COMP40370 Practical 4

ADVANCED BTC PRICE PREDICTION (Part A)

Prof. Tahar KECHADI

Academic year 2025-2026

Assignment Files

- ./Practical-04-A.pdf — Assignment questions (this file).
- ./requirements.txt — Python environment requirements
- ./*BTCUSDT-1m.zip* — Data file for the question.
  - ./training/
  - ./validation/
  - ./test/

Expected output files

- ./Practical-04.ipynb — Python notebook programs.
- ./Practical-04.html — Notebook in HTML showing the outputs.

Overview

This practical is a mandatory, graded assignment that builds on the optional work from Practical 3. The objective is twofold: (Part A) to implement and thoroughly compare a selection of advanced machine learning and deep learning models to enhance the baseline BTC price trend prediction task; (Part B) to present your methodology, results, and findings in a formal academic report.

In this part, you will apply your skills in model selection, implementation, and evaluation. You will use the fixed dataset splits (training, validation, test).

Dataset Description

The dataset contains BTCUSDT (Bitcoin/Tether) 1-minute candlestick data:

Features

`open_time`:	Opening timestamp in milliseconds
`open`:	Opening price
`high`:	Highest price during the interval
`low`:	Lowest price during the interval
`close`:	Closing price
`volume`:	Trading volume
`close_time`:	Closing timestamp in milliseconds

Data Split

Training:	2024 (Jan-Dec)	~527,000 rows
Validation:	2025-01	~44,000 rows
Test:	2025-02 to 2025-04	~128,000 rows

Your Tasks

1. **Model Selection and Training:**

You must select and implement **at least three** distinct classification models. Your selection can include any appropriate machine learning (e.g., advanced ensembles, Support Vector Machines) or deep learning models (e.g., CNN, LSTM, GRU, Transformer-based architectures).

The goal is to explore different model families to achieve the best possible classification performance on the test set.

The method for creating dataset labels must be consistent with Practical 3:

- `1` (`UP`): Future price (30 min ahead) > current price
- `0` (`DOWN`): Future price (30 min ahead) ⩽ current price

```python
def create_target(df, horizon=30):
    """Create target variable for price trend prediction"""
    df = df.copy()

    # Future price after 'horizon' minutes
    df['future_price'] = df['close'].shift(-horizon)

    # Binary classification: 1 if price goes up, 0 if down or same
    df['target'] = (df['future_price'] > df['close']).astype(int)

    return df
```

2. **Custom/Improved Model Requirement:**

Of the three (or more) models you compare, **at least one** must be a model that you have personally modified or improved. Simply using an off-the-shelf model (e.g., sklearn.svm.SVC() with default parameters) does not satisfy this requirement.

Examples of an "improved" model include:

- A custom-designed deep learning architecture (e.g., a novel combination of CNN and LSTM layers).
- A unique ensemble of several different models.
- A standard model combined with a novel feature engineering or selection strategy that you design and justify.

3. **Evaluation:**

You must train your models on the training set and use the validation set to monitor the training and tune the hyperparameters.

The final, definitive performance of all models must be reported on the test set.

Your notebook should contain a clear comparison of all models using appropriate metrics (Accuracy, Precision, Recall, F1-Score) and visualisations (e.g., confusion matrices, performance comparison charts).

4. **Submission:**

All **code**, **outputs**, and **brief textual analysis** should be contained within a single Jupyter Notebook (Practical-04.ipynb).

You must also submit an HTML version of the notebook (Practical-04.html) with all cell outputs visible.

5. **Grading Dimensions:**

In Part A, grading will be based on the following four dimensions:

- **Requirement Complete:** Submission compliance requirements, such as strict adherence to the specified dataset split, completeness of metrics, and whether the number and types of models compared meet the requirements.
- **Correctness**: Review the provided processes, such as data processing, training, metric calculation, and comparison, to ensure they are reasonable and correct.
- **Professionalism**: Review the professionalism of the provided code files, such as whether they are readable, whether necessary intermediate or final steps are printed or visualised, and whether they include the required comments or explanations to clarify the code's meaning or motivation.
- **Innovation**: Evaluate the innovation of the improved model in the submission, whether it demonstrates creativity, and whether the enhancement is significant.

HELPFUL RESOURCES

TensorFlow Documentation: https://www.tensorflow.org/api_docs

PyTorch Documentation: https://pytorch.org/docs/stable/index.html

Scikit-learn Documentation: https://scikit-learn.org/stable/

The final deadline for the submission of Practical 4 (Part A and B) is Sunday, 30th of November at 23:59. Part A submissions should be in a single file with FirstName_LastName-P4.zip (or tar.gz) format. Part B submissions should be in a single PDF file. All submissions must be done in Brightspace.