Project link:

Team name: Crypto Mafia

https://github.com/NickKaparinos/Automated-Cryptocurrency-trading-using-Deep-RL/tree/master

https://www.researchgate.net/publication/382760407_Trading_Bot_for_Cryptocurrency_Market_Based_on_Smart_Price_Action_Strategies

https://www.kaggle.com/datasets/vc9408/btc-data-from-2022-to-2024 (dataset)

You will meet with your mentor TA and review the work you have done since your project has been approved. Reach out to your mentor TA by 04/10 to schedule a meeting within the week of 04/14 - 04/18. Prior to this meeting, please try to have an idea of:

- **Understanding**: have as thorough of an understanding as possible of the paper you're replicating/problem you're solving
-
- **Data**: come with ideas on what data you'll need to use (and how you can access it)
    1. Financial
- **Methods**: have a rough idea of what kind of architecture you plan on implementing
- **Metrics**: have a proposal for your base, target, and stretch goals.

**Note:**

1. **Base goal** = what you think you definitely can achieve by the final due date.
2. **Target goal** = what you think you should be able to achieve by the due date.
3. **Stretch goal** = what you want to do if you exceed your target goal.
   Further note that these goals are flexible and can be re-evaluated at later checkpoints.)

The outline that you submit/write-up should contain the following:

## Title

Deep Learning-Informed Cryptocurrency Trading

## Who

Ilan Brauns, Rosen Iliev, Thomas Stanger, Chris Jeong (ibrauns, riliev, tstanger, cjeong6)

- **Introduction**: What problem are you trying to solve and why?
    - If you are doing something new, detail how you arrived at this topic and what motivated you.
        - Watching how volatile and seemingly irrational the crypto market can be, we became interested in whether deep learning could make sense of it. Our goal was not just to trade but to predict future price movements with measurable accuracy. Traditional models and rule-based bots often fall short in the face of sudden swings and nonlinear relationships. Deep learning, with its ability to recognize complex patterns in chaotic time-series data, felt like a natural fit.
        - What motivated us most was the challenge of predicting the price of Bitcoin. This is considered near-impossible by many, but we were drawn to the idea of

        making it probabilistically predictable. While it is not perfectly accurate, our model should be consistently close within a ±5% margin. That target reflects a realistic, risk-aware view of crypto forecasting. It’s also a way to quantify success without falling into the trap of overfitting or hindsight bias.
    - What kind of problem is this? Classification? Regression? Structured prediction? Reinforcement Learning? Unsupervised Learning? Etc.
        - This is a regression problem because the goal is to predict a continuous numerical value (the future price of Bitcoin). The model learns from historical time-series data to estimate the price at a given date, and success is measured by how close the prediction is to the actual price, ideally within a ±5% margin of error.
- **Related Work**: Are you aware of any, or is there any prior work that you drew on to do your project?
    - Please read and briefly summarize (no more than one paragraph) at least one paper/article/blog relevant to your topic beyond the paper you are re-implementing/novel idea you are researching.
        - We found this github repository that implements a similar idea with reinforcement learning. We are implementing something totally different, but took inspiration from this idea and implemented our own related project.
    - In this section, also include URLs to any public implementations you find of the paper you’re trying to implement. Please keep this as a “living list”–if you stumble across a new implementation later down the line, add it to this list.
- **Data**: What data are you using (if any)?
    - If you’re using a standard dataset (e.g. MNIST), you can just mention that briefly. Otherwise, say something more about where your data come from (especially if there’s anything interesting about how you will gather it).
        - General:
            - https://coinmetrics.io/community-network-data/
        - Bitcoin:
            - https://www.kaggle.com/datasets/prasoonkottarathil/btcinusd
        - Ethereum:
            - https://www.kaggle.com/datasets/prasoonkottarathil/ethereum-historical-dataset
        - Solana:
            - https://www.kaggle.com/datasets/varpit94/solana-data
        - Datasets of a variety of cryptocurrencies/indices
    - How big is it? Will you need to do significant preprocessing?
        - 
- **Methodology**:
    - What is the architecture of your model?
        - We are going to try out various different models and compare the results when trading. This includes MLP, CNN, RNN, LSTM, and Transformer.
    - How are you training the model?
        - We are going to use a historical bitcoin dataset and feed in data from certain spans of time, and then determine a future price using that data. We will then use a standard loss function (i.e. MSE) to determine the loss between the prediction and the actual future price, from the dataset. Using this loss we can train the model to accurately predict future prices.
    - If you are implementing an existing paper, detail what you think will be the hardest part about implementing the model here.
        - We are not directly implementing any existing paper - the paper was inspiration for the project and we are basing our idea off the article.
    - If you are doing something new, justify your design. Also note some backup ideas you may have to experiment with if you run into issues.

        - Lots of models utilize reinforcement learning models but we are not using reinforcement learning as one of our models for this project.
- **Metrics**: What constitutes "success?"
    - What experiments do you plan to run?
        - We plan to run experiments using both simulated and real portfolios
    - For most of our assignments, we have looked at the accuracy of the model. Does the notion of "accuracy" apply for your project, or is some other metric more appropriate?
        - Variance from the Coindesk Market Index will be our metric for this project.
    - If you are implementing an existing project, detail what the authors of that paper were hoping to find and how they quantified the results of their model.
    - If you are doing something new, explain how you will assess your model's performance.
    - What are your base, target, and stretch goals?
        - Base
            - Get the bot to trade within -20% of the Coindesk Market Index
        - Target
            - Get the bot to trade within -5% of the Coindesk Market Index
        - Stretch
            - Get the bot to trade at least identical to the Coindesk Market Index
- **Ethics**: Choose 2 of the following bullet points to discuss; not all questions will be relevant to all projects so try to pick questions where there's interesting engagement with your project. (Remember that there's not necessarily an ethical/unethical binary; rather, we want to encourage you to think critically about your problem setup.)
    - **Why is Deep Learning a good approach to this problem?**
        - **Non-linear pattern recognition**
            - Markets behave non-linearly
                - The markets are influenced by a plethora of factors, e.g., news, shifts in public opinion, general market volatility, economic factors
                - Small inputs like a tweet can cause unpredictable outputs
                    - Lower layers can learn simple patterns, while higher layers combine them into complex strategies
            - Can distinguish signal from noise in volatile, messy crypto data where human intuition often fails
                - Can learn edge cases or irregularities that traditional linear models overlook
            - Some deep learning models, such as Transformers, can pick up on complex, non-obvious price action patterns that traditional models miss
                - Transformers can learn that the sequence of indicator changes matters more than just the values at a single point
        - **Compared to classical ML**
            - Scales better with more data
                - classical ML models plateau or even degrade beyond a certain dataset size
                    - With CNNs or Transformers, performance scales with dataset size
            - More capacity
                - More layers allow the model to capture complex representations across millions of data points
            - Modern compute
                - DL frameworks leverage GPUs, allowing massive datasets to be processed in parallel

  - **How are you planning to quantify or measure error or success? What implications does your quantification have?**
    - Success in this case will be if the bot predicts within a +/-5% [subject to change] margin of error the price of Bitcoin at any date desired. This quantification reflects the ability to predict seemingly unpredictable markets as well as keeping in mind the volatile nature of cryptocurrencies.
      - This quantification balances ambition with realism. It acknowledges the volatility of the crypto market while setting a clear, measurable standard for performance. It also reflects the goal of extracting signals from seemingly unpredictable price movements, showing that deep learning can offer reliable forecasts in a chaotic environment.
- **Division of labor**: Briefly outline who will be responsible for which part(s) of the project.

| Division | People | Responsibilities |
|---|---|---|
| Project Manager | Rosen | ● Coordinates Team Deadlines<br>● Organising Meetings with TAs<br>● Ensuring team meets development deadlines and targets<br>● Provides support to individual team members when needed<br>● Spearheading preparation for Deep Learning Day |
| Deep Learning Manager | Ilan | ● Model design and selection<br>● Oversee dataset selection and handling<br>● Train the model and evaluation<br>● Tweaking model hyperparameters for optimal performance |
| Project Infrastructure | Chris<br>Thomas | ● Writing the majority of the supporting code base<br>● Containerisation of project components into classes<br>● Ensuring there are no circular dependencies in the code base<br>● Implement dataset preprocessing |