CSC 180-01 Intelligent Systems (Fall 2025)

Project 4: Time-Series Forecasting using FCNN, CNN and RNN

Due at 10:30 am, Monday, November 17, 2025

Demo: Monday, November 17, 2025

Note that you must **print and fill in** your names on the Evaluation Form and bring the copy to your demo session to receive credit.

1. Problem Formulation

Time series forecasting is an important area of AI. It is important because there are so many prediction problems that involve a time component. This time component makes time series problems more difficult to handle. In this project, you practice with time series data to predict stock price.

This project is threefold:

- Task 1: Predict [Close] of a day based on the last 7 days' data [Open, High, Low, Volume, Close] using a **full-connected neural network model**. In other words, we want to predict the price in the green cell using all the numbers in the red cell. Use the first 70% of the available records for training and the remaining 30% of the available records for test. Report the RMSE of the best model on your test data. Show the "regression lift chart" on your test data.

  Hint: Each record has 7 * 5 = 35 input features and 1 output feature.

- Task 2: Do the same as Task 1 but use a **CNN model**. Report the RMSE of the best model on your test data. Show the "regression lift chart" on your test data.

  Hint: The red cell can be viewed as a 1D image of 7 pixels, each pixel with 5 channels, or as a 2D image of 7 * 5 = 35 pixels, each pixel with 1 channel.

- Task 3: Do the same as Task 1 but use a **LSTM model**. Report the RMSE of the best model on your test data. Show the "regression lift chart" on your test data.

  Hint: Each record can be viewed as a sequence of 7 vectors, each vector with 5 dimensions.

	A	B	C	D	E
1	# http://finance.yahoo.com/quote/GOOG/history?ltr=1				
2	# Open	High	Low	Volume	Close
3	828.66	833.45	828.35	1247700	831.66
4	823.02	828.07	821.655	1597800	828.07
5	819.93	824.4	818.98	1281700	824.16
6	819.36	823	818.47	1304000	818.98
7	819	823	816	1053600	820.45
8	816	820.959	815.49	1198100	819.24
9	811.7	815.25	809.78	1129100	813.67
0	809.51	810.66	804.54	989700	809.56
1	807	811.84	803.19	1155300	808.38
2	803.99	810.5	801.78	1235200	806.97

2. Dataset

The following link takes you to the complete historical stock prices for 6 top companies in different sectors (including Apple, Google, J.P. Morgan, Royal Dutch Shell). **Choose one company you would like to work with.**

https://drive.google.com/drive/folders/1JQ8r9iMKU0s-m7kZVOjkpF77gNr82JYS?usp=sharing

Note that the downloaded CSV file has seven columns as follows:

Date, Open, High, Low, Close, Adj_Close, Volume

Remove "Date" and "Adjusted_Close" columns since we do not need them.

3. Requirements

- Split data to training and testing. Use training data to train your models and evaluate the model quality using test data

- Do feature normalization. **Notice that you should never normalize the output feature when training any regression models.** Otherwise, the RMSE of the regression model will be also normalized.
- You must use EarlyStopping and ModelCheckpoint when training FCNN, CNN, and LSTM.
- Tuning the following hyperparameters to record how they affect performance in your report. Tabulate your findings.
  - **Activation:** relu, sigmoid, tanh
  - **Number of layers and neuron count in each layer**
  - **Optimizer:** adam and sgd
  - **Number of CNN layers** (for CNN only)
  - **Number of LSTM layers** (for LSTM only)

4. Grading Breakdown

You may feel this project is described with some certain degree of vagueness, which is left on purpose. In other words, **creativity is strongly encouraged**. Your grade for this project will be based on the soundness of your design, the novelty of your work, and the effort you put into the project.

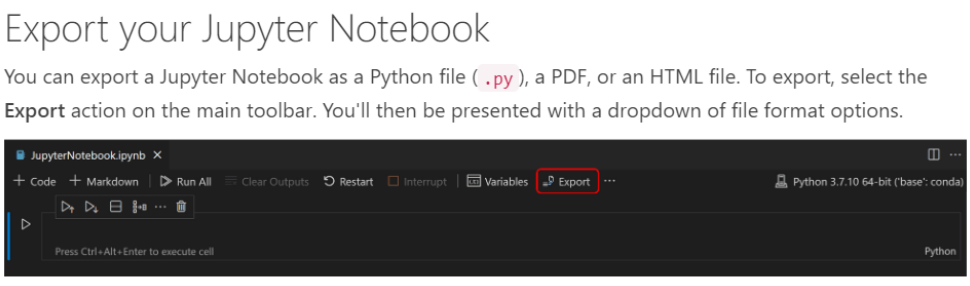
Use the evaluation form on Canvas as a checklist to make sure your work meets all the requirements.

5. Teaming

Students may work independently or in a team of at most 3 people. Think clearly about who will do what on the project. Normally people in the same group will receive the same grade. However, the instructor reserves the right to assign different grades to team members depending on their contributions.

6. Deliverables

(1) The HTML version of your notebook that includes all your source code.



- 5 pts will be deducted for the incorrect file format.

NOTE: Comment each notebook cell with author name and date.

(2) **Your report in PDF format,** with your name, your id, course title, assignment id, and due date on the first page. As for length, I would expect a report with more than one page. Your report should include the following sections (but not limited to):

- Problem Statement
- Methodology
- Experimental Results and Analysis
- Task Division and Project Reflection
- Additional Features

In the section "Task Division and Project Reflection", describe the following:

- who is responsible for which part,
- challenges your group encountered and how you solved them
- and what you have learned from the project as a team.

(3) A **separate text file** named "additional.txt", which describes the additional features you implemented.

All the deliverables must be submitted **by team leader** on Canvas before

10:30 am, Monday, November 17, 2025

NO late submissions will be accepted.

7. Additional Features

- In the project, you predict [Close] of a day based on the last 7 days' data. Can you find the best N value (number of the days we should look back in the past) that yields the most accurate model? Plot RMSE against N value.

- Can you build models using bidirectional LSTM layers and GRU layers, respectively, and compare their performance with your best FCNN/CNN/LSTM models? You can find the implementation for bidirectional LSTM layer and GRU layer here:

  https://keras.io/api/layers/recurrent_layers/

- Can you create a single LSTM model to predict the stock prices for the next few days at a time (e.g., the prices in the next five days)? Plot the true prices and predicted prices in the same chart.

  Hint: train a multi-output regression model with one single output layer using TensorFlow sequential API or a model with multiple regression output layers using TensorFlow functional API.