Politecnico di Torino

Master Thesis Oral Defense

Deep Reinforcement Learning for Portfolio Optimization

Student:

Gioele Scaletta

April 4, 2024

Supervisors:

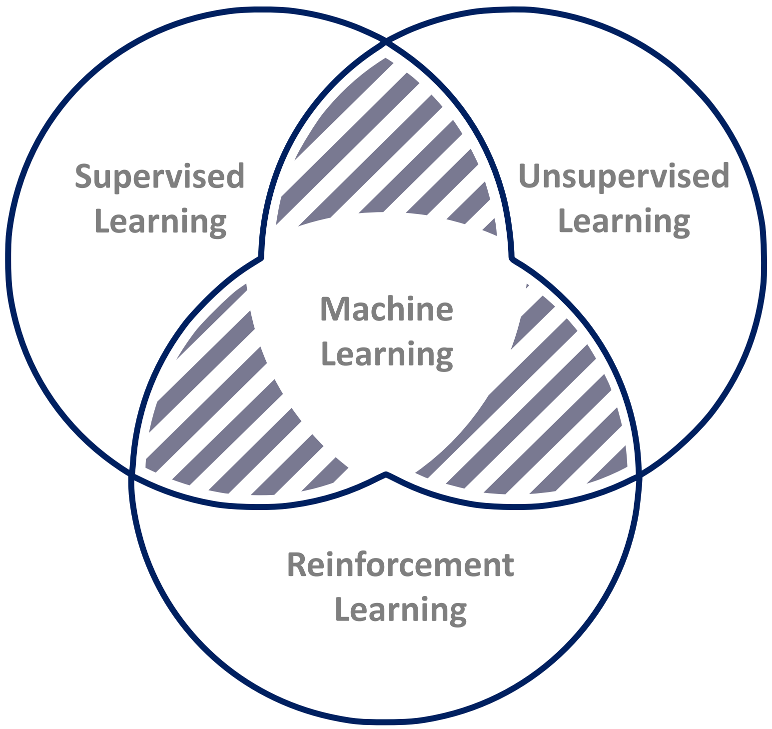
Luca Cagliero
Jacopo Fior

Agenda

Agenda

Reinforcement Learning

Supervised Learning
uses **data with labels** to train the models for predictions

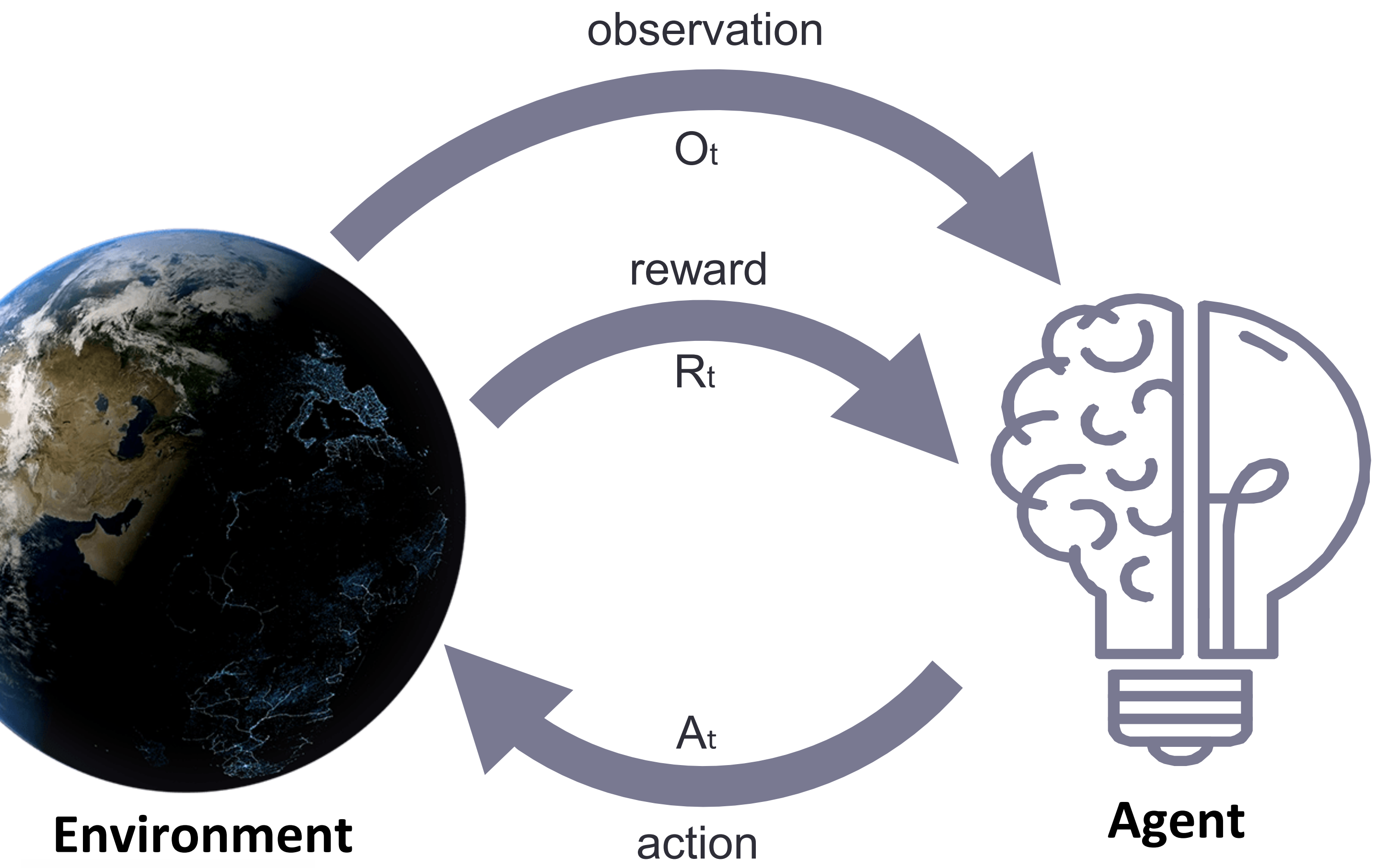


Unsupervised Learning
discovers patterns in **raw unlabeled data**

Reinforcement Learning is designed to **optimize decision-making**
It **learns by train-and-error** taking actions and receiving a reward

Politecnico di Torino
1859

Reinforcement Learning



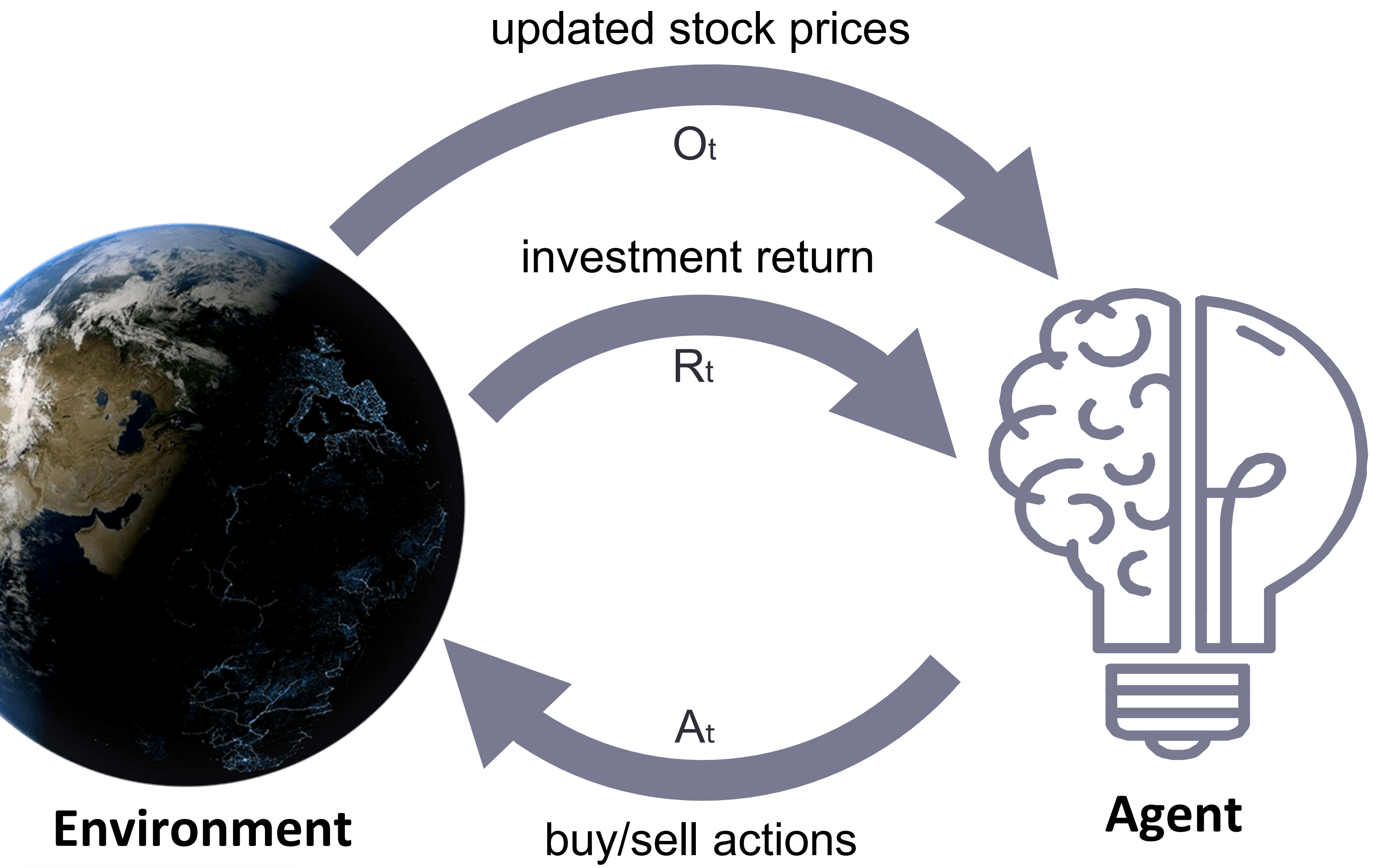
At each step t the **Agent**:
- Receives observation O_t
- Receives scalar reward R_t
- Executes action A_t

At each step t the **Environment**:
- Receives action A_t
- Emits observation O_{t+1}
- Emits scalar reward R_{t+1}

State

Agent's internal representation of history of observations and rewards. The environment contains the stocks from the DJI30 index

Reinforcement Learning



At each step t the **Agent**:
- Receives observation O_t
- Receives scalar reward R_t
- Executes action A_t

At each step t the **Environment**:
- Receives action A_t
- Emits observation O_{t+1}
- Emits scalar reward R_{t+1}

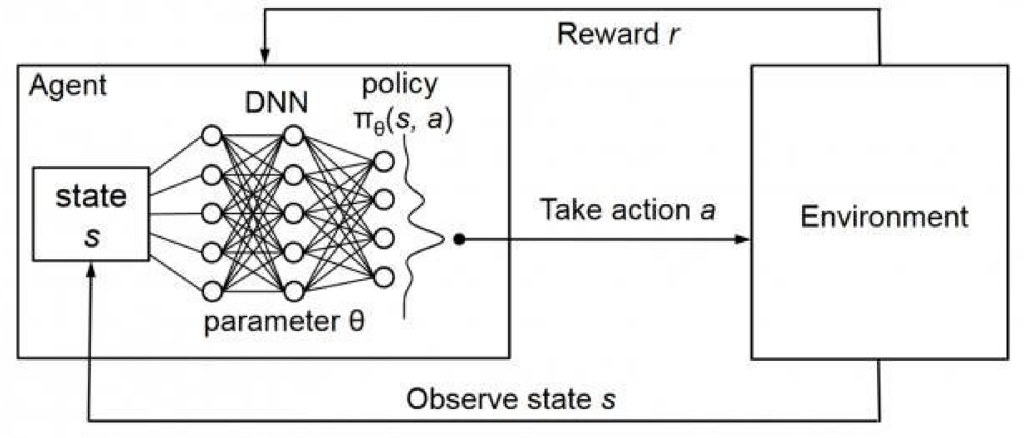
State

Agent's internal representation of history of observations and rewards. The environment contains the stocks from the DJI30 index

Deep Neural Networks in Reinforcement Learning

In this context, **Deep Learning** can be thought of a **universal toolkit that can learn any function**

Among all approaches for RL, deep RL tries to **utilize powerful representations** offered by neural networks to **approximate** complex **components of the agent** such as the policy



Previous methods rely on **mere iterative updates of functions outputs** for each **state-action pair**
Neural networks instead **approximate those values** with **gradient descent**.
Proximal Policy Optimization (PPO) was chosen because of its **stability** and **data efficiency**

Agenda

1 Deep Reinforcement Learning

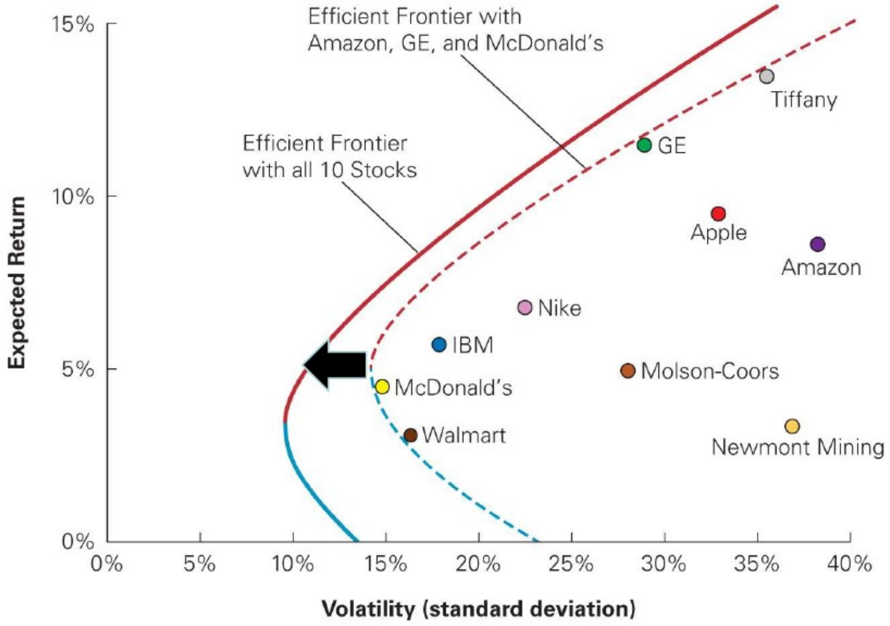
2 Portfolio Optimization

3 Deep RL for Portfolio Optimization

4 Implementation and Variants

5 Results and Conclusions

Markovitz Portfolio Theory

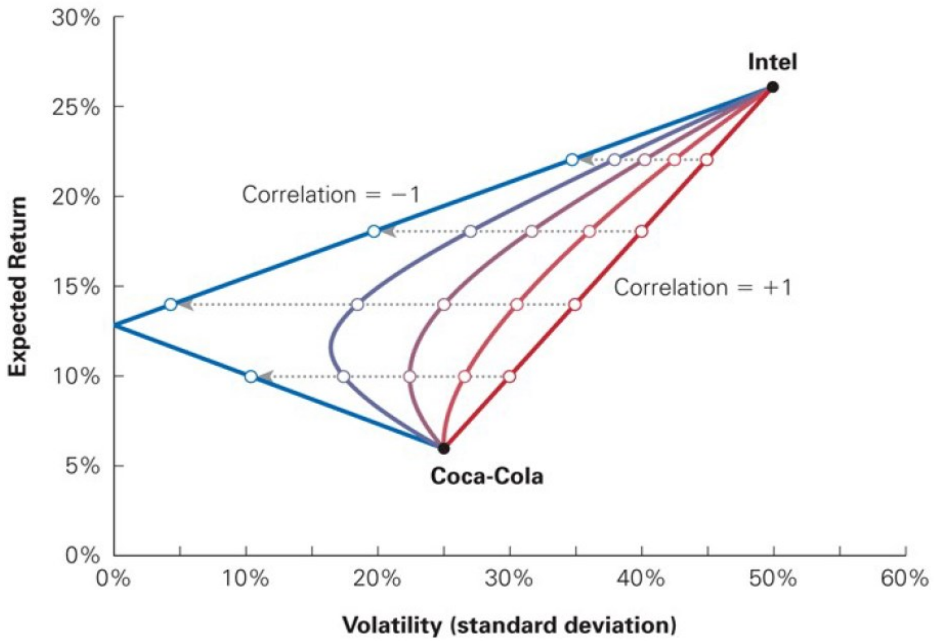


By combining more and more stocks in a portfolio, we **reduce unsystematic risk** through **diversification**.

The **Efficient Portfolio** or Market Portfolio is the portfolio that **only has systematic risk**.

Correlation influences Portfolio volatility, but not return.

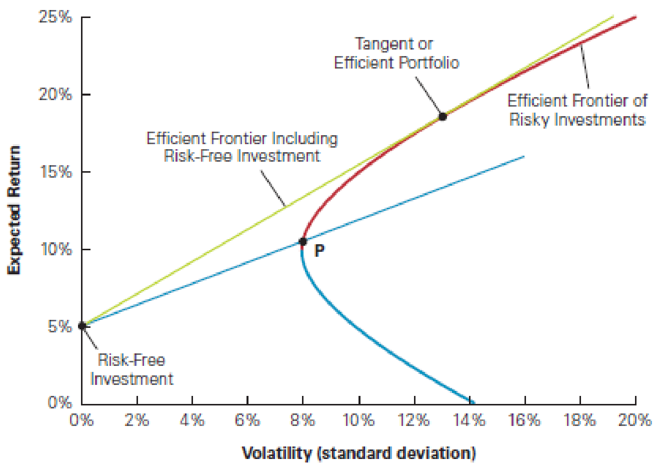
Lower stocks' **correlation** will make the Portfolio **less volatile**



Politecnico di Torino
1859

Tangent Portfolio and Market inefficiencies

The portfolio with the **best reward-to-volatility** (Sharpe ratio) is where the line with the **risk-free** investment is **tangent to the Efficient Frontier**.



$$\text{Sharpe Ratio} \quad ❯ \quad \frac{Portfolio\ Excess\ Return}{Portfolio\ Volatility} \quad ❯ \quad \frac{E[R_P] - rf}{\sigma_P}$$

Therefore, all investors should own the **tangent portfolio**, however, the model makes **assumptions** that are not true in reality because of **market inefficiencies**. Market inefficiencies are what makes above-average returns strategies possible.

Market Inefficiencies

- **Information asymmetries**
- **Biases** and irrational behaviour
- Transaction **fees and taxes**

Investors Biases

- **Familiarity** bias
- **Overconfidence** bias
- Informational **cascade effects**

Politecnico di Torino

Technical Analysis and Indicators

PREDICTION TOOL

Technical analysis provides investors with tools to **predict demand and supply** and its effect on prices. Exploiting market **trends**, price **patterns**, **signals** and **charts** to visually analyze price movements.

OBJECTIONS AND CRITICS

Its efficacy is debated since, in the financial market, the **past** is **not** a **proxy for** the **future**. **Future** prices **do not depend** on **past** prices

HISTORY REPEATS ITSELF

However, empirically, it is observed that **history** still **tends to repeat itself** and this repetitive nature of price trends is often attributed to the **market inefficiencies** described before.

Indicators used

- **Simple Moving Average (SMA)**
- **Exponential Moving Average (EMA)**
- **Moving Average Convergence Divergence (MACD)**
- **Bollinger Bands**
- **Relative Strength Index (RSI)**
- **Commodity Channel Index (CCI)**
- **Directional Movement Index (DMI)**

Politecnico di Torino

Agenda

Deep RL for Portfolio Optimization

What motivated the intersection of this two fields and what are the challenges related with it?

Motivations

- **Similarity** between **RL paradigm** definition and the **stock market** functioning
- **Bypassing** the pretentious step of **predicting the future price** of the assets directly outputting **investment actions**. A full understanding of the stock market underlying forces is not needed
- RL is designed for **decision-making** based on diverse information and this fits well Portfolio Optimization

Challenges

- The representation of the environment embedded in the **model's state** at a specific time-step **must** contain **all useful information** from the **history.**
- RL has been successful in **games** where **data** can be **produced endlessly**. **Stock market** data are publicly available but **limited**.

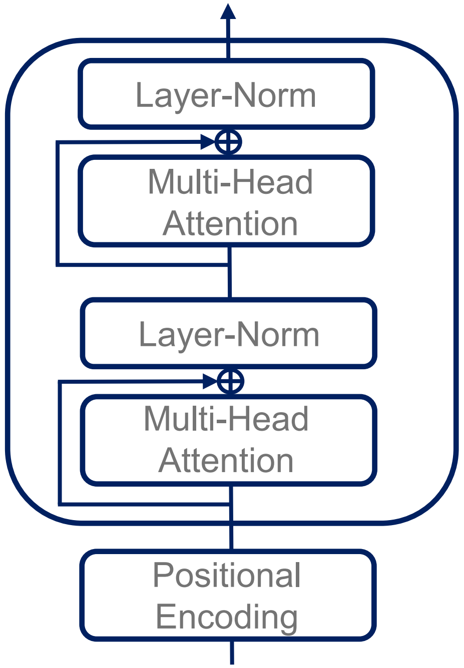
Variants

- GTrXL
- Behavioural Cloning
- TD3+BC

Politecnico di Torino

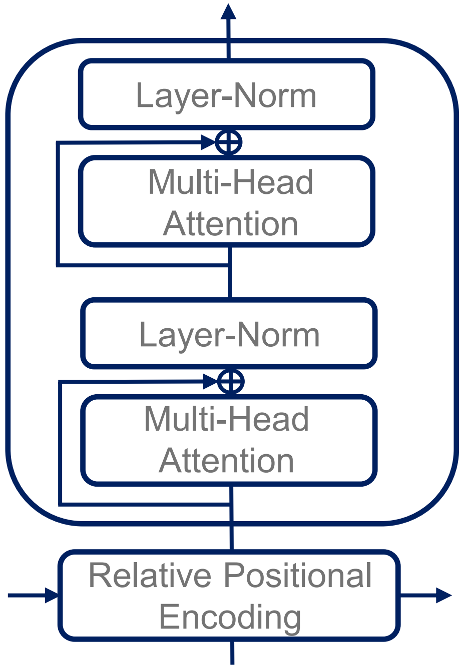
Agenda

Variant 1: GTrXL



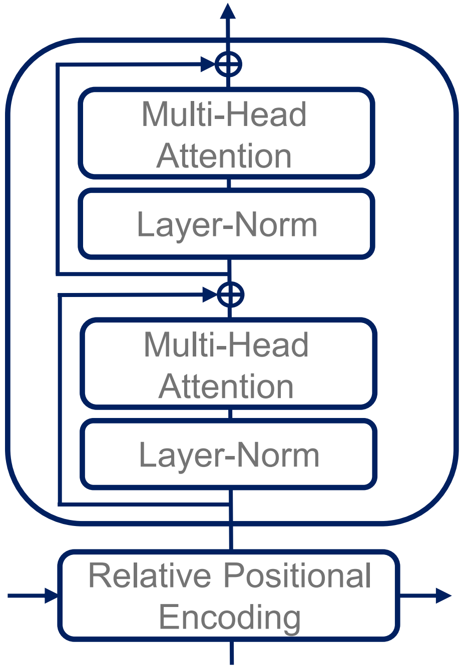
Transformer

The Transformer Architecture was introduced in 2017 for sequence modelling tasks.
Self-attention mechanism allows to model **long-term dependencies**
Only the **encoder** is pictured above



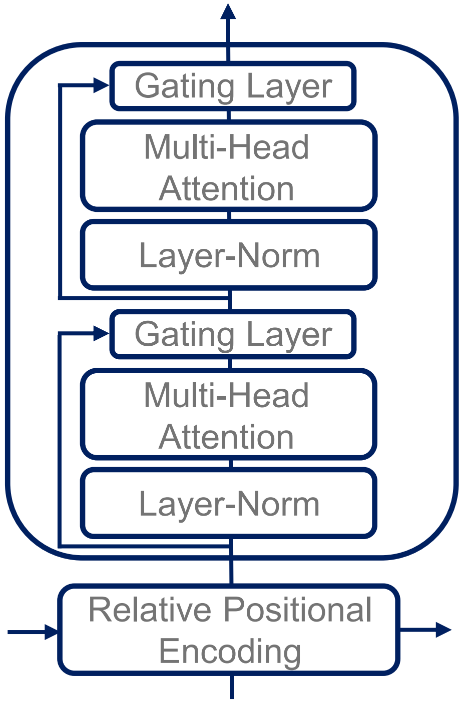
Transformer-XL

It integrates **recurrence** into **deep self-attention networks** thus addressing context fragmentation.
For this to be possible, it introduces a different positional encoding called **relative positional encoding**.



GTrXL
Identity Map Reordering

Layer normalization is moved to the **input stream** of the submodules.
This enables an **identity map** from the **input** to the **output** of the transformer.
Allowing to learn **reactive behaviours** before **memory-based** ones



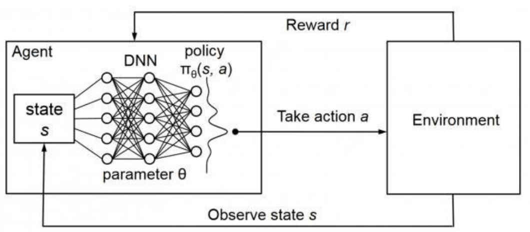
GTrXL
Gating Layers

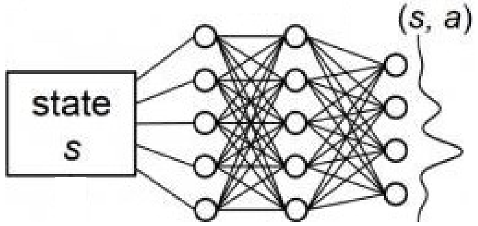
Replaces the **residual connections** with **gating layers** optimizing **stability**.

Variant 2: Behavioural Cloning pre-training

The agent learns from a **dataset of expert demonstrations**

The pre-training is performed as if in a **supervised** learning manner with **states as input** and **actions as labels** and then **training** is **completed with PPO**





The **RL components** described before are **not used** anymore

Actions calculation timeline

distance:
60 days
Ex: 10th Jan-10th March

window:
10 days
Ex: 1st-10th

Beginning of second average calculation

Jan — 10th Jan — 1st March — 10th March

Price 10th Jan: 190$
Avg price 1st-10th Jan: 185$

Price 10th March: 192$
Avg price 1st-10th March: 195$

Simplified entry in the expert state-action pair dataset (date; state; expert action):

(10th Jan; 190; $\frac{195}{185}$ - 1)

The main **parameters** used are the following:

- ***window***: number of days over which the average is computed
- ***distance***: distance between start and end window

An **expert state-actions pairs dataset** is created by repeating the process described on the left for the all training data. (apart from the first *window* days)

Variant 3: TD3+BC with multi-environment S&P 500 training

Previous variants' deficiencies

- **High instability** quantified by evalutating change in performance when changing the seed
- **Lack of diversification** in the investment strategy: the model tends to show a constant **tendency to invest more in a single stock** regardless of the states
- In variant 2, the **pre-training effect vanishes** soon during training

Main implementation features

- The **new model** used was **TD3+BC:**
    - **An offline RL algorithm** incorporating **Behavioural Cloning** into the **Twin Delayed deep deterministic policy gradient (TD3)** model
    - It constrains the policy optimization by adding a **regularizing term** to the **policy gradient** with **expert actions**
- The input **indicators** have been **reduced** and a **new indicator based on efficient frontier weights was added**
- The model was **trained on different environments** composed of sets of **stocks sampled randomly from the S&P 500** index which is a bigger index that **tracks the market similarly to** the **DJI30** (still used for testing).

Agenda

Results

	GTrXL	Behavioural Cloning	TD3+BC
Return difference w.r.t plain PPO	-1.6%	+2.6%	+5.2%
Sharpe difference w.r.t plain PPO	-0.12	+0.07	+0.17

GTrXL low results can be due to technical indicators already condensing past prices information GTrXL **may be more effective** if used **on raw prices**.

This variant shows **better performance,** but the undesired **overinvesting behaviour** is still present

All runs with this variant **beat** the **DJI30 index** used as baseline that has **10.4%** as average annual return.

All variants show high instability that was quantified by evaluating the **effect on performance** when **changing the seed** that is around **20%.** The effect of **changing hyperparameters has similar magnitude.**

Politecnico di Torino

Focus on TD3+BC variant single runs behaviour

TD3+BC BEST RUN PERFORMANCE

Cumul. Ret **404%**

Annual Ret **28.3%**

Sharpe Ratio **1.14**

TD3+BC WORST RUN PERFORMANCE

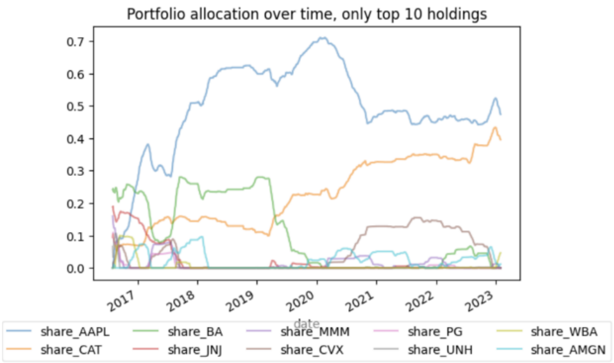
Cumul. Ret **92%**

Annual Ret **10.6%**

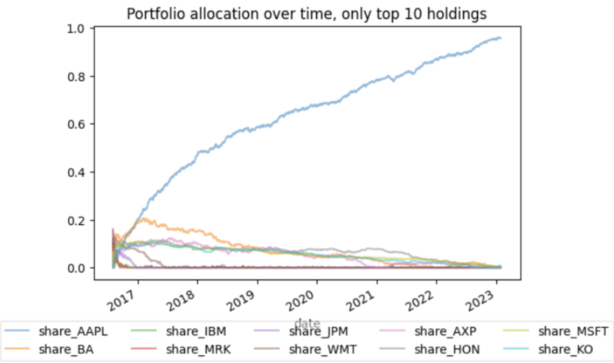
Sharpe Ratio **0.55**

THIRD VARIANT DIVERSIFICATION IMPROVEMENT

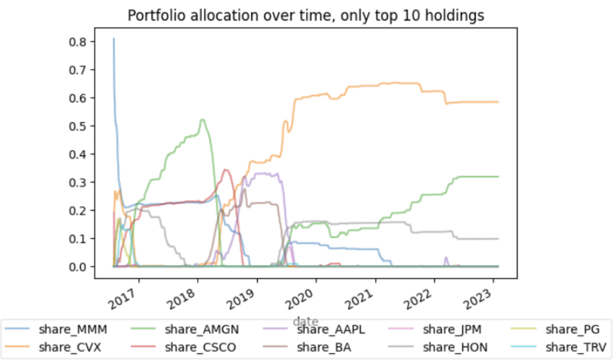
TD3+BC



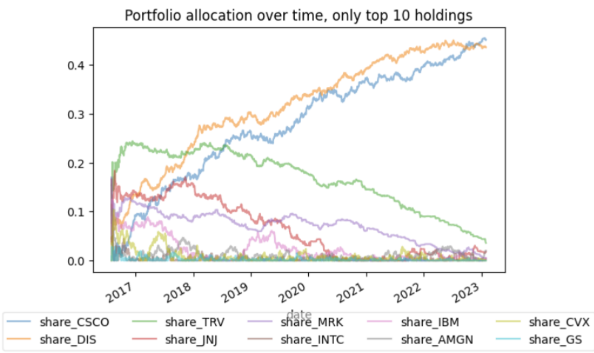
BC-PPO



TD3+BC



BC-PPO



Politecnico di Torino

Training: 2004-01-03 - 2016-07-30
Test: 2016-08-01 - 2023-01-02

Conclusions

The issues related with **seed influence** and **instability remain** and are not fully solved by the proposed variants.

Technical indicators are therefore **not informative enough** for the model to show stable and **consistent decision-making**

Perfomance is very promising since the last variant **always beats the reference index**.

It also diversifies more and eliminates the **tendency to overinvest** in single stocks

Applications could be **very impactful**

The reccomended approach would **focus** on **input data** by finding **other sources of data** other than refining their selection, manipualtion and augmentation and embedding in the state **rather than** increasing the **model complexity**.

Politecnico di Torino

Thank you!

Politecnico
di Torino

Annex – Future Works

REWARD FUNCTION

To have a more diversified investment strategy, a different reward function could be used so as to also **reward diversification** by negatively weighing volatility:

- **Sharpe ratio**
- **Differential Sharpe Ratio**
- A **Custom Reward** obtained by dividing Return by volatility raised to a specific power based on how much we want to weigh volatility

OTHER DEEP LEARNING MODELS

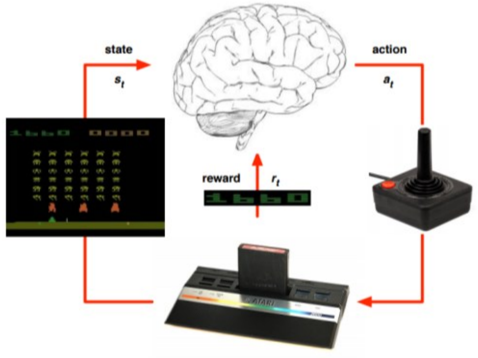
- **Neuroevolution**
- **Graph Neural Networks** applied to time-series
- **Transformer-based decision-making** models
- However this is **probably not impactful** since we observed that the **input data do not contain enough information**

INPUT DATA SELECTION AND MANIPULATION

- **Focus even more on the input data selection** by trying to find the most important technical indicators maybe using SHAP values
- Find **other sources of data** that have suitable **granularity** and **go back in time far enough**
- **Introduce** more **noise in the environment** during training

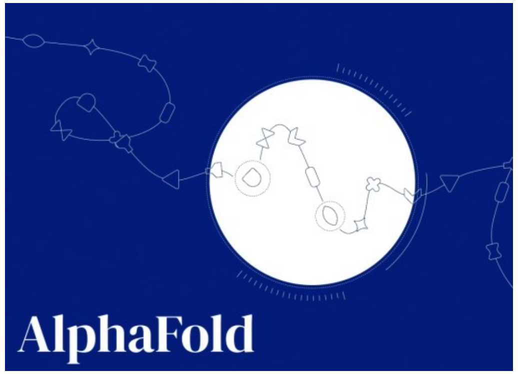
Annex - Applications of Deep Reinforcement Learning

Atari, 2014



Main first success.
It takes as input game's **raw image pixels** as **state** and outputs game decisions

AlphaFold, 2021

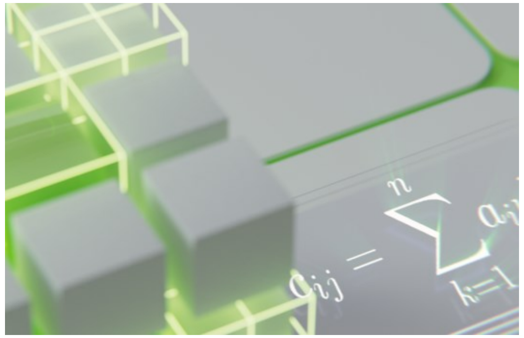


The first successful application **outside of game-playing domain.**
The model predicts the way **amminoacids** spontaneously **fold** to form **3D proteins** structure

AlphaGo, 2016

It **won** 4 to 1 games against the **world champion**.
Go is a game where **intuition is fundamental** and it is not possible to calculate all combinations like in chess

AlphaTensor, 2022



The model automates **algorithmic discovery** for **matrix multiplication** optimizing complexity and speed.

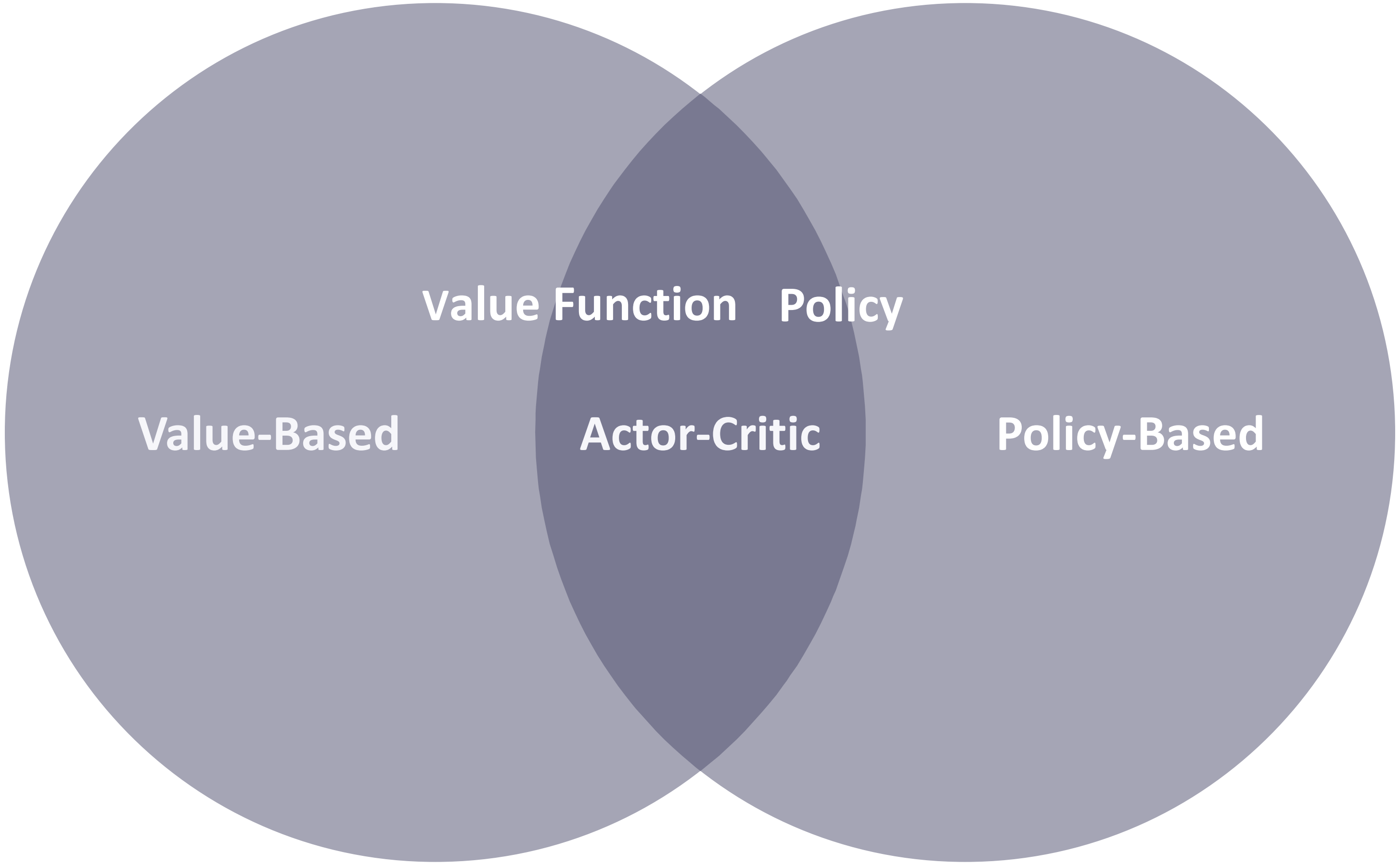
Politecnico di Torino

Annex - Agents

VALUE FUNCTION
Function that estimates total future expected return starting from a state or state-action pair.
Value-based agents only have value function and the policy is implicit

POLICY
Function that, given a state, outputs an action, it is the agent behaviour.
Policy-based agents output actions using the policy directly and there is no value function

ACTOR-CRITIC
Uses both policy and value function.
Critic updates **action-value function** while **Actor** updates **policy** in direction suggested by Critic



Politecnico di Torino

Annex - Important Concepts in Reinforcement Learning

In order to have a complete picture of Reinforcement Learning, some fundamental definitions are missing:

EXPLOITATION VS EXPLORATION

Make the **best action** to **maximize** the **reward** given the **information** that have been **collected** so far

Try **suboptimal actions** to gain **new experience** and find **more information** about the environment

MODEL-FREE VS MODEL-BASED

When the **functioning** of the **environment** is **unknown**

An optimistic **model** of the **environment** is built and then **planning** is used

EXPERIENCE REPLAY

Experience **not** anymore **sequential** but:

- Collection of **experience transitions** as (s_t, a_t, r_t, s_{t+1})
- Transition **sampling**
- Stochastic **gradient descent** on sample

The main **Advantages** include:

- **Reduction** of **autocorrelation** and consequent **instability**
- Data more **independently** and **identically distributed**
- Better **data efficiency**

Annex - Capital Asset Pricing Model

$E[R]$ ❯ Risk-Free Interest Rate + Risk Premium ❯ $r_f + \beta \times (E[R_{Mkt}] - rf)$

Common vs Independent Risk

- **Common or Systematic Risk** impacts the whole market and shows perfect corralation. Ex: market wide news
- **Independent or Unsystematic Risk** impacts a specific security and therefore it is uncorrelated. Ex: single company news
- **Volatiliy** is a measure of Total Risk

Beta and Market Risk Premium

- **Beta β** is the sensitivity to systematic risk and is the change in return for a 1% change in the market protfolio return
- The **Market Risk Premium** $(E[R_{Mkt}] - rf)$ over the risk-free rate is the reward for holding the market Portfolio $(\beta=1)$
- The result of the CAPM $(E[R])$ is the **Expected Return of a security**

Annex - Algorithm Implementation

PROXIMAL POLICY OPTIMIZATION (PPO) ALGORITHM

Main features

It is an **Actor-Critic** algorithm that aims at taking the biggest possible improvement step on its **policy** by using the available data without causing convergence issues.
It achieves it by **adding a penalty** to the **objective function**

Choice rationale

This feature improves **data efficiency** with a very simple trick **reducing** algorithm **complexity**

Main libraries used

Ray RLlib

It has been used for the PPO implementation as well as the variants implementations with some modifications.

FinRL

It has been used to take inspiration for the environment, data preprocessing and results manipulation parts

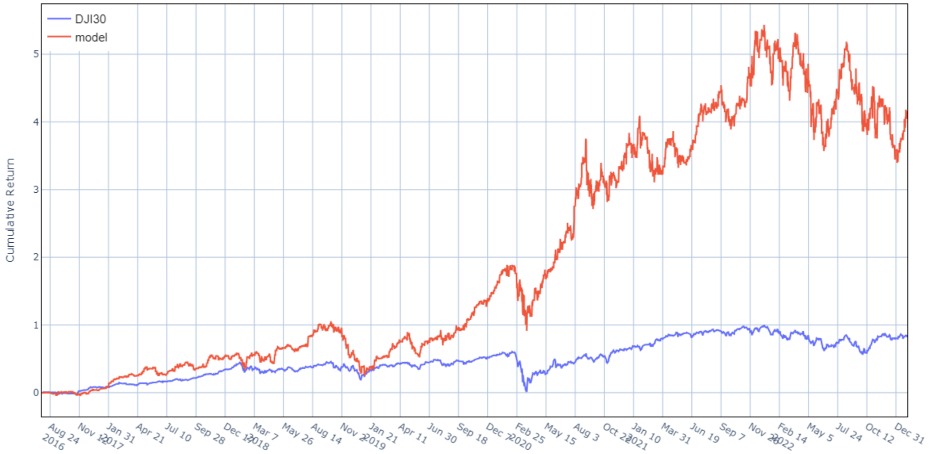
Annex - Results with different time granularities

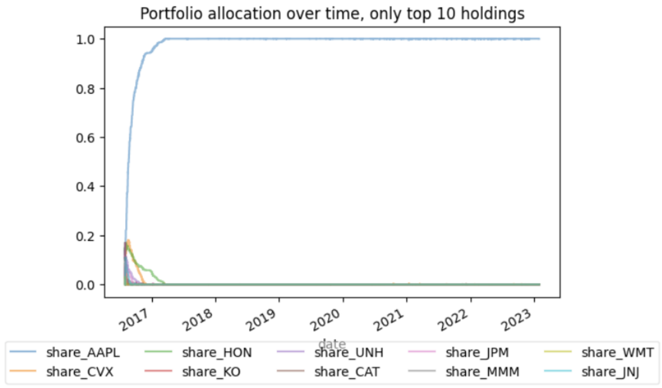
1 minute, 5 minutes, 30 minutes, 1 hour
additional **time granularities** have been tried
hourly granularity seemed the **most promising** one.

However, experimenting more, while, in some instances, higher time granularity seem to better exploit the additional information leading to **improved results**, it also often lead to **worst ones**.

Therefore, the conclusion is that **higher granularity** leads to even **greater instability**.

Most successful run with *hourly* granularity

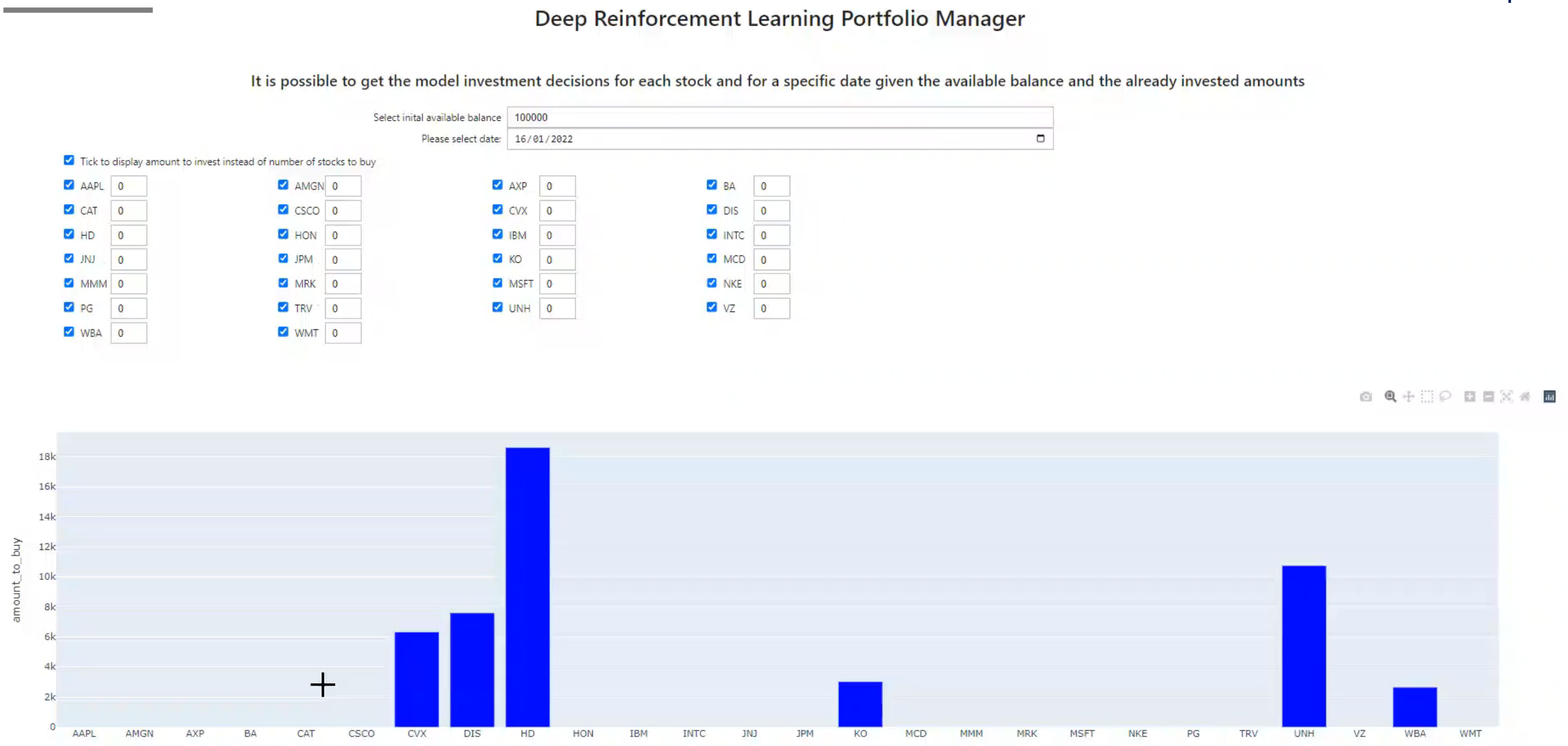




Training: 2004-01-03 - 2016-07-30
Test: 2016-08-01 - 2023-01-02

Politecnico di Torino

Annex - Web Application Demo



Politecnico di Torino

Annex - Implementation

```
│   ├── model
│       └── agent.py
│   ├── environment
│       └── environment.py
│   ├── utility_functions
│       ├── data_retrieval_preprocessing.py
│       ├── plot.py
│       ├── API.py
│       └── func.py
│   ├── colab_notebooks
│       ├── run_results
│       ├── notebook_gtrxl.ipynb
│       ├── notebook_time_granularities.ipynb
│       ├── notebook_imitation_learning.ipynb
│       └── notebook_ppo.ipynb
│   ├── VoilaWebApp.ipynb
│   ├── references
│   ├── update_presentations
│   ├── config.py
│   └── main.py
```

This code interacts with **Ray RLlib library** to use and customize their implementation of the **Reinforcement Learning models**

This folder contains the **environment** which was inspired by **FinRL library**

This folder contains different functions mainly related with **data retrieval**, **preprocessing** and **results manipulation**

This folder contains the **notebooks** used to **run** the **algorithms** in the different configurations.

The **main** and **config** files used respectively to run locally and to specify configuration parameters

Politecnico di Torino

Annex – Different time granularities runs details

Time granularity	Training Period	Test Period	Return of PPO with pretraining	Sharpe of PPO with pretraining	Marwil without PPO return
5 min	01/01/2019 – 2021/12/31	01/01/2022 – 2022/12/31	-2.50 %	-0.01	-4.72 %
30 min	*	*	-30.80 %	-0.25	-40.90 %
1 hour	*	*	**9.00 %**	**0.18**	-3.59 %
daily	*	*	-9.31 %	-0.45	-5.37 %
1 min	01/06/2020 – 2021/12/31	01/01/2022 – 2022/05/31	-0.41 %	-0.14	**-0.41 %**
5 min	01/01/2016 – 2018/12/31	2019/01/01 – 2019/12/31	-1.26%	0.01	0.12%
30 min	*	*	**16.50%**	**0.30**	11.30%
1 hour	*	*	6.69%	0.16	10.08%
daily	*	*	1.81%	0.19	**15.41%**

Politecnico di Torino