Informatics

Adaptive Devisentermin-Hedgingstrategien unter Verwendung von Deep Reinforcement Learning

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Datenwissenschaften

eingereicht von

Branimir Raguz
Matrikelnummer 12123474

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Prof. Dr. Clemens Heitzinger

Wien, 20. März 2025

Branimir Raguz

Clemens Heitzinger

Adaptive Foreign Exchange Hedging Strategies Using Deep Reinforcement Learning

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Branimir Raguz
Registration Number 12123474

to the Faculty of Informatics

at the TU Wien

Advisor: Prof. Dr. Clemens Heitzinger

Vienna, March 20, 2025

Branimir Raguz

Clemens Heitzinger

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Branimir Raguz

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT-Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 20. März 2025

Branimir Raguz

Danksagung

Ich möchte mich ganz herzlich bei meinem Betreuer, Prof. Dr. Clemens Heitzinger, für seine wertvolle Unterstützung und Anleitung während dieser Arbeit bedanken. Sein Rat und seine Expertise haben mir geholfen, tiefer in dieses Fachgebiet einzutauchen und die Herausforderungen dieser Thesis zu meistern.

Ein riesiges Dankeschön geht an meine Familie, die mich immer unterstützt und motiviert hat. Ebenso möchte ich mich bei meinen Freunden bedanken, mit denen ich gemeinsam gelernt, schwierige Phasen überstanden und das Studium umso bereichernder gemacht habe.

Acknowledgements

I want to sincerely thank my supervisor, Prof. Dr. Clemens Heitzinger, for his guidance and support throughout this thesis. His advice and expertise helped me dive deeper into this field and tackle the challenges that came with it.

A huge thank you to my family for always supporting me and keeping me motivated during this journey. I also want to thank my friends for learning together, pushing through tough times, and making this university experience so much better.

Kurzfassung

Diese Arbeit untersucht den Einsatz von Deep Reinforcement Learning (DRL) zur Steuerung von Devisenrisiken, indem zwei DRL-Algorithmen – Double Deep Q-Network (DDQN) und Proximal Policy Optimization (PPO) – mit traditionellen technischen Benchmark-Strategien, Relative-Strength-Index (RSI) und Moving-Average-Crossover (MAC), verglichen werden. Wir entwickeln eine realistische Umgebung, in der Zinsdifferenziale und dynamische Transaktionskosten (Spreads, Kommissionen, Slippage) in die Reward-Funktion integriert sind. Jede Strategie wird auf USD/CHF-Daten von 1980 bis 2024 anhand von Kennzahlen wie Gesamtrendite, annualisierte Rendite, Sharpe-Ratio, maximaler Drawdown, Volatilität und Beta bewertet. Die Ergebnisse zeigen, dass PPO die beste Performance liefert – mit den höchsten Renditen (14,96% gesamt, 4,76% p.a.), den besten risikoadjustierten Kennzahlen (Sharpe 0,20) und kontrollierten Drawdowns – während MAC als einfache, kosteneffiziente Alternative wettbewerbsfähige Renditen erzielt (12,23% gesamt, 3,92% p.a.). RSI erweist sich als zu konservativ und DDQN zeigt übermäßige Volatilität. Trotz dieser starken Ergebnisse konnten die DRL-Agenten jedoch keine Positionen über längere Zeit halten und führten zu viele Transaktionen aus – ein Ergebnis, das für FX-Hedging unerwünscht ist und in künftigen Arbeiten behoben werden muss.

Abstract

This thesis investigates the use of deep reinforcement learning (DRL) for managing foreign exchange (FX) risk by comparing two DRL algorithms—Double Deep Q-Network (DDQN) and Proximal Policy Optimization (PPO)—against traditional technical benchmarks, Relative Strength Index (RSI) and Moving Average Crossover (MAC). We develop a realistic environment that incorporates interest-rate differentials and dynamic transaction costs (spreads, commissions, slippage) into the reward function. Each strategy is evaluated on USD/CHF data from 1980–2024 using metrics such as total and annualized return, Sharpe ratio, maximum drawdown, volatility, and beta. Results show that PPO delivers the strongest performance—achieving the highest returns (14.96% total, 4.76% p.a.), best risk-adjusted metrics (Sharpe 0.20) and controlled drawdowns—while MAC offers a simple, cost-efficient alternative with competitive returns (12.23% total, 3.92% p.a.). RSI proves overly conservative and DDQN exhibits excessive volatility. However, despite these strong results, the DRL agents were unable to maintain positions for extended periods and executed too many transactions—an outcome that is undesirable for FX hedging and must be addressed in future work.

Contents

Kurzfassung **xi**

Abstract **xiii**

Contents **xv**

1 Introduction **1**
1.1 Problem Statement 1
1.2 Research Objective 1
1.3 Challenges in Developing Effective FX Hedging Strategies 2
1.4 Contributions 2

2 Related Work **5**

3 Methodology **7**
3.1 Baseline Strategies 7
3.2 Introduction to Reinforcement Learning 10
3.3 Deep Reinforcement Learning Models 11
3.4 LSTM as Base Policy 15
3.5 Evaluation metrics 16

4 Results **19**
4.1 Dataset, Features, and Preprocessing 19
4.2 Reinforcement-Learning Environment 22
4.3 Practical Application of the Strategy 24
4.4 Financial Costs and Trading Fees 24
4.5 Modeling Process 26
4.6 Training Results and Learning Progress 27
4.7 Test Set Performance Analysis 30

5 Discussion **33**
5.1 Interpretation of Results 33
5.2 Limitations of the Study 33
5.3 Future Research Directions 34

6 Conclusion 35
6.1 Summary of Findings 35
6.2 Final Thoughts on DRL for FX Hedging 35
6.3 Potential for Real-World Implementation 36

Overview of Generative AI Tools Used 37

Übersicht verwendeter Hilfsmittel 39

List of Figures 41

List of Tables 43

List of Algorithms 45

Bibliography 47

CHAPTER 1

Introduction

1.1 Problem Statement

The **foreign exchange (FX) market** is the **largest financial market** in the world, with an average daily trading volume of $5.1 trillion, according to [1]. Due to the effects of globalization, there is a surge in companies which have subsidiaries in different countries and serve clients around the world.
Therefore, it is extremely important to manage the **risk** associated with **currency fluctuations**. Businesses and investors dealing with multiple currencies face **risks** due to **exchange rate movements** that affect **profits**, **financial planning**, and the overall functioning of the firm and investments.

Following simple **technical strategies** such as a **Relative Strength Index (RSI)** or a **Moving Average Crossover (MAC)** provides a decent level of protection, but a more **dynamic hedging strategy** is needed.
This thesis explores how **Deep Reinforcement Learning (DRL)** can be applied to **hedging the currency risk** and whether we are able to develop an **adaptive strategy** that balances **risk** and outperforms the **baseline strategies**.

1.2 Research Objective

The goal of the thesis is to develop an FX hedging strategy using Deep Reinforcement Learning (DRL). More specifically, we will develop and compare one off-policy (Double Deep Q-Learning DDQN), and one on-policy (Proximal Policy Optimization PPO), against two simple technical strategies like the Relative Strength Index (RSI), and the Moving Average Crossover (MAC).

By comparing these 4 strategies, we will understand how the DRL hedging strategies match up to simple technical strategies. The performance will be measured by a set of different financial metrics like total return, Sharpe Ratio, maximum drawdown, and more. These metrics will provide a broad way to assess the performances between strategies.

1.3 Challenges in Developing Effective FX Hedging Strategies

Developing an FX hedging strategy with traditional ML is very difficult due to non-stationarity of financial data, high volatility, and noise. FX markets are influenced by macroeconomic events, interest rates, political factors, and other global developments. This makes it extremely challenging to make reliable long-term predictions. A major issue is that hedging requires long-term planning and maintaining a position over an extended period of time. Standard ML models aren't very capable in generalizing well for long term predictions. Financial time series also have a low signal-to-noise ratio which causes models to overfit to random noise. The superiority of a DRL model is that it learns an adaptive policy because it interacts with the environment. The model dynamically adjusts its strategy in response to real-time market conditions, and if modeled correctly can avoid being constrained by outdated historical patterns. Furthermore, a DRL model optimizes for long-term rewards, which is excellent in the context of hedging a currency risk. It would be ideal if the strategy would make few but qualitative decisions, like staying in a hedge for 6–18 months, or letting the value of the currency increase by being outside of a hedge for long.
With the constant update of the strategy and policy of the DRL based on market conditions, the model has the potential to be superior in developing a hedging strategy for the FX market.

1.4 Contributions

1.4.1 Contributions of this Thesis

The contributions of this thesis to the field of FX hedging are the following:

Introducing the interest rate differential calculations inside the actual reward function of the DRL model. This is done to make the reward function as close to reality as possible, and in the hope that the model would then realize when to actually hedge the currency pair, and when to let the currency of his choice increase in value. The interest rate differential is embedded in the future contracts used for hedging the currency pair, and are therefore a very viable factor.

The comparison was made between the simple technical strategies and the DRL strategies. This was not anything new, but it helped to expand the existing literature on DRL hedging in FX, which is an area with limited research. The comparison was also done in a systematic way using a diverse set of financial metrics. The financial metrics include total return, Sharpe ratio, and maximum drawdown.

Realistic transaction costs were introduced, including spread, brokerage fees, and slippage when entering or exiting positions. This in combination with the reward function enables a clear representation of the DRL hedging in FX.

1.4.2 Usage of AI

Chat GTP was used to generate the abstract of the thesis. It was used in Section 3.1 to organize the text, and write small sentences that clearly explain parts of the strategies. It was used very briefly in Section 4.1 and Subsection 4.4.2. It was used in Section 4.2 to clearly explain and organize the reward function with the formulas. All the larger paragraphs were written without ChatGPT. Exceptions are Sections 3.3 and 3.4 which were co-written with ChatGPT, focusing on the organization and structure of the formulas and text. 1/5 of the thesis was written jointly with ChatGPT.

1.4.3 Theoretic Results

As this approach is relatively new in the literature, there were no results to compare against. All the results that are presented in this thesis were deduced/proved by me.

1.4.4 Explanation of Code

The code retrieves the original Close price for USD/CHF from TradingEconomics. Using the ‘ta’ library, it computes technical indicators (features). The dataset is then preprocessed and prepared for training. Functions for calculating financial metrics (Sharpe ratio, total return, maximum drawdown) are implemented.

The Moving Average Crossover (MAC) strategy and the Relative Strength Index (RSI) strategy are backtested.

The entire environment is built from scratch with Gymnasium: the reward function is defined (including transaction fees), and an LSTM policy network is implemented for both DDQN and PPO using Ray RLlib. Random-search hyperparameter tuning is performed for DDQN and PPO, and the best configurations are tested five times to assess stability and performance. Graphs and visualizations are generated.

Libraries Used

- `gymnasium` — RL environment framework
- `tradingeconomics` — USD/CHF data retrieval
- `ta` — technical indicators computation

- `pandas`, `numpy` — data manipulation & numerical calculations
- `scikit-learn` — data normalization
- `ray[rllib]` — implementation of DDQN and PPO
- `matplotlib` — plotting and visualizations

CHAPTER 2

Related Work

“Superhuman capabilities of RL learning algorithms have been demonstrated in various areas, and the list of extraordinary capabilities of AI systems built on reinforcement learning is continually expanding. Prominent examples are playing backgammon, Atari 2600 games, many more computer games, card games, chess, Go, and shogi at superhuman levels. Probably most famously, however, reinforcement learning is the last and crucial step in training large language models such as ChatGPT” [6]. It was not long until RL or DRL was increasingly being applied to algorithmic trading. The paper [14] from the University of Edinburgh provided a detailed analysis of DRL applications in trading.

Due to different market assumptions and setups of the experiments, the study showed that it is very difficult to compare the DRL models to each other. Some DRL models have reported annual returns exceeding 20%. This information should be taken with a grain of salt regarding the real-world applicability, as those returns outperform even the best hedge fund and long term benchmarks like the S&P 500.

The paper also addresses the difficulty of selecting the appropriate trading timeframe. It is complicated, due to the fact that long-term price movements are highly unpredictable due to macroeconomic events. Predicting short-term prices can be easier, but in the context of high frequency trading, the transactions costs quickly accumulate, and the noise hinders the models performance. Striking the right balance remains an open research questions.

A limitation in the field is that, there is little high-quality practically tested research. Most studies and experiments present their results in the context of backtesting their models on past data. While this is perfectly fine if the researchers align the transactions costs and all the factors as realistic as possible, that is very rarely done. This makes it unclear if the models would perform this well in real live trading. There is also a question to be asked: if a model were genuinely profitable, would the researchers even disclose it? However, the study showed promise and potential of DRL in algorithmic trading.

- [20] This paper from the University of Oxford demonstrates the effectiveness of DRL when using Deep Q-Networks (DQN) in creating a successful trading strategy. It was shown that a DQN agent trained on historical price data can outperform a traditional trading strategy. This matches our assumption which is that DRL can learn the complex market dynamics, and due to it interacting with the environment, can adapt its trading decisions.
- [17] This paper from the University of Liege, Belgium, also explored the effectiveness of DRL in algorithmic trading. The conclusion of the paper is that DRL agents are highly sensitive to the design of the reward function, and the state representation. It was discovered that for a DRL agent to be profitable, there is a need to carefully feature engineer, and shape the reward function. This matched our assumption as well, as DRL is highly sensitive. They finally compared a DQN strategy with baseline approaches, and showed the superior performance of DRL.

There is a lack of existing research in applying DRL to FX hedging with forward contracts, and we aim to provide a capable starting point for it. The goal of this thesis is not to earn as much money as possible, but to manage the risk of the hypothetical client. By developing a DRL-based agent that learns when to fully hedge or when not to, we apply an existing method applied to a new setting in the financial context.

CHAPTER 3

Methodology

3.1 Baseline Strategies

We have selected two simple technical based trading strategies for a benchmark comparison for our DRL hedging strategy. The two strategies we have selected the Relative Strength Index (RSI) and the Moving Average Crossover (MAC). These are the most used technical strategies in scientific papers, and the industry as a simple comparison strategy.

3.1.1 Relative Strength Index (RSI)

The **Relative Strength Index (RSI)** is a **momentum oscillator** used to identify **overbought** or **oversold** conditions in a market. It ranges from **0 to 100** and is defined as [8]

$$\mathrm{RSI} := 100 - \frac{100}{1 + \mathrm{RS}} \tag{3.1}$$

where

$$\mathrm{RS} := \frac{\text{Average gain over a specified period}}{\text{Average loss over a specified period}}. \tag{3.2}$$

Typically, the RSI is calculated using a 14-day period. In our implementation, we use the RSI to generate hedging signals as follows:

- **Overbought Condition (RSI > 70):** If the **RSI** is above **70**, the asset is considered **overbought**. In this situation, we would exit the hedge, as the CHF is expected to gain value in comparison to the USD. Therefore, the money that is incoming regularly will be worth more and more upon conversion to USD.

- **Oversold Condition (RSI $<$ 30):** If the **RSI** is under **30**, the asset is considered **oversold**. In this situation, we would enter the hedge, as the CHF is expected to lose value in comparison to the USD. By hedging, we protect our money in CHF to potentially be worth less in the future. We lock in the current exchange rate, and hold it until the model suggests and exit from the hedge.
- **Neutral Condition (30 $\leq$ RSI $\leq$ 70):** In this range, we maintain the current position of being in hedge, or outside the hedge.

This **RSI-based strategy** provides a **simple yet effective** way to **dynamically adjust** the **hedging position** based on **market momentum**.

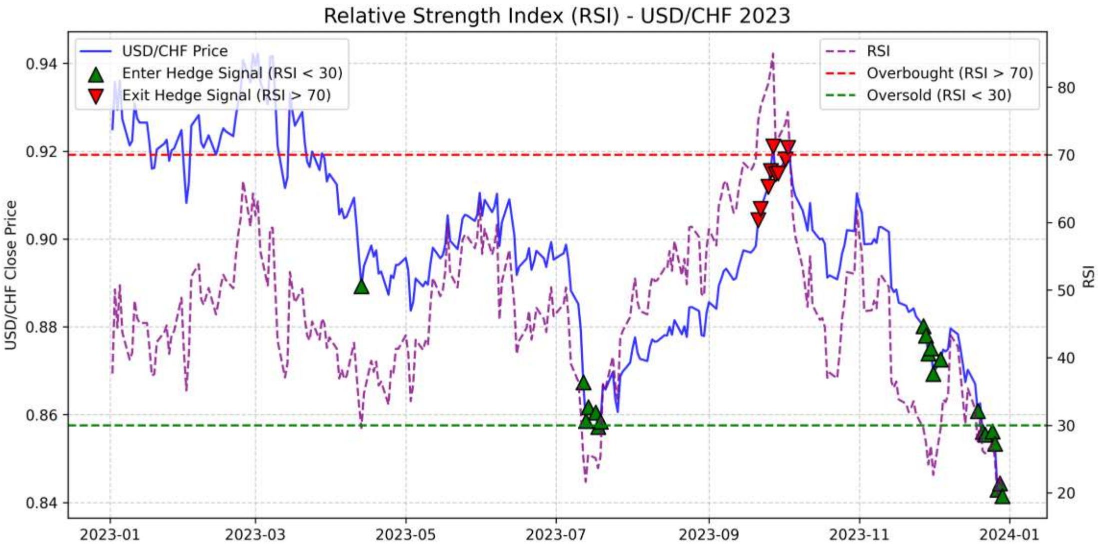

Figure 3.1: Relative Strength Index (RSI) and corresponding trading signals throughout the test period.

Figure 3.1 presents the **Relative Strength Index (RSI)** throughout the **test period**, highlighting key **trading signals**. Periods where the **RSI** exceeds the overbought threshold (RSI $>$ 70) indicate a signal to **exit the hedge**. Periods where the RSI fall short of the **oversold threshold** (RSI $<$ 30), the strategy suggest to **enter the hedge**. This graphs helps with visualizing the signals, and understanding the strategy better.

3.1.2 Moving Average Crossover (MAC)

The **Moving Average Crossover (MAC)** strategy is a **trend-following strategy** that uses two **moving averages**: a **shorter-period moving average** ($\text{SMA}_{\text{short}}$) and a **longer-period moving average** (SMA_{long}) [9]. A **moving average** is calculated by taking the **average price** over a specified period. In this thesis, we calculate the **moving average** as the **sum of the closing prices**, divided by the **number of business days**, i.e.

$$\text{SMA} := \frac{\text{Sum of closing prices over a specified period}}{\text{Number of periods}}. \tag{3.3}$$

The strategy generates trading signals when the two moving averages cross each other. In our implementation, we use the MAC to generate hedging signals as follows:

- **SMA$_{short}$ Crosses Above SMA$_{long}$:** This is considered a bullish signal, indicating a upward trend. In this situation, we would enter the hedge as the CHF is expected to lose value in comparison to the USD.
- **SMA$_{short}$ Crosses Below SMA$_{long}$:** This is considered a bearish signal, indicating a downward trend. In this situation, we would exit the hedge as the CHF is expected to lose value in comparison to the USD.
- **No Crossover:** In the absence of a crossover, we maintain the existing positions (either hedged or unhedged).

This MAC-based strategy is widely used in the financial industry as a benchmark for comparison and can deliver satisfactory results when the financial instruments have long and stable trends.

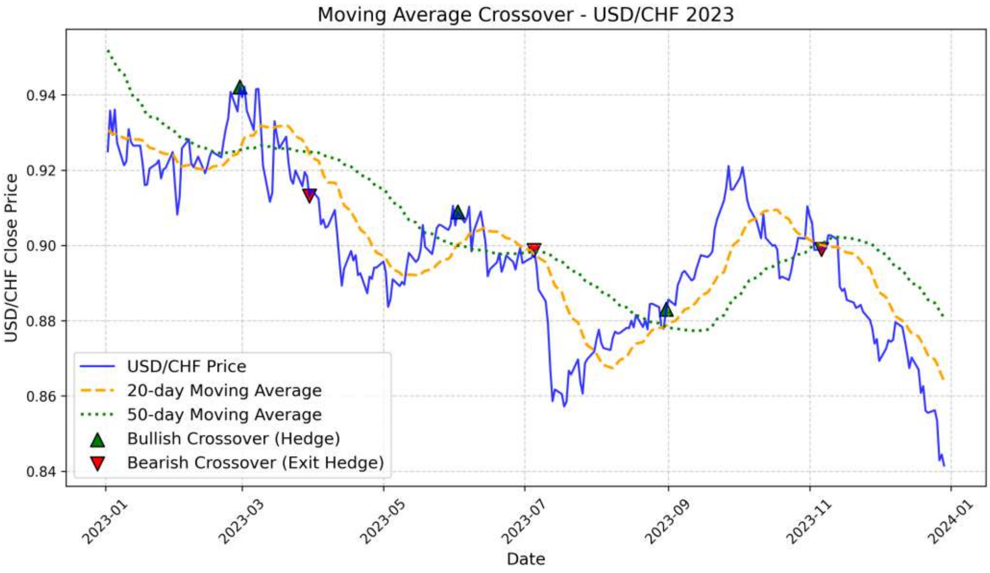

Figure 3.2: Moving Average Crossover (MAC) strategy applied to the USD/CHF exchange rate throughout the test period.

Figure 3.2 presents the **Moving Average Crossover (MAC)** strategy applied to the **USD/CHF exchange rate** during the **test period**. The chart shows and highlights the places where the long and short term moving average cross in one way or another. Bullish signals are marked in green (when the short crosses the long moving average). Bearish signals are marked in red (when the long crosses below the short moving average). The visualization is helpful to understand the core of this simple strategy.

3.2 Introduction to Reinforcement Learning

Reinforcement Learning (RL) is a type of machine learning where an agent learns by interacting with an environment to maximize a numerical reward [16]. It is different than supervised learning in a sense that it does not need exact labels for a model to be trained successfully. Instead, an agent must figure out what is the best strategy by interacting with the environment through trial and error.

One of the main challenges of RL is that the significance of an action and its consequences are not known immediately. Therefore, the agent needs to balance out with making good decisions that provide high reward right away as well as making decisions that will lead to higher rewards in the future. This idea of delayed rewards makes the RL setting more complicated and dynamic than other ML fields.

An RL setup consists of an agent and an environment. The agent is the actual decision maker which in each step observes the current state of the environment, and then based on it makes a decision. An environment is everything the agent interacts with. It responds after every action taken from an agent, by providing a new state and a reward. Based on the reward, the agent knows if the action was good or bad. Throughout time, the agent improves upon its strategy by interacting with the environment with a goal to maximize its total rewards.

Another important concept in RL is the **exploration-exploitation trade-off**. The agent must balance between:

- **Exploration** – trying new actions to discover better strategies.
- **Exploitation** – using what it has already learned to maximize rewards.

A good learning algorithm finds a balance between the two. It explores enough to gather concrete information about the environment and which actions bring good reward. Then, based on the research, it maximizes the reward. RL is widely used in areas like robotics, finance, gaming, etc.

The RL process can be visualized in Figure 3.3, which illustrates the interaction between an agent and its environment. In each time step i, the agent observes the current state

S_i, receives a reward R_i, and takes an action A_i. This influences the environment, transitioning it to a new state, and the process continues until a terminal state is reached or the episode ends [6].

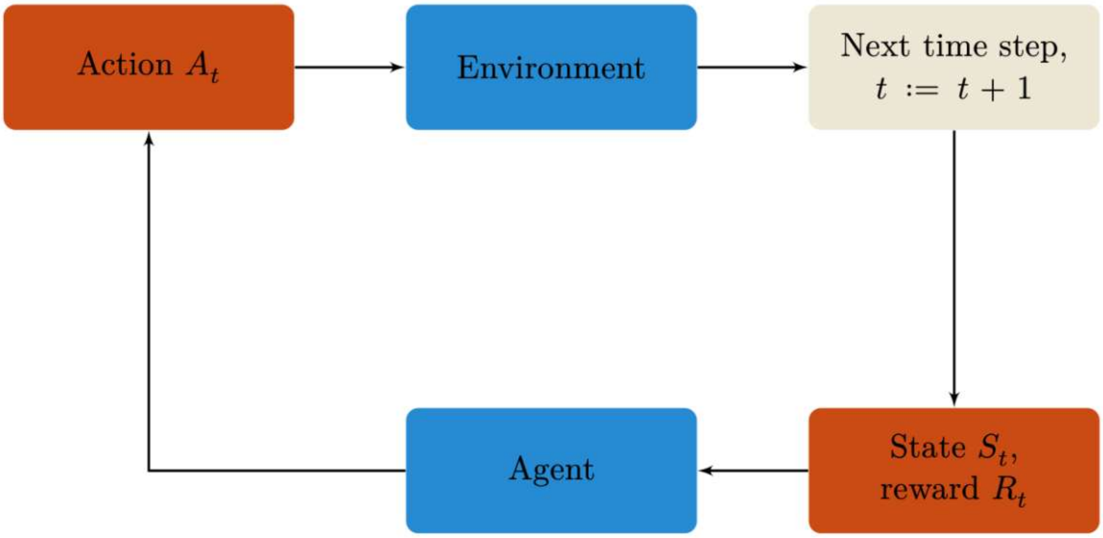

Figure 3.3: Interaction between an agent and an environment in RL [6].

The return

$$G_t := \sum_{k=t+1}^{T} \gamma^{k-(t+1)} R_k = R_{t+1} + \gamma R_{t+2} + \cdots \tag{3.4}$$

is the total sum of rewards received by the agent, discounted by a factor γ [6]. The goal of the agent is to maximize the expected return $E[G_t]$.

3.3 Deep Reinforcement Learning Models

3.3.1 Double Deep Q-Network (DDQN) Model

Deep Q-Networks (DQN) introduced deep learning into reinforcement learning. It allowed agents to learn optimal policies directly from high-dimensional state spaces [13]. In traditional Q-learning, the agent learns an action-value function $Q(s, a)$ that estimates the expected cumulative reward for taking action a in state s and following an optimal policy thereafter. The Q-values are updated using the empirical Bellman target, which is the approximation of the expected cumulative reward based on the current state, action, and the Q-values of subsequent states.

The Q-values are updated using the update rule

$$Q_{t+1}(s,a) := \begin{cases} (1-\alpha_t)Q_t(s_t,a_t) + \alpha_t \left(r_{t+1} + \gamma \max_a Q_t(s_{t+1},a) \right), & (s,a) = (s_t,a_t) \\ Q_t(s,a), & (s,a) \neq (s_t,a_t) \end{cases} \tag{3.5}$$

[6], where α_t is the learning rate, r_{t+1} is the reward received after taking action a in state s, and γ is the discount factor for future rewards. However, storing a Q-table for large state spaces is infeasible, leading to the development of **Deep Q-Networks (DQN)**, which approximate $Q(s,a)$ using a deep neural network. The DQN model replaces the traditional Q-table with a deep neural network $Q(s,a;\theta)$ and learns optimal action values through **experience replay** and a **target network** to stabilize training. θ represents the parameters (weights and biases) of the deep neural network used to approximate the Q-values in the DQN model.

Despite these improvements, DQN suffers from **overestimation bias** due to the **max operator** in action selection and evaluation. This overestimation leads to unstable learning and suboptimal policies [5].

Double Deep Q-Network (DDQN): An Improvement Over DQN

To address the limitations of DQN, **Double Deep Q-Networks (DDQN)** [5] introduce a modification that **decouples action selection from action evaluation**. Instead of using the same Q-network for both selecting and evaluating the best action, **DDQN maintains two separate value estimates**:

- The **online network** $Q(s,a;\theta)$ is used for action selection.
- The **target network** $Q(s,a;\theta^-)$ is used for action evaluation.

This distinction prevents overestimation bias and results in more accurate Q-value estimates. Mathematically, in **DQN**, the target Q-value is defined as [5]

$$Y_t^{\text{DQN}} := R_{t+1} + \gamma \max_a Q(S_{t+1},a;\theta^-). \tag{3.6}$$

In **DDQN**, the key modification is that the **online network** selects the best action, while the **target network** evaluates it, i.e., [5]

$$Y_t^{\text{DDQN}} := R_{t+1} + \gamma Q(S_{t+1}, \arg\max_a Q(S_{t+1},a;\theta), \theta^-). \tag{3.7}$$

This adjustment significantly reduces overoptimism in value estimation, leading to **more stable learning** and **better policy performance**.

By preventing overestimation of Q-values, **DDQN improves policy learning stability and performance** compared to standard DQN. It has been successfully applied to various reinforcement learning tasks, demonstrating **better convergence and more robust decision-making** in environments with complex state spaces [5].

Algorithm 1 Double Q-learning for calculating $Q \approx q^*$ and $\pi \approx \pi^*$ [6].

```
Initialization:
Choose learning rate $\alpha \in (0, 1]$
Choose $\epsilon > 0$
Initialize $Q_1[s, a] \in \mathbb{R}$ and $Q_2[s, a]$ arbitrarily for all $(s, a) \in \mathcal{S} \times \mathcal{A}(s)$, except that
the value of the terminal state is 0
Loop                                                                    // for all episodes
  initialize $s$
  while episode not finished do
    repeat                                                              // for all time steps
      Choose action $a$ from $s$ using an ($\epsilon$-greedy) policy derived from $Q := \frac{Q_1+Q_2}{2}$
      Take action $a$ and receive new state $s'$ and reward $r$
      if random number chosen uniformly in $[0, 1) < 1/2$ then
        $Q_1[s, a] := Q_1[s, a] + \alpha \left( r + \gamma Q_2 \left[ s', \arg\max_{a' \in \mathcal{A}(s')} Q_1[s', a'] \right] - Q_1[s, a] \right)$
      else
        $Q_2[s, a] := Q_2[s, a] + \alpha \left( r + \gamma Q_1 \left[ s', \arg\max_{a' \in \mathcal{A}(s')} Q_2[s', a'] \right] - Q_2[s, a] \right)$
      end if
      $s := s'$
  end while
```

Algorithm 1 [6] displays in pseudocode the learning process of the double Q-learning algorithm. To adjust this for the DRL setting, the Q-value functions Q_1 and Q_2 are approximated using deep neural networks instead of tables.

3.3.2 Proximal Policy Optimization (PPO) Model

Proximal Policy Optimization (PPO) is a reinforcement learning algorithm that improves policy gradient methods by balancing **sample efficiency, simplicity, and reliable performance** [15]. PPO is designed as an alternative to **Trust Region Policy Optimization (TRPO)**, maintaining its stability benefits while being **simpler to implement** and **more adaptable** to different problems.

Policy Gradient Methods

Policy gradient methods optimize a stochastic policy $\pi_\theta(a|s)$ using **stochastic gradient ascent** on the expected return. The gradient estimator is defined as [15]

$$g := \mathbb{E}_t \left[\nabla_\theta \log \pi_\theta(a_t|s_t) A_t \right], \tag{3.8}$$

where A_t is an estimator of the **advantage function**, measuring how much better an action is compared to the average action in that state. The objective function is defined as [15]

$$L_{\text{PG}}(\theta) := \mathbb{E}_t \left[\log \pi_\theta(a_t|s_t) A_t \right]. \tag{3.9}$$

However, **standard policy gradient methods suffer from large, unstable policy updates**, leading to poor sample efficiency and performance degradation. TRPO addressed this issue but introduced computational complexity due to **second-order optimization constraints**.

PPO: A Simpler Alternative to TRPO

PPO introduces a new objective function with **clipped probability ratios**, which prevents overly large policy updates and ensures stable learning. The policy update is constrained by clipping the **probability ratio**, i.e. [15]

$$r_t(\theta) := \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, \tag{3.10}$$

where π_θ is the new policy and $\pi_{\theta_{\text{old}}}$ is the previous policy before the update.

The Clipped Surrogate Objective

PPO optimizes the **clipped surrogate objective** [15]

$$L_{\text{CLIP}}(\theta) := \mathbb{E}_t \left[\min\left(r_t(\theta) A_t, \text{clip}\left(r_t(\theta), 1 - \epsilon, 1 + \epsilon \right) A_t \right) \right], \tag{3.11}$$

This objective ensures that updates do not excessively change the policy, as the **clipping mechanism** prevents r_t from moving too far from 1. If an update causes a large change in r_t, the clipped term prevents further increases, stabilizing learning.

Algorithm 2 Proximal Policy Optimization (PPO) for policy optimization [6]

```
loop
for all actors from 1 to $N$ do
    run policy $\pi_{\theta_{\text{old}}}$ for $T$ time steps
    compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$ using GAE
end for
optimize surrogate objective $J_{\text{CLIP+VF+S}}$ w.r.t. $\theta$
    using $K$ epochs and minibatch size $M \leq NT$
$\theta_{\text{old}} := \theta$
end loop
```

The key features of PPO are that it enables multiple updates per batch of data, which improves sample efficiency. It also prevents drastic changes that could mess up with the learning. It does this by controlling the policy updates with clipping the objective function. PPO also uses the first-order optimization which makes it simpler and faster than TRPO.

PPO has demonstrated **state-of-the-art performance** in various reinforcement learning benchmarks, including **robotic control (MuJoCo) and Atari environments** [15]. Its balance between efficiency, stability, and ease of implementation makes it one of the most widely used RL algorithms today.

3.4 LSTM as Base Policy

Long Short-Term Memory (LSTM) networks were introduced by Hochreiter and Schmidhuber to address the vanishing gradient problem in traditional Recurrent Neural Networks (RNNs) [7]. Unlike standard RNNs, which struggle to maintain long-term dependencies, LSTM networks incorporate a **memory cell** that allows information to persist over long sequences, making them highly effective for sequential decision-making tasks.

LSTM Architecture

An LSTM cell consists of three primary gates that regulate information flow:

- **A forget gate** determines which information from the previous state should be discarded.
- **An input gate** controls what new information should be stored in the cell state.
- **An output gate** regulates how much of the current cell state should be passed as output.

Mathematically, the LSTM cell updates its states using the following equations:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), &(7)\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f), &(8)\\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), &(9)\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o), &(10)\\
h_t &= o_t \odot \tanh(c_t), &(11)
\end{aligned}
$$

where σ is the logistic sigmoid function, and i, f, o, and c denote the *input gate*, *forget gate*, *output gate*, and *cell input activation* vectors, all of which are the same size as the hidden vector h. Here, $\odot$ denotes elementwise (Hadamard) multiplication. The weight matrix subscripts have the standard meaning: for instance, W_{ih} denotes the matrix from the previous hidden state h_{t-1} to the input gate i_t, while W_{ic} and similar terms represent peephole connections from the previous cell state. The biases b_i, b_f, b_c, and b_o are omitted in most simplified descriptions [4].

LSTM in Reinforcement Learning

LSTM networks are particularly useful in **reinforcement learning** when dealing with **partially observable environments**. Instead of relying solely on immediate observations, an LSTM-based policy can retain memory of past states, enabling the agent to make better-informed decisions.

In this research, LSTM serves as the **base policy** for the reinforcement learning model, allowing the agent to capture long-term dependencies in market conditions. By maintaining a history of past prices and hedging decisions, the model can learn **temporal patterns** that are crucial for optimizing FX hedging strategies.

LSTM-based policies have shown strong performance in various sequential decision-making tasks, including algorithmic trading, portfolio optimization, and risk management. By leveraging LSTM as the base policy in **DRL**, this research aims to improve the stability and adaptability of FX hedging strategies.

3.5 Evaluation metrics

3.5.1 ML metrics

Temporal Difference (TD) error is the difference between the predicted Q-value and the actual Q-value [13]. The TD error

$$\text{TD Error} := r + \gamma \max Q(s', a') - Q(s, a) \tag{3.12}$$

will be the main loss function of the DDQN model. A TD error moving toward zero over time shows effective learning.

The policy loss

$$L_{\text{CLIP}}(\theta) := \mathbb{E}_t \left[\min\left(r_t A_t, \text{clip}\left(r_t, 1-\epsilon, 1+\epsilon\right) A_t\right)\right] \tag{3.13}$$

in PPO clips the probability ratio r_t, and in doing so prevents larger updates. This always makes the new policy relatively similar to the old policy, and there are no excessive jumps in policies, which stabilizes training [15]. The policy loss will be the main loss function of the PPO model.

The Cumulative Reward metric

$$\text{Cumulative Reward} := \sum_{t=1}^{T} r_t \tag{3.14}$$

captures the total reward throughout the testing. As the reward is a representation of how much money was earned, we can look at cumulative reward as cumulative earnings throughout the period.

3.5.2 Financial metrics

The **Sharpe Ratio**

$$\text{Sharpe Ratio} := \frac{R_p - R_f}{\sigma_p} \tag{3.15}$$

reveals the average investment return, minus the risk-free rate of return, divided by the standard deviation of returns for the investment [12].

The **Maximum Drawdown** (MDD)

$$\text{Maximum Drawdown} := \max(P_{\text{peak}} - P_{\text{trough}}) \tag{3.16}$$

measures the maximum fall in the value of the investment, as given by the difference between the value of the lowest trough and that of the highest peak before the trough. MDD is calculated over a long time period when the value of an asset or an investment has gone through several boom-bust cycles [11].

Volatility

$$\text{Volatility} := \sqrt{\frac{1}{N}\sum_{i=1}^{N}(R_i - \bar{R})^2} \tag{3.17}$$

measures the dispersion of returns around its mean. Higher volatility indicates higher risk.

The **Beta**

$$\beta := \frac{\text{Cov}(R_p, R_b)}{\text{Var}(R_b)} \tag{3.18}$$

of an investment security is a measurement of its volatility of returns relative to the entire market. A company with a higher beta has greater risk and also greater expected returns [10].

The **Return of the Strategy**

$$\text{Return} := \frac{P_{\text{final}} - P_{\text{initial}}}{P_{\text{initial}}} \tag{3.19}$$

measures how much the strategy has earned throughout the entire test period:

The **Annualized Return**

$$\text{Return}_{\text{p.a.}} := (1 + \text{Return})^{\frac{1}{T}} - 1 \tag{3.20}$$

measures the compounded yearly return of the strategy. It provides a clean metric on what the strategy earns per year.

CHAPTER 4

Results

4.1 Dataset, Features, and Preprocessing

The currency pair used in this research is USDCHF (United States Dollars / Swiss Francs). Therefore, the dataset consisted of the USDCHF close price and a couple of technical indicators which were derived from the USDCHF close price. The three technical indicators are: Relative Strength Index (RSI), Moving Average Convergence Divergence (MACD), and Exponential Moving Average (EMA). These features are the most commonly used simple technical indicators to identify trends and momentum.

The dataset is divided into a training set from 1980 to 2022 with 10,903 observations and a test set from 2022 to 2024 with 805 observations. The training data is used to develop and optimize the reinforcement learning models, while the test data is used to evaluate its performance. Figure 4.1 shows the exchange rate of USD/CHF throughout the test period.

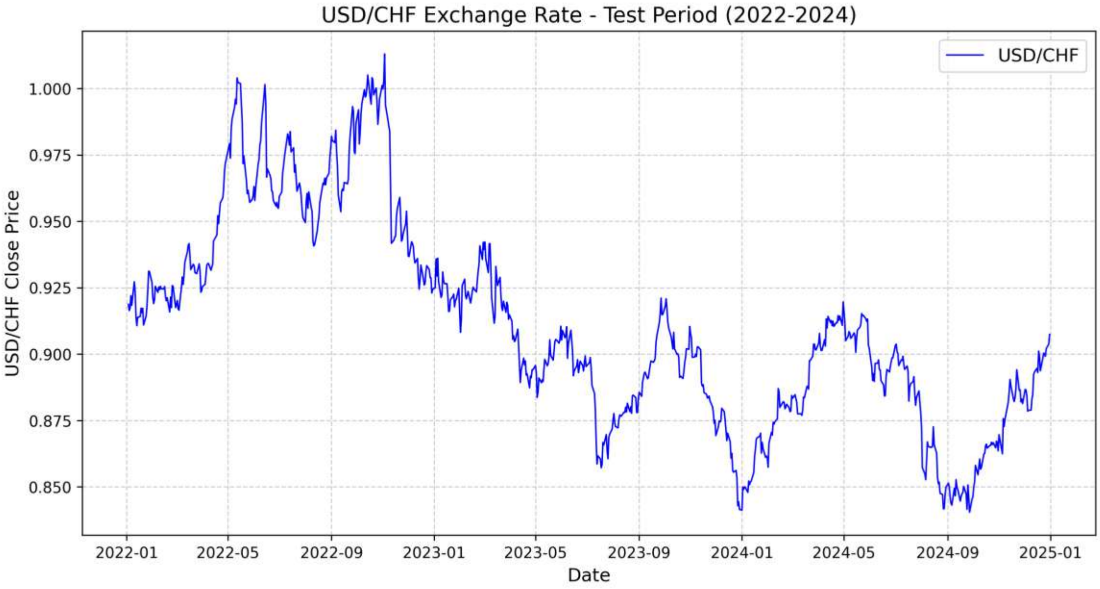

Figure 4.1: USD/CHF Exchange rate throughout the test period, sourced from [18].

Features

The dataset consists of USDCHF close prices, RSI (Relative Strength Index), EMA (Exponential Moving Average), and the MACD (Moving Average Convergence Divergence). The data provider for the USDCHF closing price was Trading Economics [18]. All the other features were created using the respective mathematical formulas of technical indicators. The basis for their calculations were the close prices of USDCHF.

Relative Strength Index (RSI) is already explained in Section 3.1.1. It is a momentum oscillator used to identify buy or sell signals in a market. We have used it as a raw numerical feature in the dataset. The RSI was calculated over a 14-day rolling period. It ranges from 0-100 and is provided as input to the model.

Exponential Moving Average (EMA) is a technical moving average indicator that assigns greater weight to recent price data. This makes it more sensitive to price fluctuations compared to **Simple Moving Average (SMA)** [2]. The EMA is calculated using the recursive formula

$$EMA_t := \alpha \cdot P_t + (1 - \alpha) \cdot EMA_{t-1}, \tag{4.1}$$

where:

- P_t is the price at time t.
- α is the smoothing factor, calculated as $\frac{2}{n+1}$, where n is the number of periods.
- EMA_{t-1} is the previous EMA value.

EMA can react faster than SMA, as it assigns more weight to recent prices. This makes it more effective for capturing trends and shifts in some scenarios. The Exponential Moving Average (EMA) is useful for understanding market trends. When the EMA is rising, it means prices are going up (uptrend), and when it's falling, it suggests prices are going down (downtrend). Traders also use crossover strategies, where a short-term EMA (like a 9-day EMA) crossing above a long-term EMA (like a 50-day EMA) signals a possible buying opportunity, while crossing below suggests selling. Additionally, the steepness of the EMA helps measure how strong a price movement is, making it a useful tool for spotting market momentum. The 10-day EMA was used in the research as input to the model.

Moving Average Convergence Divergence (MACD) is a momentum indicator that measures the relationship between two moving averages of an asset's price [3]. It consists of three components. The first is the MACD line

$$\text{MACD line} := \text{EMA}_{12}(P_t) - \text{EMA}_{26}(P_t), \tag{4.2}$$

which is the difference between the **12-day** EMA and the **26-day** EMA. This line represents the core momentum signal.

The second is the signal line

$$\text{Signal Line} := \text{EMA}_9(\text{MACD line}), \tag{4.3}$$

which is the 9-day EMA of the MACD line. The third is the MACD histogram

$$\text{MACD Histogram} := \text{MACD line} - \text{Signal Line} \tag{4.4}$$

which is the difference between the MACD line and the signal line.

The MACD is widely used as one of the most reliable technical indicators for momentum-based trading strategies [3]. The buy signal is derived when the MACD line crosses above the signal line, while the sell signals is derived when the MACD line crosses below the signal line. We define

$$\text{Final MACD value} := \begin{cases} 1, & \text{MACD line} > \text{Signal Line (Buy Signal)}, \\ -1, & \text{MACD line} < \text{Signal Line (Sell Signal)}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.5}$$

Preprocessing

The financial time series requires intense preprocessing to make it possible for a model to accurately forecast. As the quantitative finance methods were outside of the scope of the thesis and the masters program, the preprocessing consisted of the simple data science techniques. **A 5-day rolling moving-average** filter was applied to each feature. This reduced the short term fluctuations. **Min-Max Scaling** was applied to normalize the features between **0 and 1**. This is a crucial step in machine learning, as the models learn better when the features are on the same scale.

4.2 Reinforcement-Learning Environment

Set of actions and states will be very similar to a classical trading problem in reinforcement learning. In actions the classical Buy is replaced by Enter Hedge, and the Sell is replaced by the Exit Hedge. In states, the Long is replaced by Hedged, and the Short is replaced by Protected. This is done to accommodate for the FX component of the research.

Actions:

- **Enter Hedge** – The agent opens a hedge position, securing the current exchange rate and becoming exposed to the interest rate differential.
- **Exit Hedge** – The agent exits the hedge position and moves to a protected state, allowing the exchange rate to fluctuate freely.

States:

- **Hedged** – The agent maintains a hedge position and earns or pays the interest rate differential between the USD and CHF.
- **Protected** – The agent does not hedge, generating profit by benefiting from exchange rate fluctuations.

When in the **Hedged** state, the agent is exposed to the **interest rate differential** between the United States and Switzerland, which influences the cost of maintaining the hedge. Conversely, in the **Protected** state, the agent profits by shorting the exchange rate movement.

This structure provides a **realistic framework for companies with international subsidiaries** receiving payments in foreign currencies. By optimizing when to hedge (locking in the exchange rate) and when to remain unhedged (allowing the rate to fluctuate), the model helps businesses minimize risk and maximize profits.

Interest Rate Differential (IR Diff)

IR Diff represents the difference between the **USD** and **CHF** interest rates, calculated using historical central bank data. This is very important in hedging decisions, as it influences the **cost or benefit of maintaining a hedge position**, making it expensive or profitable based on interest rate differentials. It also influences the reward structure in the RL model, and it affects the long-term profitability of strategies. The **IR Diff** was sourced from Trading Economics [18]. If the USD interest rate is higher than the CHF interest rate, the hedger profits from the difference in rates because they borrow the lower interest rate currency CHF, and invest in the higher interest rate currency USD. It is helpful to think of it as an opportunity cost. The hedger has chosen not to keep their money in CHF where they could earn 2% from government bonds, but to invest it in the USD where they can earn 5% from government bonds.

Reward Function

At each time step t, the reward function is influenced by whether the agent executes a trade (switching between `Hedged` and `Protected`), and whether the episode reaches its final time step (truncation). When a trade is executed or the episode ends, the reward is computed as

$$r_t := \begin{cases} 100 \cdot \left(\overline{IR}_t\right) \cdot \dfrac{d_t}{252} \cdot (1 - \text{fee}_{\text{bid}}) & \text{if in } \texttt{Hedged} \text{ state,} \\ 100 \cdot \ln\left(\dfrac{P_{\text{last\_trade}}}{P_t}\right) \cdot (1 - \text{fee}_{\text{ask}}) & \text{if in } \texttt{Protected} \text{ state,} \\ 0, & \text{otherwise,} \end{cases} \tag{4.6}$$

where:

- $P_{\text{last\_trade}}$ is the price at the last executed trade.
- P_t is the current price.
- d_t is the number of business days (or ticks) spent in the `Hedged` position.
- $\overline{IR}_t = \frac{1}{d_t} \sum_{\tau = t_{\text{last\_trade}}}^{t-1} \text{IR Diff}(\tau)$ represents the **average interest rate differential** over the period $[t_{\text{last\_trade}}, t)$.
- fee_{bid} and fee_{ask} account for transaction-related costs, including spread, brokerage fees, and slippage when entering or exiting positions.

If no trade is executed at time t and the episode continues, the reward is simply 0. Thus, the reward is realized only when the agent transitions between states, or if the episode reaches termination (truncation).

This reward function encourages the agent to strategically time its hedging decisions based on **interest rate fluctuations, exchange rate movements, and transaction costs**.

4.3 Practical Application of the Strategy

4.3.1 Who Benefits?

This hedging strategy benefits all who actively manage currency risk. Those are companies with foreign subsidiaries who must decide when to convert their earnings in the foreign currency. It also benefits, exporters, institutional traders. They use such strategies to avoid exchange rate fluctuations.

In this research, the analysis was looked from the perspective of a large international company. The company receives payment in a foreign currency as it has a subsidiary in another country.

4.3.2 Example Use Case

AlpineTech, a U.S.-based manufacturer of high-precision industrial equipment, operates a subsidiary in Switzerland. It regularly receives money in Swiss Francs (CHF) from its Swiss sales. AlpineTech reports its financials in U.S. Dollars (USD), and it is crucial that it converts CHF to USD at the correct times. And, it must have a good strategy on when to lock the exchange rate.

Two possible positions are explained down below:

- **Position–Hedged**: The client enters a hedging position when the strategy suggests that the USD/CHF exchange rate will increase. This means the CHF will weaken against the USD. By entering into a forward contract, the current exchange rate is locked. Future CHF conversion are secured from losing their value.
- **Position–Protected**: The client exits the hedging position and remains unhedged when the strategy suggests that the USD/CHF exchange rate will decrease. This means the CHF will gain value in comparison to the USD.

Any company which has subsidiaries in multiple countries face this problem. For the purposes of the research, we have focused on managing USDCHF currency risk to optimize hedging.

4.4 Financial Costs and Trading Fees

In FX trading, there are multiple fees components that affect every transaction and the profitability of the strategy. These are spreads, commissions, slippage, and interest rate differentials. Below is a general breakdown of the main fees involved in USD/CHF trading. The issue with most research papers is that they assume a fixed percentage fee model. While this assumption gets the results close to reality, it is never really the reality. In this thesis, a dynamic fee model will be set up to make things as accurate as possible.

4.4.1 General Breakdown of FX Hedging Fees

The **spread** is the difference between the buying and selling prices of a currency pair. It is expressed in pips which is the smallest price movement unit, or 0.0001. The USDCHF is a highly liquid and stable currency pair, so we estimate the spread to be 1 pip.

Banks and brokers require fees to facilitate trades. This fee covers the execution of the transaction and is around 1 pip as well.

Holding a hedge position through time incurs **interest rate differential** costs. The USD/CHF currency pair is influenced by the difference between the U.S. Federal Reserve and Swiss National Bank interest rates. If the USD interest rate is higher than the CHF rate, traders will earn the difference when holding CHF. If CHF rates are higher, they pay the differential. This components is incorporated inside the reward function, but it's also technically a fee.

Slippage occurs when a trade is executed at a different price than expected, often due to high market volatility. This introduces an unpredictable fee that depends on market conditions. For the sake of simplicity, we consider this to be 1 pip, but in real life this would depend on the volatility and is unpredictable.

4.4.2 Example of Currency Hedging with USD/CHF–Fee Structure

For this example, an imaginary US-based company is expecting revenue in Swiss Francs throughout the next 3 months. The CHF is expected to lose value against the USD, so the company would like to hedge and enter into a forward contract. The breakdown of fees is as follows:

- **Trade Fee on Exiting the CHF Hedge–0.02%**: When converting CHF to USD (selling CHF), we incur a total transaction cost of **0.02%** of the trade volume. This includes both the **spread** and the **bank/broker fee**. For a **1 million CHF** trade, this results in a total cost of **200 CHF**.

- **Trade Fee on Entering the CHF Hedge–0.03%**: When entering the hedge, the total transaction cost is **0.03%** of the trade volume. This includes the **spread**, the **bank/broker fee**, and **slippage**. For a **1 million CHF** trade, this amounts to a total cost of **300 CHF**.

Table 4.1: Example FX Transaction and Associated Costs

Description	**Amount (CHF)**
Imaginary transaction volume:	1,000,000
Trade fee on selling CHF (0.02%):	200
Trade fee on buying CHF (0.03%):	300
Total Cost:	**500**

The total cost of the hedge amounts to approximately **500 CHF**, which represents **0.05% of the total transaction volume**. This cost arises from the combined fees associated with entering and exiting the hedge, including the **spread, bank/broker fees, and slippage**. While this percentage may seem minimal, it can have a significant impact over multiple transactions, especially for companies engaging in frequent hedging activities or managing large currency exposures.

4.5 Modeling Process

The RL models were trained with a batch size of 1008 spread across six CPU cores in parallel. The lookback period was 25 observations, or 1 month of data. Meaning, each state was of shape (25,4). The training was done in 300,000 timesteps, while the training set had a size of 10,000 observations. This means there were 30 epochs in the training phase. Each episode lasted for 252 timesteps/observations (1 year).

Hyperparameter Tuning

We have tuned the parameters of the RL models using a random search based approach. We defined a range of values, and picked out multiple random combinations. For both **DDQN** and **PPO**, the learning rate and the discount factor (γ) were tuned. The learning rate was tested within the range of $\mathbf{10^{-5}}$ to $\mathbf{10^{-3}}$, while the discount factor was explored between **0.9** and **0.99**. However, the other hyperparameters were tuned differently for each algorithm. For **DDQN**, the **epsilon decay timesteps**, which control the gradual reduction of the exploration strategy (epsilon-greedy), were adjusted within a range of $\mathbf{50,000}$ to $\mathbf{200,000}$. For **PPO**, the **clipping parameter**, which stabilizes training by constraining policy updates, was tested between **0.1** and **0.4**.

For each model, **ten different hyperparameter configurations** were evaluated, including a **default baseline configuration** and nine randomly sampled variations from the predefined ranges.

Model Selection and Evaluation

The best-performing hyperparameter configurations were selected based on the learning progress. The learning progress plots of different configurations were analyzed to find the ones that exhibited stable convergence over iterations. Cumulative reward was also taken into account in the training phase.

The selected models were further validated by running **five consecutive training sessions** with the best hyperparameter settings. The chosen hyperparameter configurations are as follows:

- **Best DDQN Configuration:**
  - **Learning Rate (LR)**: 0.000240
  - **Discount Factor** (γ): 0.908433
  - **Epsilon Decay Timesteps**: $57,811$
  - **Mean Reward Achieved**: 49.123
  - **Mean Loss**: 0.040886
- **Best PPO Configuration:**
  - **Learning Rate (LR)**: 0.000428
  - **Discount Factor** (γ): 0.907824
  - **Clip Parameter**: 0.367654
  - **Mean Reward Achieved**: 61.967
  - **Mean Loss**: -0.025326

The best configurations for both PPO and DDQN were picked by analyzing the learning curves, rewards, and stability of learning over multiple training runs. The PPO showed a higher mean reward than DDQN, which would mean that it came up with a more effective policy for the environment.

4.6 Training Results and Learning Progress

It is important to realize that the learning progress plots will not be similar to those seen in simpler DRL applications. Financial markets are inherently non-stationary which means that market conditions, price, and influences change through time. As a result, we cannot expect a typical convergence plot in which the loss gradually approaches zero. The goal is to see a slow stabilization around zero. This indicates better adapting capabilities of the model over time to market conditions, while maintaining consistency in learning.

In Figure 4.2, the TD error begins to stabilize around the 30th to 50th training batch. This is approximately the area where the model shifts from exploration to exploitation in our best model configuration for DDQN. After this, the model slightly diverges from the zero TD error range, showing fluctuations in learning. Finally, it stabilizes once again after approximately 250 training batches, and stayed consistent until the end.

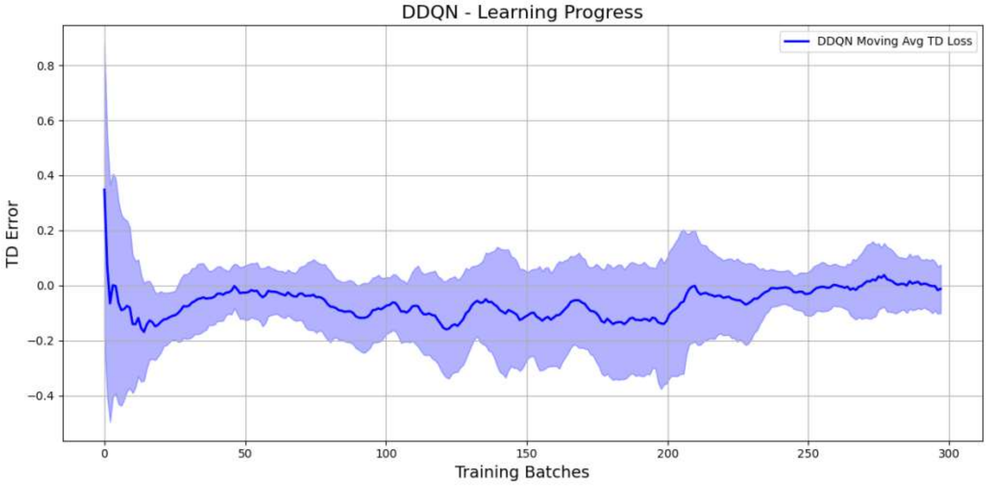

Figure 4.2: DDQN Learning Progress.

In Figure 4.3, we observe the learning progress of the best-performing PPO model configuration. Unlike DDQN, which relies on temporal difference (TD) error minimization, PPO's loss function is based on policy gradient updates with a clipped surrogate objective. Due to these fundamental differences, direct comparisons between DDQN and PPO learning curves are not entirely meaningful, as the loss values and their respective interpretations vary.

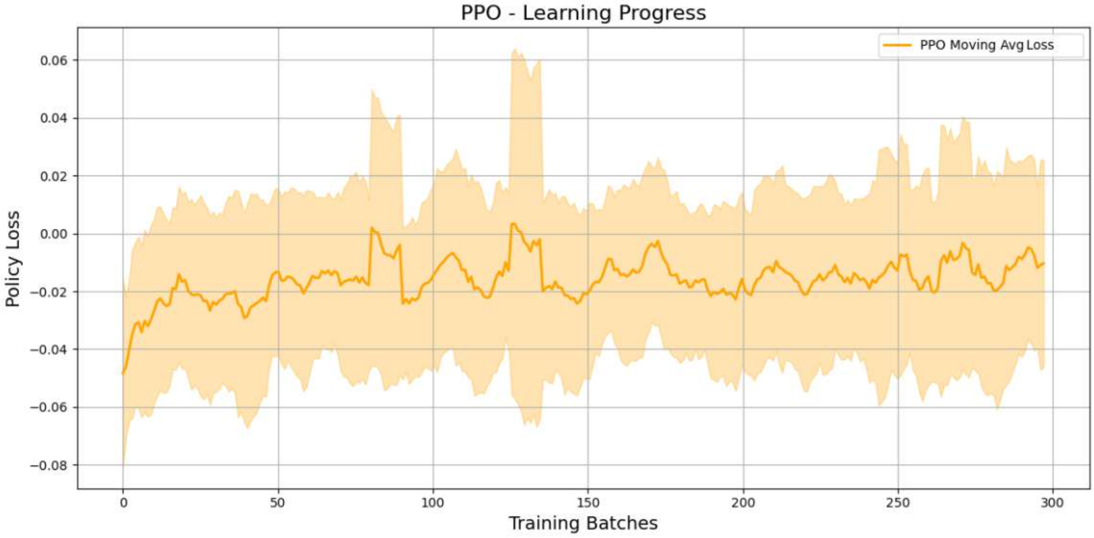

Figure 4.3: PPO Learning Progress.

Still, some qualitative observations can be made. DDQN shows a relatively stable convergence pattern. The TD error consistently oscillates around zero after an initial exploration phase. PPO is more volatile in its loss values over the training period. Despite this instability, final performance evaluation matters more than learning progress alone. While DDQN appears to show a more structured stabilization process, the actual effectiveness of each model will be determined by their ability to generalize the hedging strategy in the test environment.

In Figure 4.4, we observe the mean training reward of the batches throughout the training phase for both PPO (orange) and DDQN (blue).

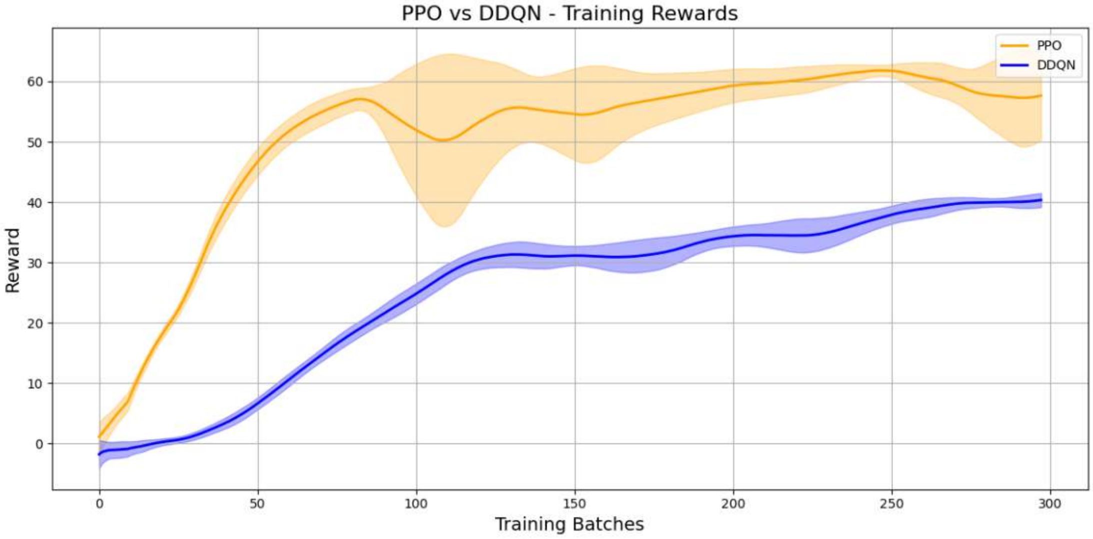

Figure 4.4: Training rewards.

PPO starts off with strong improvements, rewards increase quickly within the first 50 batches. The variance is a lot bigger, meaning PPO fluctuates a lot during training. It reaches a peak around batch 100, and after that it experience some instability before stabilizing near batch 200.

DDQN has a more gradual learning curve. The rewards increase consistently with less variance. The PPO achieves higher peak rewards, but the DDQN shows greater stability.

The training reward plot shows the differences in models. PPO is more adaptive and explores aggressively, but it can be unstable. DDQN provides more structured learning but may take longer to reach optimal performance. The final evaluation will determine which model generalizes better on the test set.

4.7 Test Set Performance Analysis

In Figure 4.5, we observe the cumulative returns of each strategy throughout the test period. The comparison includes the MAC strategy (blue), RSI strategy (orange), DDQN (green), and PPO (red).

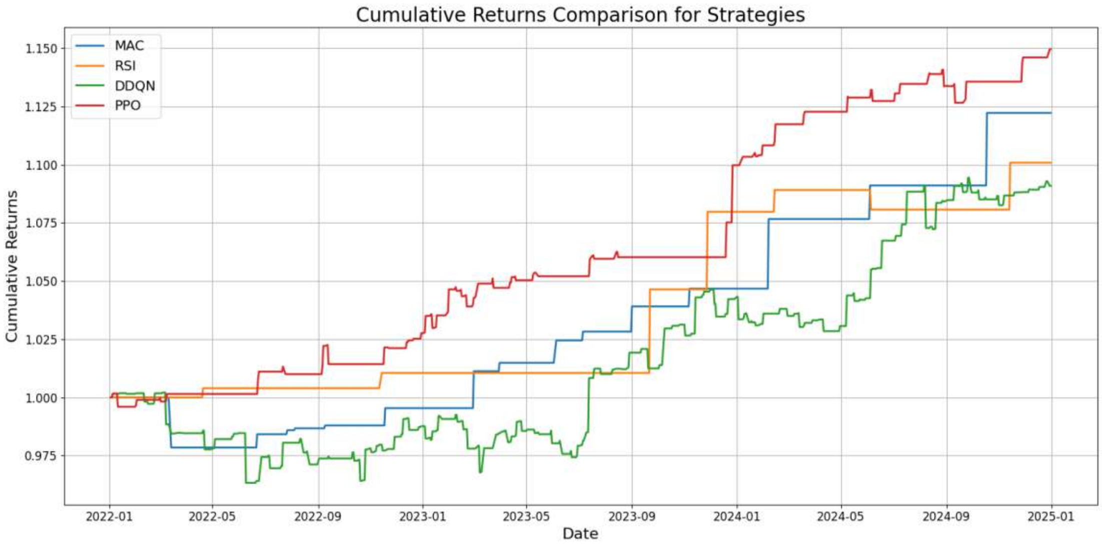

Figure 4.5: Cumulative earnings.

PPO comes out on top over the other strategies. It shows a strong upward trend in cumulative rewards. There are plenty of successful trades with high returns in PPO, and the step-like increase in rewards looks promising.

RSI and MAC perform similarly, with slow growth. These strategies look like they could provide solid returns with little volatility and risk.

DDQN lagged in the beginning, but it caught up over time. It showed higher volatility than PPO.

The Figure 4.5 clearly shows that both **PPO** and **DDQN** execute significantly more trades compared to the baseline strategies. The number of trades or executions are always important to analyze in any financial experiment. Increased transactions could be considered good, as a higher number of correct trades would help the case of a trading model to be good. But, in the context of FX hedging, the objective is to make fewer, more high quality trades with longer periods. This optimizes risk management and cost efficiency.

PPO earned the most amount of money among all the strategies. It showed its effectiveness as a policy-based RL approach. DDQN started off slow, but stabilized half way throughout the test period. RSI and MAC strategies, showed that they can compete with advanced ML models. Simple technical indicators are a lot cheaper, and if they produce consistent slower returns, they could be good for risk management.

The performance metrics in Table 4.1 provide a clear picture of how each strategy performed in the test period. PPO achieved the highest total and annualized return (14.96% and 4.76%), outperforming all other strategies. The MAC strategy followed closely with a total return of 12.23% and an annualized return of 3.92%. The RSI-based strategy had slightly lower returns (10.09% total, 3.25% annualized), while DDQN had the lowest return (9.09% total, 2.94% annualized).

Metric	**MAC**	**RSI**	**DDQN**	**PPO**
Return Total (%)	12.23	10.09	9.09	14.96
Return p.a. (%)	3.92	3.25	2.94	4.76
Sharpe Ratio	0.09	0.03	0.00	0.20
Maximum Drawdown (%)	2.15	0.77	3.87	1.25
Yearly Volatility (%)	2.97	2.92	3.47	2.48
Beta	0.02	0.02	0.06	0.03

Table 4.2: Performance Metrics for Different Strategies. Best values are highlighted in gray.

PPO also had the highest Sharpe Ratio (0.20). The MAC strategy came second with (0.09). RSI had a near-zero Sharpe Ratio (0.03), and DDQN had 0.00. DDQN and RSI failed to provide any added benefit over the risk the strategy took on itself. PPO came the best in this metric as well.

The maximum drawdown metric shows the largest losses from peak to trough during the test period. RSI had the smallest drawdown (0.77%), followed by PPO with 1.25%. MAC had a slightly higher drawdown of 2.15%, while DDQN had the largest drawdown (3.87%). This suggests that DDQN lacked stability and struggled to manage risks.

In terms of volatility, PPO exhibited the lowest yearly volatility (2.48%), reinforcing its great performance throughout the test period. RSI and MAC had similar levels of volatility (2.92% and 2.97%, respectively). DDQN showed the highest volatility (3.47%), confirming that its returns fluctuated significantly.

Finally, the beta values show that all strategies had low market sensitivity, meaning their returns were largely independent of broader market trends.

In conclusion, PPO was by far the best strategy when looking at all the metrics. It combined high returns with relatively low risk. MAC also performed surprisingly well, and showed that for a simple strategy, it can be good. RSI and DDQN provided a close return performance, but when looking at risk adjusted returns, volatility, and drawdown, they do not seem good.

CHAPTER 5

Discussion

5.1 Interpretation of Results

PPO stands out with the highest total and annual returns (14.96% and 4.76%), the best Sharpe ratio (0.20), and a low maximum drawdown (1.25%). It strikes the right balance between making money and keeping risks under control. The MAC strategy is surprisingly good too: it delivers good returns (12.23% total, 3.92% annual) with moderate volatility. RSI and DDQN lag behind on risk-adjusted measures. RSI hardly boosts returns relative to risk, and it ends up too conservative. DDQN, despite many trades, has the highest volatility and drawdown, making it unstable. Overall, PPO is the clear winner for hedging, while MAC is a good low-cost fallback.

Despite the PPO coming out on top, the DRL models have their weaknesses and this research highlighted them. They are sensitive to hyperparameters, and their performance is highly dependent on careful tuning, which makes them difficult to generalize across different market conditions. There is also high variance in learning, as PPO showed fluctuating learning curves and DDQN struggled with convergence, leading to inconsistent performance. Finally, overtrading is a concern, as both DRL models executed significantly more trades than the baseline strategies. While this indicates high responsiveness, it does not align with the goals of FX hedging.

5.2 Limitations of the Study

Market Fees Assumptions: The research incorporated all the trading fees which would be present in the real-world scenario. We have made it as dynamic and realistic as possible. Still, we have to acknowledge that until tested in real life on new data, with all the systems connected together, we are unsure of the actual profit. This is the reality of financial experiments when backtesting.

Feature Selection Constraints: The research used simple technical indicators like RSI, MACD, EMA as features. They are useful for short-term trading strategies. But, in FX hedging there is a need to rely on long-term macroeconomic factors. Incorporating these features could improve the model to make long term stable transactions.

Lack of Live Testing: Again, the strategies were evaluated in a backtest on historical data. This missed market impact, execution delays, and weird slippage situations. A live trading environment brings a lot of other factors, which we could not anticipate. The results provide insights, but further validation live would be helpful.

Single Currency Pair Analysis: The study focused on **USD/CHF**. Performance on other currency pairs remains an open question.

These limitations suggest that further refinements are needed before deploying DRL-based hedging strategies in real-world financial systems.

5.3 Future Research Directions

Incorporating Macroeconomic Indicators: Variables such as **interest rate projections, inflation data, and geopolitical risk scores** could enhance the model's predictive capabilities. Additionally, adopting a more rigorous approach to preprocessing input features, as commonly practiced in quantitative finance, could further improve model performance. From my experience, standard data science and machine learning techniques alone are insufficient for financial datasets unless combined with a deep understanding of proper data preprocessing methods. For future research, I recommend leveraging the insights from [19] to enhance feature engineering and data preparation.

Extending to Other Currency Pairs: The model should be tested on additional FX pairs such as **EUR/USD, GBP/JPY, and emerging market currencies** to assess its generalizability.

Exploring Hybrid Approaches: A potential improvement would be to combine **DRL with rule-based filters**, where traditional indicators act as constraints to prevent excessive trading.

Deploying DRL in Live Trading: Future research should implement these strategies in **real-time trading environments** to assess their feasibility

By exploring these directions, the application of DRL in FX hedging can be further refined and made more robust for real-world financial decision-making.

CHAPTER 6

Conclusion

6.1 Summary of Findings

This thesis explored the application of DRL for FX hedging. It introduced a realistic reward function with the Interest Rate differential incorporated. It compared two reinforcement learning algorithms with two traditional baseline strategies. It has presented a realistic fees structure, and comprehensive metrics system. We found that PPO had the best overall performance, with the highest returns, lowest drawdowns, and strongest risk-adjusted metrics. The MAC strategy also performed well, with consistent returns, and acceptable risk adjusted metrics. In contrast, RSI and DDQN did not improve much on risk-adjusted results: RSI was overly cautious and DDQN produced too much volatility. These findings show that policy-based RL might be an effective tool for FX hedging, and that simple moving average crossovers remain a viable alternative. Furthermore, the study emphasized the importance of feature selection, showing that while technical indicators such as RSI, MACD, and EMA were useful, incorporating macroeconomic indicators could enhance model performance.

6.2 Final Thoughts on DRL for FX Hedging

The findings of this research suggest that DRL has potential as an FX hedging tool, particularly in its ability to dynamically adjust to market conditions. However, its superiority over traditional strategies remains marginal, and its practical implementation requires careful calibration of trading frequency and cost considerations. While PPO demonstrated strong adaptability, traditional approaches like MAC still provided competitive results with fewer trades and lower complexity. Moreover, the agents in DRL were not able to stick to one position for longer periods which made them execute too many transactions. This would not work in a real life setting, and is an issue that has to be solved among other things in order to deploy DRL for FX hedging.

6.3 Potential for Real-World Implementation

For DRL-based FX hedging strategies to be viable in real-world settings, several improvements and considerations must be addressed:

- **Integration of Macroeconomic Indicators:** Enhancing the feature set with variables such as interest rate projections, inflation trends, and geopolitical risk scores could improve decision-making.
- **Live Trading Validation:** Testing DRL strategies in a real-time trading environment is crucial to assess their practical viability beyond historical backtesting.
- **Cost Optimization:** Addressing transaction costs, particularly spreads, fees, and slippage, is necessary to refine the profitability of DRL-based strategies.
- **Hybrid Approaches:** A promising direction is the combination of DRL with rule-based filters. In the context of FX Hedging, a rule that would force the model to make fewer trades.

Overall, this research contributes to the growing body of work on DRL in finance, demonstrating the limitations in FX hedging. Future studies should continue refining the approach, incorporating more economic indicators, and conducting live market tests to bridge the gap between theoretical performance and real-world applicability.

Overview of Generative AI Tools Used

Chat GTP was used to generate the abstract of the thesis. It was used in Section 3.1 to organize the text, and write small sentences that clearly explain parts of the strategies. It was used very briefly in Section 4.1 and Subsection 4.4.2. It was used in Section 4.2 to clearly explain and organize the reward function with the formulas. All the larger paragraphs were written without ChatGPT. Exceptions are Sections 3.3 and 3.4 which were co-written with ChatGPT, focusing on the organization and structure of the formulas and text. 1/5 of the thesis was written jointly with ChatGPT.

ChatGPT – Accessed from September 1, 2024, to April 20, 2025.

Übersicht verwendeter Hilfsmittel

ChatGPT wurde verwendet, um die Zusammenfassung der Thesis zu erstellen. Es wurde in Abschnitt 3.1 genutzt, um den Text zu strukturieren und kurze Sätze zu formulieren, die die Teile der Strategien klar erklären. Es wurde sehr kurz in Abschnitt 4.1 und Unterabschnitt 4.4.2 verwendet. In Abschnitt 4.2 wurde es eingesetzt, um die Belohnungsfunktion zusammen mit den Formeln klar zu erklären und zu strukturieren. Alle größeren Absätze wurden ohne ChatGPT geschrieben. Ausnahmen bilden die Abschnitte 3.3 und 3.4, die gemeinsam mit ChatGPT verfasst wurden, wobei der Fokus auf der Organisation und Struktur der Formeln und des Textes lag. Ein Fünftel der Thesis wurde gemeinsam mit ChatGPT geschrieben.

ChatGPT – Zugriff vom 1. September 2024 bis zum 20. April 2025.

List of Figures

3.1 Relative Strength Index (RSI) and corresponding trading signals throughout the test period. 8
3.2 Moving Average Crossover (MAC) strategy applied to the USD/CHF exchange rate throughout the test period. 9
3.3 Interaction between an agent and an environment in RL [6]. 11

4.1 USD/CHF Exchange rate throughout the test period, sourced from [18]. . 20
4.2 DDQN Learning Progress. 28
4.3 PPO Learning Progress. 28
4.4 Training rewards. 29
4.5 Cumulative earnings. 30

List of Tables

4.1 Example FX Transaction and Associated Costs 26

4.2 Performance Metrics for Different Strategies. Best values are highlighted in gray. 31

List of Algorithms

1 Double Q-learning for calculating $Q \approx q^*$ and $\pi \approx \pi^*$ [6]. 13
2 Proximal Policy Optimization (PPO) for policy optimization [6] 15

Bibliography

[1] Bank for International Settlements. Triennial Central Bank Survey: Foreign Exchange Turnover in April 2016. *Monetary and Economic Department*, September 2016. Annex tables revised on 11 December 2016. URL: https://www.bis.org/publ/rpfx16.htm.

[2] Corporate Finance Institute. Exponential moving average (EMA) – guide, formula, and example, 2024. Accessed: March 11, 2025. URL: https://corporatefinanceinstitute.com/resources/career-map/sell-side/capital-markets/exponential-moving-average-ema/.

[3] Corporate Finance Institute. MACD oscillator – technical analysis, 2024. Accessed: March 11, 2025. URL: https://corporatefinanceinstitute.com/resources/career-map/sell-side/capital-markets/macd-oscillator-technical-analysis/.

[4] Alex Graves. Generating sequences with recurrent neural networks, 2014. URL: https://arxiv.org/abs/1308.0850, arXiv:1308.0850.

[5] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double Q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2094–2100. AAAI Press, 2016.

[6] Clemens Heitzinger. *Reinforcement Learning: Algorithms & Convergence*. Lecture notes, TU Wien, November 2024. Available at: mailto:Clemens.Heitzinger@TUWien.ac.at. URL: http://Clemens.Heitzinger.name.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi:10.1162/neco.1997.9.8.1735.

[8] Corporate Finance Institute. Relative strength index (RSI), 2023. Accessed: March 8, 2025. URL: https://corporatefinanceinstitute.com/resources/career-map/sell-side/capital-markets/relative-strength-index-rsi/.

[9] Corporate Finance Institute. Simple moving average (SMA), 2023. Accessed: March 8, 2025. URL: https://corporatefinanceinstitute.

com/resources/career-map/sell-side/capital-markets/simple-moving-average-sma/.

[10] Corporate Finance Institute. Beta–definition, formula, and importance, 2024. Accessed: March 11, 2025. URL: https://corporatefinanceinstitute.com/resources/valuation/what-is-beta-guide/.

[11] Corporate Finance Institute. Maximum drawdown–definition, formula, and importance, 2024. Accessed: March 11, 2025. URL: https://corporatefinanceinstitute.com/resources/career-map/sell-side/capital-markets/maximum-drawdown/.

[12] Corporate Finance Institute. Sharpe ratio–definition, formula, and importance, 2024. Accessed: March 11, 2025. URL: https://corporatefinanceinstitute.com/resources/career-map/sell-side/risk-management/sharpe-ratio-definition-formula/.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, and et al. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015. doi:10.1038/nature14236.

[14] Tidor-Vlad Pricope. Deep reinforcement learning in quantitative algorithmic trading: A review, 2021. URL: https://arxiv.org/abs/2106.00123, arXiv:2106.00123.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[16] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, MA, USA, 2nd edition, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html.

[17] Thibaut Théate and Damien Ernst. An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, 173:114632, July 2021. URL: http://dx.doi.org/10.1016/j.eswa.2021.114632, doi:10.1016/j.eswa.2021.114632.

[18] Trading Economics. USD/CHF OHLC Data, 2024. Accessed: March 11, 2025. URL: https://tradingeconomics.com/usdchf:cur.

[19] Ruey S. Tsay. *Analysis of Financial Time Series*. Wiley-Interscience, Hoboken, NJ, 3rd edition, 2010.

[20] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deep reinforcement learning for trading, 2019. URL: https://arxiv.org/abs/1911.10107, arXiv:1911.10107.