Data Preparation** and **Modelling**
- Section 1 creates the enhanced dataset with sentiment and volatility features
- Section 2 trains the transformer model and performs comprehensive analysis
- Total runtime: 2-4 hours on T4 GPU for full dataset
- Monitor GPU usage: Edit → Notebook settings → View resources
- Save intermediate results to avoid losing progress

3 Results



Reference

1. Google Colab. (n.d.). *Colab.google*. colab.google. <https://colab.google/>