NATIONAL RESEARCH UNIVERSITY HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science

Бучков Вячеслав Константинович

Buchkov Viacheslav Konstantinovich

MASTER’S GRADUATE QUAILIFICATION WORK

MASTER GRADUATION PAPER

Хеджирование деривативов в условиях неполного рынка с использованием методов глубинного обучения

Hedging Derivatives Under Incomplete Markets with Deep Learning

Field of study: 01.04.02 «Applied Mathematics and Information Technologies»

Degree programme: Machine Learning and Data-Intensive Systems

Reviewer
Samoylov G.E.
Senior Vice-President, Raiffeisenbank

Supervisor
Piftankin G. N., PhD
Executive Director, Sber

Moscow, 2024

Аннотация.

Данное исследование направлено на создание фреймворка динамического хеджирования производных финансовых инструментов в условиях неполного рынка статистическими методами. Выработанный подход требует только определения функции выплат дериватива как формулы отображения путей изменения цены базовых активов в скалярное значение прибыли, которую получит держатель дериватива — остальные шаги выполняются автоматически в рамках обучения полного цикла. Полученная модель общего вида возвращает вектор весов для реплицирующего портфеля, который далее может быть транслирован в реальные рыночные заявки, замыкая процесс полного цикла для практического использования.

Работа представляет реализацию готового процесса получения хеджирующей стратегии в условиях финансовых рынков с транзакционными издержками, достигая решения в общем случае. Используется подход глубинного обучения с прямой оптимизацией градиентов, в противопоставление уже существующим подходам обучения с подкреплением. Исследование выводит вид функции потерь для оптимизации модели, сравнивает подход с базовыми вариантами как классической финансовой математики через дельта-хеджирование по формуле Блэка-Шоулза и моделированию Хестона, так и с алгоритмами обучения с подкреплением — Soft-Actor Critic и Proximal Policy Optimization.

Проводится обучение и тестирование на реальных рыночных данных bid-ask цен валютных пар USDRUB и EURRUB с частотностью 1 минута на объём 500 тысяч валюты за период 2017-2024, используя в моделировании кривые процентных ставок, действующие на соответствующий период времени. Результаты исследования достигают уровня 1% статистической значимости превышения результатов бейзлайна для ребалансировки раз в 30 минут, а также уровня 0.01% для ребалансировки раз в 5 минут. Дополнительно к этому — в целях демонстрации общности подхода с точки зрения добавления признаков в модель — подход тестируется для предобученных эмбеддингов текстов финансовых Телеграм-каналов, что позволяет улучшить результаты для достижения статистической значимости <<0.01% (с t-Student статистикой на уровне 133.68).

Abstract.

This thesis implements the framework for dynamic statistical hedging under incomplete markets of the derivatives. The framework requires only definition of the payoff function as a formula that translates the paths of base assets into a scalar value of derivative's holder PnL, and everything else is processed in the end-to-end learning manner. The created holistic model returns weights for the hedging portfolio, which can be easily converted into the real market orders, deeming the process to be ready for practical implementation.

The study creates a seamless approach to creating a hedging trading strategies in the market with frictions, attaining the general structure of the resulting framework. The approach of deep learning via direct gradient optimization, contrary to the previous studies in the area of reinforcement learning, is implemented. The paper derives the applicable custom loss for optimization, compares the framework's results with both classical financial mathematics approach of delta-hedging via Black-Scholes and Heston models, and the reinforcement learning baselines of Soft-Actor Critic and Proximal Policy Optimization algorithms.

The model backtesting is done on the real 1 minute market data of USDRUB and EURRUB bid-ask 500k VWAP prices for the period of 2017-2024, using the real interest rate curves, present at the moment. The results show 1% statistical significance for the best-attained baseline outperformance for the 30 minutes frequency of the replicating portfolio rebalancing, and 0.01% significance for 5 minutes rebalancing frequency. Moreover, for the purposes of the simplicity of adding arbitrary feature illustration, the pre-trained embeddings of the financial Telegram channels posts are added, which drives results of the baselines outperformance to the level of <<0.01% significance (133.68 t-Student statistic).

Table Of Contents.

Chapter I. Introduction.

Section 1.1. Description of the field.

Undoubtedly, the world of finance has been rapidly developing throughout the time by not only improvements of the tools and approaches for the existing solutions, but also by engineering brand new tools in order to suit the goals of such a social construct as money. While people have been keen on switching from monotonous work towards artesian applications of their willingness to learn, financial technologies have been making leaps towards substituting need for constant monitoring of money-related tasks via seamless experience both in terms of technological implementation and in relation to the nature of investments themselves.

Therefore, one of the most bright notions in this regard is derivatives as the way to obtain explicit exposure to the variable needed for person’s aim in the investment management area, while the service provider takes upon itself the difficulties related to the implementation of such a made-up variable.

And thus the long way of stepping away from simplistic linear investments towards rebalancing and non-linear exposure have driven the investment banking sphere to become way more client-oriented than it has been before. Now derivatives are used to make investment portfolios that suit retail client’s risk awareness, stabilize cash flows of the corporate clients, protect large financial institutions from the tail risks in their Treasury books and many more applications that a sales desk may come up with.

However, such a popularization of the derivatives — and increasing complexity of their nature — has driven trading divisions of investment banks to employ more and more complex approaches to creating their trading algorithms. Indeed, as a derivative is simply a function that determines payments in relation to some other asset’s price realization — called “based asset price”, a **trading division, by undertaking obligation or right to pay or receive under such a contract, requires to make some operations with some assets in order to produce payoff that will be “as close as possible” to the one that was promised to the client**. This is what called “hedging a derivative”.

Moreover, in a general sense **hedging a derivative means creating some portfolio at each point of time (thus, portfolio might contain different weights of assets in each point of time), such that its payoff will be “the closest to the derivative’s payoff”**. The portfolio that is used for hedging is usually called “replicating portfolio”. The notion of “closeness” in this regard will be defined hereon later, however, the presented approach is not exhausting, and it’s important to note that an investment bank might employ the hedging scheme in any manner that seems appropriate to their goals.

Section 1.2. Purpose of the study.

While mathematically a derivative looks to be quite a well-defined notion, being some, trivial or complex, formula of the base asset's price path (in general case a derivative might depend not only on the final point of the price path, but also on the features of the asset price behavior during derivative's lifetime), the hedging strategy appears to be quite a non-trivial exercise, as the derivatives might appear to be some complex non-linear function that would not follow basic properties, needed for closed-out solution such as one that exists for Black-Scholes-Merton formula (Black & Scholes, 1973; Heston, 1993) of the hedging weights given parameters of the derivative. Therefore, a **potential approach might be to find an approximation of such a solution via deep learning**.

Furthermore, while most of the closed-out solutions tend to assume frictionless market, i.e., market conditions without any transaction costs, imperfections of the actions of other agents etc. and require some adjustments to the initial solutions to be found separately (Burnett, 2020). From this the idea of the research in the area of deep learning arises — **we can attempt to train an end-to-end solution that will be able to account for any frictions via adjustments in dataset and Profit&Loss (PnL) function directly, while <u>being able to solve for any derivative seamlessly,</u> by just taking the conditions as an input**.

Summarizing, this research aims to **<u>introduce a general framework for hedging any derivative as an end-to-end solution, which requires only the historical data of the market variables and the function that defines the derivative in question</u>**. It should be duly noted that the goal of the research presented is not to price a derivative, but to find a replicating portfolio construction algorithm that seeks to eliminate the risk obtained by already having a derivative in the portfolio, which is an actual problem that an investment bank seeks to solve.

Section 1.3. Suggested approach.

Usually, the research in this area is concentrated on implementation of the Reinforcement Learning algorithms in order to solve the hedging puzzle (Bühler, Gonon, Teichmann, Wood, 2018; Du, Jin, Kolm, Ritter, Wang, Zhang, 2020; Bühler, Phillip, Wood, 2022). However, this research shows that the solution via classical deep learning methods provide us with the opportunity to directly optimize our loss function without imperfections between Q-value estimation of reward path and reward itself.

Therefore, in this research a neural network is designed to implement end-to-end learning. A network works on dataset of available features and the spot prices of the base asset. The target PnL is constructed as the payoff of the derivative, depending on the spot path. **The model returns the matrix of hedging weights** $T \times A$, where each column vector of size A shows the weights of the assets in the replicating portfolio, having the $\sum_{a}^{A} w_a = 1$, which is attained by applying softmax function or providing optimization without such a restriction. More details on the architecture is presented in Section 3.3.

Further, the loss function is constructed as the **"closeness" level of the target PnL to the obtained dynamic PnL of the hedging portfolio as the function of hedging weights matrix**. In the basic approach the mean squared error loss is used, but more custom approach is also presented for the sake of generalization of the suggested approach. In terms of application to the real-world trading, usually the target is set to be minimum for the derivative plus the replicating portfolio PnL. However, as this paper's approach assumes symmetric optimization in terms of marker frictions (i.e., taking an asset long and taking an asset short brings / costs the same interest rate), such tasks will produce equivalent weights for the replicating portfolio. But, if needed, the optimization may be easily transformed to the more general one by slight modifications of this framework's approach.

Amongst the features used, **the pre-trained embeddings of texts of the financial news and finance-based Telegram channels are used**, which showed to be add significant improvement to the final metrics. However, such embeddings are used for the illustration purposes of the generality of the approach, so it can be modified to train embeddings too, or to use more advanced NLP architectures.

Overall, suggested framework requires only to define the derivative by the payoff function and determine the hedging goals in terms of the restrictions of the trading desk. As targeting expected value of the replicating portfolio return to be as close to the expected value of the derivative's return is the most usual target of the trading desk, such approach is considered to be target throughout this paper.

Section 1.4. Novelty of the research presented.

First of all, the **research targets implementation of the general framework that requires only payoff function of the derivative to produce more optimal hedging than the classical approaches**, thus, it is considered to be the main value.

Secondly, the research uses **direct optimization of neural network** via classical deep learning techniques, allowing to account for any market frictions directly by construction on PnL calculation function (where one can apply any transaction cost, non-linear rule, restriction for exposure and etc.), while keeping the relative optimization stability, compared to reinforcement learning approaches.

Thirdly, framework introduces the approach that allows for **replicating portfolio optimization by not only the base asset itself, but by any other assets available**, exploiting the correlation patterns in order to decrease costs for hedging. While such definition of the hedging task might bring unusual exposures in the hedging portfolio and allow for severe overfitting, the sufficiently large and representative dataset and re-defined loss will help to overcome such issues within the framework presented.

Finally, the resulting objective-oriented programming realization with the easy-to-deploy neural network show the results that severely overcome the hedging offered by the classical approaches in not only the general market conditions, but also provide lower "tail" — unexpected crisis in the economics, followed by the extreme market movements — exposure in expectation, thus, **the framework can be applied to the practical use of trading desks seamlessly**.

Section 1.5. Structure of the study.

The presentation of the study in this paper is structured according to the following logic. Chapter II overviews the existing researches in the are of both classical closed-out approaches to the hedging strategy formulation plus the statistical attempts to determine the optimal hedge with the focus on the reinforcement learning and deep learning techniques, as this is of the main interest of the presented study. Chapter III introduces the framework, details methodology on the end-to-end learning, presents the model architecture and defines the loss functions.

Further, the baselines for the study findings are defined. Chapter IV shows the baseline in terms of the classical approaches to hedging via financial logic and well-known models from financial mathematics. Chapter V works with the reinforcement learning baseline for reference to the previous authors' works.

Moreover, Chapter VI defines the architecture of the model that this paper suggests, along with the important details on the optimization procedures.

Later on, Chapter VII corresponds to the discussion of the results of the implementation of the framework and backtesting it on the real market data. Moreover, it dives deep into the text features extraction and the embedding processing in the neural network. Chapter VIII provides details on ablation study in regards to the model itself and the hedging experiments designs as such. Finally, Chapter IX is determined to conclude the bottom-line of the presented study, discuss its findings and determine the potential vectors of the further studies in this area.

Chapter II. Literature review.

Section 2.1. Derivatives hedging models.

The study focuses on algorithmic optimization techniques of the hedging portfolio construction. However, the derivatives as an instrument have been studied ever since they gained popularity. And while the algorithmic approximation seems to prevail as a novelty, the closed-out solutions have been developing rapidly.

Overall, any closed-out solution is inevitably an approach that assumes some distribution of returns and market behavior patterns and bases its derivations on those assumptions. Undoubtedly, the base for all of the studies of the non-linear derivatives hedging — first non-trivial hedging case, compared to linear derivatives that produce ideal hedge without any stochasticity, as linear exposure on the base asset is always available — have started with the famous derivation of Black-Scholes model (Black & Scholes, 1973). While the paper was focused on finding the fair price of such derivative, the authors found the hedging solution parallel to that. As price of the option is derived via decay of the option's value throughout its lifetime, the replicating portfolio implementation bases its weights derivation on the simple idea that at each point the option value can be approximated by the tangent line — thus, by the first derivative of the option's price by spot price. This first derivative was named "delta" and, consequently, the hedging technique became "delta-hedging".

This paper gave boost to the studies of the options pricing, giving the other authors opportunities to add some adjustments to the Black-Scholes findings. First of all, many authors tried to ease assumptions of the Black-Scholes model. For instance, (Merton, 1973) produces adjustment to pricing, and hedging, in the case of underlying's fixed cash flow payments, such as stock dividends.

Furthermore, the authors tried to address one of the arguably most important pitfalls of the Black-Scholes model — constant volatility. From the basic ideas of local volatility (Dupire, 1994) and stochastic volatility (Heston, 1993; Derman & Kani, 1994a) to complex distribution skewness adjustments (Corrado & Su, 1996) authors have been researching the best way to correct Black-Scholes' model, while keeping the idea at the same spot. Thus, the hedging rule still remained the same — just this first derivative's calculation have changed.

A more advanced attempt to account for volatility stochasticity, and rather well-implemented, is the Bergomi model (Bergomi, 2017), which uses complex non-linearity of the volatility behavior, keeping the solution (or rather the approximation) closed-out.

Moreover, while the base asset's distribution usually turns out to be well-fit by the authors in the most of the returns points, one of the previous papers' largest weak spots is the fact that they do not account for sudden changes in prices, which is usually called "tail event", which is usually considered to be some sudden change that does not fall well within the distribution, used for most of the cases of stock returns. Thus, other authors have addressed the issue of such sort — for instance, (Ahn, 1992; Trautmann & Beinert, 1970; Bates, 1991; Galluccio & Le Cam, 2005).

The next big step in the derivatives' pricing and hedging improvements was done via accounting for market imperfections in terms of liquidity and transaction costs. As classical Black-Scholes model assumes frictionless markets, many papers like (Seelama & Thongtha, 2021) and (Volk-Makarewicz et al., 2022) have adjusted option pricing and delta-hedging by including the transaction frictions in their models.

Another example of more general attempt to account for transaction costs was done by (Burnett, 2020), which introduced overall adjustment into XVA family that accounts for transaction costs. However, all of those approaches to spread adjustments work only with limited amount of costs and assume some vanilla payment schemes like percentage of volume traded, thus, lacking generalization of the transaction adjustments, which is usually quite important for the trading desk hedging procedures that should ideally account for investment bank's infrastructure as a competitive advantage.

Section 2.2. Statistical learning for the derivatives hedging.

The advance on continuous-time setting of Black-Scholes model was done in (Cox et al., 1979), as the actual trading for hedging always happens in the discrete time, which was accounted for by authors that used binomial tree pricing in order to discretize the price movements.

Thus, the next stepping stone of the binomial trees is the idea that instead of following the closed-out solution derivation path, one could attempt to simulate the base asset behavior and calculate the required weights for replication directly. The idea of price distribution generation for such purposes were introduced by (Scott, 1987) and developed further in (Dempster & Thorlacius, 1998). Such approach improved greatly on the generalization of the modeling, allowing to explicitly account for market frictions during path generation and hedging weights calculation.

The advances in the hedging attempts via algorithmic approximation began in 2017 with the paper "Statistical Hedging" (Bühler, 2013) that extended Markowitz optimization of the portfolio to the delta-hedging techniques. However, the main advances were done by "Deep Hedging" (Bühler, 2018) that expanded the idea of model-free approach to derivatives modeling, however, using synthetic data generation for the path creation of the derivatives, thus, recreating the actual distribution of the base asset returns and then using Monte Carlo "periodic policy search" for finding the optimal hedging strategy, thus, switching to the Reinforcement Learning framework.

After that the advances were done in "Deep Bellman Hedging" (Bühler, Phillip, Wood, 2022) that goes beyond the classical implementation with the well-defined derivatives and expands the framework to any portfolio of derivatives.

Another attempts to use statistical hedging can be noted in (Son, Kim, 2019) and (Du, Jin, Kolm, Ritter, Wang, Zhang, 2020), however, there authors just reached out to the more advanced techniques for the optimization of the hedging procedures.

These advanced studies highlight the importance of the hedging optimization research and show great results in terms of application to the particular tasks of the trading desk. However, the Reinforcement Learning approach is noted to be more fragile in terms of the optimization, while the frameworks of these authors could be improved in terms of generalization and trading task formulation for incomplete markets.

Chapter III. Methodology of the study.

As was mentioned before, this thesis deep dives into the hedging techniques for derivatives — instruments that have payoff, dependent on the price path of some other instrument. The replicating portfolio is defined as the strategy that rebalances funds — allocates money between all assets that are included in the universe — during lifetime of the derivative with some fixed frequency.

The trading desk, as usually done, is assumed to support zero cash position at each point of hedging path, therefore, having need to borrow extra money, in case of holding net long position, and lend out the excess funds, in case of holding net short position.

The replicating portfolio is assumed to have no restrictions on the amount of capital allocated in each asset and in all assets in total. In other words, it is assumed that the trading desk can borrow and lend any amount of money at some rates, which, in general, might be not fixed and even volume-dependent. Moreover, it is assumed that the strategy can take any leverage needed and can short asset at some given rate, which may differ from the risk-free rate and any other rate present in the modelling.

However, while such assumptions seem to be quite loose, the actual results of the optimization show that the loss function effectively restricts the weights to be within the reasonable bounds. An important note is that while the weights restriction is a common technique for the portfolio optimization problems, here the restriction is not needed, as the trading desk is surely able to borrow quite large sums from the Treasury of the investment bank, so in practice usually the restrictions of such a manner are not reasonable. Moreover, as the model PnL is constructed with respect to the amount of interest rate paid or received (details given in Section 3.5), the restriction of the leverage is implicitly taken into account inside such PnL calculation, because in imposes infinite borrowing costs on infinite leverage.

The trading desk is assumed to be a rational agent with risk-tolerance being neutral in regards to the amount of risk undertaken. And all the base assets in the universe are assumed to have no arbitrage opportunities of either Type I or Type II arbitrage.

Section 3.1. Assets universe.

Assets universe is such array of the instruments that the trading desk can trade within the restriction of the optimization task of our end-to-end solution. It is essentially the assets, for which the trading desk 1) have sufficient price path history for optimization; 2) can trade on inference within the imposed assumptions. Thus, we can consider instruments universe as the **hyperparameter for our model**.

While the framework aims to implement general approach to derivatives hedging, some slight details in the implementation might differ across different base asset classes, however, the changes needed to make is miscellaneous. Therefore, for the sake of demonstration this paper uses FX (Foreign Exchange) as the base asset class.

All assets are assumed to be nominated and traded in the same currency, however, the framework might be adjusted to allow cross-currency deals by simply adjusting the risk-free rate dataset creation (Section 3.9) and PnL calculation (Section 3.5).

Section 3.2. Hedging framework.

The proposed framework itself is composed as an end-to-end model that takes market data and payoff function as an input and produces weights for replicating portfolio during inference. The model is called $Hedger$ here and further on. The model mechanics are illustrated by the Figure 1.

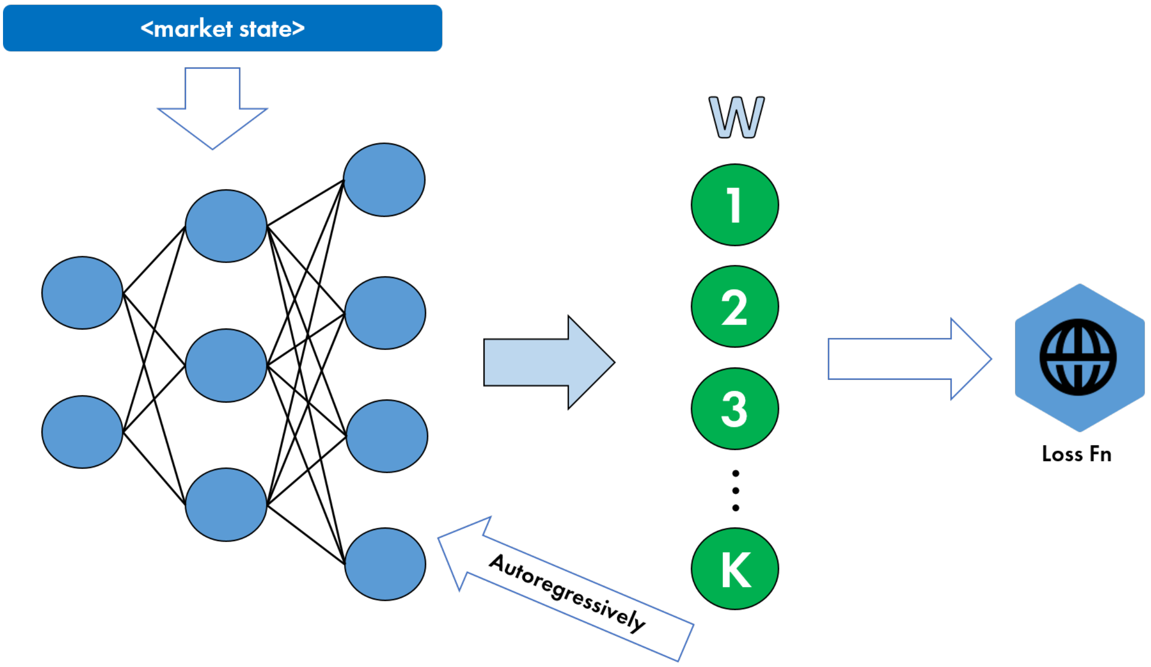

Figure 1: Hedging framework. Source: Sketch of the author.

In the framework each instrument is considered to be composed of two main features only — some price at the moment T_0 and the payoff function that determines cash flows stream for some vector of time points $t \in T$. Therefore, the Python implementation (see Appendix I) determines abstract class $Instrument$ that determines such a behavior, and the instruments that are of interest for our $Hedger$ are inherited from this base class and require implementation of pricing and payoff function. It is important to note that this universal framework of $Hedger$ might be easily modified to compose $Pricer$ implementation, using the same mechanics that are outlined in this paper and calculating instrument's price as the expected value of the hedging proceeds. However, as such modification is out of the scope of this paper, it will not be touched hereon.

The hedging itself, as was outlined in the previous chapter, is considered to be the iterative process of generating vector $w_t \in R^A$ for A assets in the universe, determined by the procedure of

Section 3.2, and such vector is generated iteratively for each moment of time $t \in (T_0; T_n)$, where T_0 is the moment of inception of the derivative and T_n is the time of maturity of the derivative.

It is extremely important to note that the time periods interval is not strict at the bounds. This is because the replicating portfolio is formed from moment zero. Thus, as we know the parameters of the derivative only at time T_0 — and even more so, we generally do not know the fact that we will have a derivative in trading book before time T_0 either —, the hedging procedure cannot start earlier than the time $T_0 + \epsilon$, where ϵ is the hedging frequency (e.g., if we are looking at 5 minute prices, the hedging frequency is 5 minutes). Consequently, **the vector of weights at the point of derivative inception is the vector of zeros at the time T_0**.

Furthermore, at the point of maturity of the derivative we have no more need to keep hedging portfolio in our trading books, as if we follow the classical assumption that the realization of the effect of new information on prices is simultaneous for all the assets in our universe (and by "simultaneous" more strictly the epsilon-simultanous effect is meant — the effect is so fast, such that if it happens at any $t \in (T; T + \epsilon]$, then all of the prices at $T + \epsilon$ account for this effect). Thus, if such assumption did not hold, then the arbitrage opportunities would have existed. Consequently, such an assumption follows the no-arbitrage assumption and is not new. And then, if we do not want to hold any replication portfolio after the derivative have matured, the **final point of time should also correspond to the zero vector of hedging weights**.

Summarizing previous two points, the matrix of hedging weights that we will use for the target metric of the hedging portfolio should consist of zero column vector at the left column and zero column vector at the right column. Such construction does not affect the optimization itself (we just generate $T - 2$ vectors of weights autoregressively), but affects the portfolio metric calculation and the design of the experiment.

Thus, the portfolio dynamics comprises from the **autoregressive generation of weights** — we know all the realized market data before current time t_{now}, and we know the matrix of weights, composed from the vectors, obtained at each time point $t \in [T_0 + \epsilon; t_{now} - \epsilon]$. The idea is presented at Figure 1 above.

Section 3.3. End-to-end learning.

As was emphasized in the previous sections, the aim of this study is to construct the hedging task as an end-to-end learning process. While the baselines of classical approaches are constructed within the given framework, they require determined hedging rule. The reinforcement learning baseline is modeled within the usual approach of this sphere.

The $Hedger$ service takes as an input only the class module of the derivative that was outlined in Section 3.2 with all necessary parameters, and then creates the object from the $Market\ Data$ module and all the hyperparameters embedded in the hedging mechanism.

As *Hedger* aims to be an end-to-end solution to the task, the training is also embedded in the mechanism, if the training for such a derivative was not done before. For the sake of this paper's aims the service does not account for potential market data changes (i.e., does not check availability of the new datapoints in order to trigger training of the model parameters from scratch). The production solution might include such re-training, but should be optimized in order to fine-tune the model and not trigger optimization on the full dataset, as it might be computationally costly. As such re-training is out of the scope of this study, it will not be discussed further.

The technical implementation approaches this via pytorch's *Dataset* that takes market data pandas *DataFrame* and available features. For the sake of end-to-end solution's completeness, the *Dataset* **does not take target PnL as an input, but rather calls to the derivative class's payoff method in order to transform base asset's price path into derivative's payoff**, deeming the solution to actually require only the definition of the derivative's payoff function, as needed by the framework's universality.

Section 3.4. Hedging task formulation.

After we have obtained the weights of the replicating portfolio, we should **calculate the profit or loss of the position** in order to compare it to the profit or loss on the derivative and try to **minimize the difference between the two**.

First of all, it should be noted that the PnL is a random variable that changes throughout the lifetime of the derivative. However, for the sake of this business task, we are interested only in two features of the PnL path — we <u>**want the PnL to be as close as possible to the derivative's payoff at maturity and to be as smooth as possible during the derivative's lifetime**</u>.

Such a formulation is dictated by the usual hedging task formulation. When investment bank trades a derivative, it receives some fixed margin from the client from the fair price of the instrument. Therefore, when the derivative is traded, the trading desk receives the instrument at fair price, books some profit, and, given that the fair price is calculated correctly, in expectation large number of such deals will produce payoffs that on average will be equal to the fair price (Hull, 2012):

$$\mathscr{P} = e^{-rt}\mathbb{E}^{Q}(\Phi(S)) \tag{1}$$

, where

$\mathscr{P}$ is the price of the derivative,

$\Phi(\cdot)$ is the payoff of the derivative,

S is the price path of the base asset,

e^{-rt} is the discounting factor that transforms future value into the present value

$E^{Q}(\cdot)$ is the expectation in risk-neutral measure.

However, this idea only defines the profit that a risk-neutral investor makes in probabilistic limit sense, deeming sample mean of the derivative portfolio payoffs to tend to this amount of cash flows, received from clients:

$$\lim_{i \to +\infty} \sum_i e^{-rt}\Phi(S_i) \stackrel{p}{=} \mathscr{P} \tag{2}$$

.

But actually, contrary to the prevailing sentiment, trading desk solves the task of minimizing the volatility of its PnL by collecting fixed margin only, and not entering into the games of holding some positions and sitting throughout the market volatility. And here the hedging comes into play. The hedging is thus defined as some operations at the financial markets that minimize the volatility of the derivatives payoffs portfolio, adding some **hedging PnL** — another random variable that is engineered to **cancel out the randomness of the derivatives payoffs portfolio**. Therefore, we want our hedging task to have the following properties:

$$\begin{cases} Var^{Q}[f_\theta(S) - \Phi(S)] = \mathbb{E}^{Q}[f_\theta(S) - \Phi(S)]^2 \to \min\limits_{\theta} \\ \mathbb{E}^{Q}[f_\theta(S) - \Phi(S)] = 0 \end{cases} \tag{3}$$

, where

f_θ is the PnL function from the price path of the derivative,

$\Phi(\cdot)$ is the payoff of the derivative.

As discounting is applied to both PnL and payoff of the derivative and is deterministic, it can be easily dispensed of.

The trivial cases of such a hedging can be easily stated. For instance, the so-called "back-to-back hedging" is when the trading desk buys a derivative from one client and sells exactly the same derivative to the other client or other investment bank (and visa versa). Such hedging guarantees zero volatility (as no actual derivative is left in trading books, while credit risk and other disruptions are out of the scope of this discussion). However, while selling the same derivative to the other client would provide investment bank with additional margin, such an operation (called "internalization") is very rare due to uniqueness of each derivative and requires extremely large network of clients. On the other hand, selling it to another investment bank would require paying them some commission / margin, thus, setting net margin from the initial client at the lower level.

Therefore, the trading desk usually solves such a puzzle via hedging a derivative on their on via the open market operations, using the knowledge of financial mathematics in order to solve the optimization task, defined above.

Furthermore, **we can combine such optimization task to less strict one**. The trading desk **suffers from volatility of the trading books PnL** (which is exactly $f_\theta(S) - \Phi(S)$, because the derivative and its hedge is considered to be tied to each other in all of the banking practices) and **from the bias in the hedging function** equally. In other words, traders are always ready to pay some margin in order to hedge the derivative more properly in practice. However, technically such result follows from our assumption of risk-neutrality of the trading desk.

Thus, if we can combine the bias and variance of our estimator, i.e. hedging strategy, the resulting loss function is non other than the **classical Mean Squared Error of the hedging PnL function versus the derivative's payoff**:

$$MSE(f_\theta(S), \Phi(S)) = \mathbb{E}[f_\theta(S) - \Phi(S)]^2 + \mathbb{V}(f_\theta(S)) \tag{4}$$

. By such a function we will be able to minimize the variance of portfolio's PnL, while providing the lowest standard deviation of the obtained portfolio:

$$MSE(f_\theta(S), \Phi(S)) = \mathbb{E}[\underbrace{f_\theta(S) - \Phi(S)}_{\textbf{Bias}}]^2 + \mathbb{V}(\underbrace{f_\theta(S)}_{\textbf{Variance}})$$

Figure 2: Financial logic of the loss function. Source: Derivation of the author.

.

Finally, our optimization task is determined as minimization of the MSE between the the PnL obtained from the hedging portfolio (see Section 3.5) and the derivative's payoff, as defined by the user of the $\boldsymbol{Hedger}$ (see Section 3.3):

$$MSE(f_\theta(S), \Phi(S)) \rightarrow \min_\theta \tag{5}$$

.

Consequently, the optimization task is defined, then, let's proceed with discussion on the PnL calculation.

Section 3.5. PnL construction.

An important note in this regard is the fact that while the path of the replicating portfolio's PnLs will consist of as many datapoints, as we will have in our asset universe's market data between the inception and maturity of the derivative, for the sake of the optimization, outlined in Section 3.4, we **require only the last point of the replicating portfolio PnL path**.

Such a conclusion is done on the base of the simple idea. Imagine, if we could definitely know, what price realization would be at the final point of time, which is the only one needed for the derivative's payoff determination. Therefore, we would be able to infer, how much of the base asset we need to hold in the portfolio in order to hedge the derivative with zero variance. Indeed, for instance, let's consider an European option, which has payoff of:

$$\max(S_T - K,\ 0) \tag{6}$$

, where

S_T is the final point of the base asset price path,

K is the strike price of the option.

Thus, if we knew for sure that the price realization will bring base asset price to be above strike, we would hold exactly 1 stock long, and if the price realization will result in the asset price being below the strike, we would hold exactly 0 stocks. The example illustrates the logic of the hedging procedures, so the approach is **generalized to any derivative in question — it is some function of the base asset price behavior**.

Therefore, **why do we need all the weights inside the time interval between the inception and the maturity**? Simply because we do not know the probability of the non-linear (in regards to the base asset) derivative payoff realizations, we **need to adjust the hedging portfolio weights in accordance with the updates of the predicted probability** of realization, as new information comes from market.

Consequently, we need to construct our metric to account for such a formulation, i.e., to be able to **combine the information across all points**. And the final PnL suits such a goal perfectly. Indeed, if we buy an asset before the positive-dependent part of the derivative's payoff becomes more probable, we gain positive value both in our PnL and in the derivative's PnL, allowing the $Hedger$ to adjust weights dynamically in order to suit such probability.

Therefore, the target for our optimization (*y* in standard ML notation) is defined as the payoff of the derivative, which we approximate by adjusting the weights of the neural network and **obtaining the replicating portfolio PnL, which should able to distribute gradients across all the weights in the matrix, and via these weights — to all the parameters of the neural network**. The architecture design will be discussed further in Chapter VI.

Moving on, the calculation of this PnL follows the standard financial logic, as we seek to obtain the amount of money, earned by our hedging strategy — i.e., <u>if an investor puts 0 units of currency at the inception (as defined by zero initial cash flow in the beginning of this Chapter) and undertakes all the risks, which amount of money — positive or negative — will he obtain at the end of the startegy lifetime</u>.

So, **this amount can be calculated via net cash flows methods**. We should determine the amount of the cash paid or received at each point of time for each asset. As an example, if the weights as zero and became negative, then the investor started to hold the base asset short, thus, he received the proceeds from the sale of this asset, when the short position was opened. And if the investor buys the asset back later at some new price, switching back negative weight to zero weight, he will be deemed to pay the proceeds. His PnL, as usual, will be exactly equal to the net cash flows of such operations — if the price of the buying back is lower, then the amount of proceeds paid is below the amount of proceeds received, thus, PnL is some positive amount of currency units in this case.

The outlined logic can be illustrated by the following formula:

$$Deals\ PnL = \sum_{t}^{T} CF_t(\Delta w_t) \tag{7}$$

.

Consequently, the PnL of the position is simply the sum of the cash flows of the sales and purchases proceeds (Luenberger, 2014).

Let's remember that the obtained weights matrix has zeros at the left vector column and at the right vector column, looking as following:

$$W = \begin{pmatrix} 0 & w_{t_0}^{a_1} & w_{t_1}^{a_1} & \dots & 0 \\ 0 & w_{t_0}^{a_2} & w_{t_1}^{a_2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & w_{t_0}^{a_A} & w_{t_1}^{a_A} & \dots & 0 \end{pmatrix} \tag{8}$$

, so, the initial cash flow is always the weight of the asset for the first rebalancing time slice available. And also, as we do not hold the position after the derivative matures, **the final weights are also zeroes, which means that we close the remaining positions at the market prices** at the price points, which are used for the base asset price determination.

However, the cash flows from the proceeds on sales and purchases of the assets in the universe are not the only ones that are included in the PnL. **As we assume zero cash at each point of time, we are required to account for the costs of borrowing funds and proceeds from placing extra funds in the deposit**. As usually trading desk places and borrows from its Treasury, we can assume that there are no restrictions on the amount borrowed and the deal is risk-free in terms of credit risk, however, the dataset in this framework can be constructed to account for such events by simply including those in the rates market data (see Section 3.9).

Thus, at each point of time a trading desk **need to borrow and lend funds for the net weight across the assets** — e.g., if we have two assets in the universe, and one held short and the other is held long for equal amount of initial proceeds, there is no need for lending or borrowing, as all assets are assumed to be nominated and traded in the same currency (see Section 3.1).

In general case, the investor **pays the difference between two rates**. For instance, as we consider FX derivatives in this study as an illustration, by holding an asset — FX pair that consists of base currency traded versus second currency (e.g., for USDRUB USD is the base currency and RUB is the second currency) — long, the investor generally borrows in the second currency, buys base currency and lends it out, obtaining the cash flow, nominated in second currency. A detailed example for USDRUB would suppose that an investor borrows S amount of RUB (where S is the spot price of the USDRUB) and converts it to get 1 USD, obtaining an exposure for an amount of 1. Therefore, the investor pays some interest rate r_{RUB} for borrowed S RUB and collects r_{USD} for the 1 USD. However, as we need to convert it into one currency, which is the base for our PnL calculation (usually defined simply by the country, where investment bank operates), the net interest rate follows to be plain difference of interest rates:

$$RUB\ interest = -\, r_{RUB} \cdot S + r_{USD} \cdot 1 \cdot S = -\, S \cdot (r_{RUB} - r_{USD}) \tag{9}$$

.

Thus, we can compute net interest proceeds as sum of negative sum of weights at each point of time. The sum of weights is multiplied by -1, because if we are long — weight is positive —, then we borrow in second currency and lend in the base currency, thus, we are net receivers of the rates difference $r_{RUB} - r_{USD}$, which is conventionally quoted as the second currency's risk-free rate minus the base currency's one:

$$Interest\ PnL = -\sum_{t}^{T}\sum_{a}^{A} w_t^a \tag{10}$$

. It is important to note that such calculation framework allows to not only account for **stochastic interest rates**, which is absent in Black-Scholes-Merton formulation (see Chapter IV), but also gives us an opportunity to include transaction costs in interest rates too, e.g., by setting lending and borrowing rates to be different even for the same currency.

As was outlined in the introduction to this paper, the study focuses on the universal framework for the incomplete markets — thus, where transaction costs exist. Therefore, the transaction costs can be easily accounted for in our formulation — we just need to include those in the PnL calculation, in order to allow the gradient optimization to affect them too. In general, **transaction costs can affect the optimality of hedge by adjusting the frequency of hedging — sometimes the potential shifts of derivative payoff probability is insignificant, compared to the amount of costs, needed to hedge against such movements by buying or selling the base asset**.

In this study **the transaction costs are defined in terms of bid-ask spread**. It means that the trading desk does cannot buy and sell asset at the same price at the same point of time, but those prices differ by the bid-ask spread:

$$Bid = Mid - \pi < Mid < Mid + \pi = Ask \tag{11}$$

, where

2π is the bid-ask spread, which is the actual market variable (not hyperparameter), which model can account for in the optimization process.

Consequently, **the PnL is easily adjustable for such parameters — formula (7) will simply use the corresponding price from the dataset in order to calculate proceeds from sales and purchases**. If needed, the brokerage costs and other transaction frictions can be added, but for the sake of generality only bid-ask spread, being not hyperparameter, but actual variable, is used.

While the formulas above use FX assets for the purposes of illustration, it should be noted that **stocks, bonds, commodities etc. also fit in this framework and do not require any changes to the ideas above**. As the proceeds from sales and purchases work perfectly the same, the interest rates difference should be only adjusted at the time of dataset creation. For instance, in case of stocks, second currency rate stays the same (currency, in which stock is traded) and base currency interest rate should be substituted by the REPO rate (REPO is the type of deal, fully called Repurchasing Agreement — one investor borrows securities from another at some interest rate for the fixed term):

$$RUB\ interest = -r_{RUB} \cdot S + r_{REPO} \cdot 1 \cdot S = -S \cdot (r_{RUB} - r_{REPO}) \tag{11}$$

.

Finally, the PnL, which is the metric for MSE calculation is given as follows:

$$PnL = Deals\ PnL + Interest\ PnL = \sum_{t}^{T} CF_t(\Delta w_t) - \sum_{t}^{T} \sum_{a}^{A} w_t^a \tag{12}$$

.

Section 3.6. Modeling many assets.

As was mentioned above, all the calculations are easily generalized for the asset universe of numerous assets. Moreover, while the findings of the study below do not touch cases, when the base asset is not in the universe, the framework allows to easily create the model that would hedge the derivative using many assets **without base asset in that set**.

The different assets in this methodology are considered to be equally attainable, but **they might have different transaction cost levels, which should be depicted by the bid-ask spread**. Any other commission-related frictions that create imbalance between assets are supposed to be included in the bid-ask spread setting:

$$\pi_{new} = \pi + c \tag{13}$$

, where

π is the bid-ask spread,

c is the other costs like brokerage commission,

using the notation and idea from equation (11) in Section 3.4.

The assets dataset is composed using **merge by datetime timestamp**, deeming assets to be available for trading only at the relevant price realization time. Moreover, for the purposes of simplicity of the study, the timestamps with absent values for any of the assets is dropped for the whole dataset, driving the rebalancing frequency to lengthen for all other assets in order to provide simultaneous hedging. Such method is used only for the purposes of easier calculations and does not bear any special financial logic.

Furthermore, while an assumption of $\sum_{a}^{A} w_a = 1$ for A assets in the universe is usually imposed, the framework allows the optimization to be constructed with such restriction by simply imposing Softmax function of:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_k^K z_k} \tag{14}$$

(Bridle, 1989) upon the produced weights of the model. Thus, for the sake of generalization, this point is also taken into account.

Section 3.7. Derivative classes in the experiment.

The study focuses on implementation of two derivatives in particular, but is easily generalizable to any derivatives, as was shown in Section 3.3. It is important to remember that the price of the derivative is irrelevant for our business task — **when the deal is done, then the price of the derivative is already received from or paid to the client**, thus, represents merely a fixed amount of money that bears no stochasticity. Thus, we do not need any implementation for pricing in this regard.

The first one is simple **Forward, done at strike, equal to the fair forward price, that has zero price at inception**:

$$F = S_0 \cdot e^{(r_s(T) - r_b(T))T} \tag{15}$$

, where

S_0 is the spot price at the inception of the derivative,

T is time till maturity of the derivative,

$r_s(T)$ is second asset's interest rate for the term T,

$r_b(T)$ is the base asset's interest rate for the term T

. And the payoff function for this derivative is defined simply as:

$$S_T - F \tag{16}$$

, where

S_T is the spot price at the maturity of the derivative

. As the spot start is simply the price of the underlying at T_0, the only parameters that are external for determining the derivative is **time till maturity and strike**.

Therefore, Forward is a **linear derivative**, which can be inferred from (16) — its price depends linearly from the spot price. Thus, Forward is chosen as a trivial case, where all hedging weights from the model should be equal to 1 in the baseline case — providing us with the opportunity to analyze potential overfitting, or, alternatively, looking at the interesting patterns in the model's optimization results.

Furthermore, for the completeness of the study we need the core derivative that will be in the focus of the study, and we **need it to possess non-linear dependance in regards to the underlying's price path**. Such derivative, which is well-studied in terms of closed-out solutions, is **European Call**. Just like Forward, it has only two parameters — **time till maturity and strike** — and has payoff of:

$$max(S_T, K) \tag{17}$$

, where

S_0 is the spot price at the maturity of the derivative,

T is time till maturity of the derivative,

K is the strike price, which is given as the input parameter

.

And as the framework used is general enough, **we do not need to know any more features of these instruments in order to produce sustainable solutions that overcome baselines of classical finance that are based on knowledge of the markets rules of behavior** (outlined in Chapter V).

In the experiments below the starting point of the derivative — S_0 — is taken always as mid price, without any transaction costs, as is almost always done in investment banking practice.

Section 3.8. Construction of the sample.

The only question left for the full experiment definition is how to construct the samples, as it might be tricky in terms of financial data.

As the model should be able to **optimize parameters, producing hedging weights, for a general case** and to not overfit to the particular market regime and its patterns, we can create samples in the following way. **The training set's point is composed as price points and features inside the N_{DAYS} interval, which is a hyperparameter of number of days till maturity of our derivative in interest. Then we shift our pointer to the next price point available (i.e., shift time by ϵ, which is the hedging frequency, defined in Section 3.2) and assume that another derivative with the start at new time point $T_0 + \epsilon$ was created for N_{DAYS}**. As the hedging frequency is relatively small (1 minute, 5 minutes, 30 minutes, 1 hour) and N_{DAYS} is sufficiently large (5 days), such samples turn out to contain different volatilities, price patterns and contain different sets of price points.

However, such methodology would provide highly correlated samples during optimization, which would make gradients biased. Therefore, as we seek general solution, **we can easily shuffle batches in order to avoid correlation of samples**. Thus, if our dataset is large enough, the probability of having two sequential datapoints is very small due to uniform distribution of shuffle positions. From this follows that our **samples will be uncorrelated and the gradients remain unbiased**.

While the model will process the training data non-sequentially, as no information about which prices were first in the timeline, and which happened afterwards, no data leak will be present due to this methodology.

As the number of datapoints inside one sample is unequal between different samples inside one batch and across all batches, custom $collate\_fn$ function is implemented to work with unequal samples.

Finally, the test sample will not be drawn from the same bucket as the training set, but rather **the whole dataset will be split by the timestamp, making training data precede the testing data** in order to implement the correct experiment, as if the trading desk would have trained the model and implemented in production trading.

Figure 3: Train/Test split. Source: Methodology of the author.

Section 3.9. Yield curve.

The final stepping stone in the methodology is the interest rates curves that should be used for PnL calculation and as features for both baseline and target solution.

As was stated beforehand, the interest rate curve will not be different for borrowing and lending, as such data is not easily obtainable, while imposing some costs on the interest rate frictions as hyperparameter will not provide sustainable ground for experiments.

For the purposes of correctness of the interpolation, the most appropriate way to attempt experiments is to take an interest rate curve for the corresponding currencies from the open datasources. Usually, the data is stored as parameters of Nelson-Siegel-Svensson model (Nelson & Siegel, 1987; Svensson, 1994):

$$r(T) = \beta_0 + \beta_1\left(\frac{1-e^{-\frac{T}{\lambda_1}}}{\frac{T}{\lambda_1}}\right) + \beta_2\left(\frac{1-e^{-\frac{T}{\lambda_1}}}{\frac{T}{\lambda_1}} - e^{-\frac{T}{\lambda_1}}\right) + \beta_3\left(\frac{1-e^{-\frac{T}{\lambda_2}}}{\frac{T}{\lambda_2}} - e^{-\frac{T}{\lambda_2}}\right) \quad (18)$$

, where

T is time till maturity of the deposit / loan from this interest rate curve,

β_0, β_1.β_2.β_3,λ_1,λ_2 are parameters of the model that are fitted into the observed interest curve . This model allows to **interpolate the points of the actual interest rate curve** and obtain any needed point further on.

For the instruments in question the parameters of the NSS curve are stored at the timestamps of historical data, so these parameters are used. For **each timestamp its own NSS curve is created**, which corresponds not only to the correct interest rate point for the modeling, but also possess the curvature that might be a good predictor for the spot prices movements, thus, can also be used as feature, which is out of the scope of the examples below, but can be a potential improvement to this study.

The curves' parameters are merged with the spot prices dataset by datetime, following the procedure of Section 3.9. The absent points were meant to be also dropped, but the examples below do not have such cases present amongst them.

While the full and comprehensive experiment design would require taking at each point the correct interest rate for the period of time ϵ (hedging frequency), such a design requires extensive computational resources, but bears little mathematical background, as the slight changes in the interest rates for the interpolated curve might just add additional noise, as interpolation is imperfect, thus, already bears some noise. Moreover, a **trading desk would usually pay or receive interest rate at the level for the end of the trading day, because intraday placements are usually absent, thus, only overnight placements and borrowings make sense**. Therefore, the experiment is designed to just take overnight rates as $r(1/365)$ and scale them to the correct Δt, which is $\Delta t = \epsilon$, our hedging frequency. Therefore, it assumes that all rebalancings are netted intraday, i.e., if we bought 1 stock and then sold it after 5 minutes, we do not need to pay any costs for the money borrowing (and do not receive any proceeds from lending money out), which follows exactly the logic, implemented in practice.

Another detail of the interest rate design is the fact that while a **fully rigorous experiment would require** to account for accrual of interest rate by the **simple compounding at frequency** ϵ, having:

$$r_{paid} = (1 + \frac{r_{o/n}}{1/\epsilon})^{1/\epsilon} - 1 \tag{19}$$

.

However, in terms of financial logic such idea would not be fully correct, because, as stated above, the actual trading desk has accruals at daily frequency, but not at the hedging frequency ϵ. This is so, because the **actual compounding occurs at the frequency, at which "the actual bank account" changes, but not at the hedging frequency**.

But as both of these approaches would require significant time for computation, while bear miscellaneous impact on the final result, because interest rate PnL is way lower than the deals PnL in virtually any environment, such restrictions are eased and the continuous compounding is used in the experiments:

$$r_{paid} = e^{r_{o/n}} - 1 \tag{20}$$

.

Chapter IV. Classical approach baseline.

Section 4.1. Hedging rule formulation.

As was outlined in the analysis of existing literature in Charter I, the closed-out solutions follow simple idea in terms of calculation of the hedging weights. Suppose that we are given some function of market data parameters that defines the derivative price:

$$\Pi(\cdot) = e^{r_{diff}T}\mathbb{E}^{Q}[\Phi(S)] \tag{21}$$

, where

S is the spot price path,

T is time till maturity of the derivative,

r_{diff} is the rates difference, as outlined in Section 3.9

.

Therefore, all the approaches that are based on closed-out solutions or Monte-Carlo simulation define hedging via **price function approximation by the tangent line**, attained by holding the base asset for some weight that sets small changes of the derivative's value to be netted out with the replicating portfolio. Consequently, as holding this **weight should be defined by the point of tangency**, the weight is simply the first derivative of the instrument's value at the point of current spot price. Thus, its is just:

$$w_t = \frac{\partial \Pi(\cdot)}{\partial S} \tag{22}$$

, for each point of time t, deeming the decay pattern of the derivative's value to be stochastically controlled by such a hedge. **This is exactly the delta-hedging rule**, stated in Chapter I.

Furthermore, as in our framework the hedging process is limited to the publicly traded assets only, and as experiments disallow non-linear variables to be hedged out, the described pattern is enough for our baseline that one would use during classical delta-hedging.

It is important to note that **if one decides to include non-linear instruments in the framework too, then the framework itself should not be modified, as it is general enough**. However, the baseline, formulated here, should be adjusted in order to maintain comparability.

Section 4.2. Weights calculation for forward.

The forward as an instrument represents the trivial case that can be hedged with zero volatility, being linear instrument. As the forward's price is zero at inception by our design in experiment due to setting the strike of the forward to be F from Section 3.7 (however, if not so, the logic of the hedging does not change), the price of the forward at time $t > T_0$ is simply the present value of the forward's strike and the strike of the new forward, if done at the new market price with the same maturity, as the time till maturity left from the initial forward's time. Thus, it can be calculated as following:

$$\mathscr{F} = e^{-r_t^{diff}(T-t)}(e^{r_t^{diff}(T-t)}S_t - F) = S_t - e^{-r_t^{diff}}F \tag{23}$$

, where

$\mathscr{F}$ is the price of the forward,

S_t is the price of the base asset at time t,

T is time till maturity of the derivative,

r_t^{diff} is the rates difference, as outlined in Section 3.9

.

Consequently, using the rule (22), the hedging weight is the first derivative, which gives us:

$$w_t^{\mathscr{F}} = \frac{\partial \mathscr{F}}{\partial S_t} = \frac{\partial(S_t - e^{-r_t^{diff}}F)}{\partial S_t} = 1 \tag{24}$$

, proving the fact that forward is a linear derivative, **from which follows that we should set all the hedging weights in (8) to 1**.

The derivation supports the idea of how usually the hedging for forwards is determined, as it classically involves **borrowing money, buying 1 unit of asset (stock, other currency and etc.) and collecting the cash flows from this asset** (dividends and deposit interest rate for the examples above, respectively).

That being the case, such a baseline allows us to check the overall logic behind the model's optimization and make sure that the architecture is complex enough to find the solution to such a simple case, but not too complex to overfit.

Section 4.3. Weights calculation by Black-Scholes Model.

The next stepping stone in the baseline construction is finding the most close-to-the-ground idea of the options' hedging weights determination. It should follow from the Black-Scholes-Merton model, which is usually used as the starting point for any options discussion.

The model uses several assumptions about the markets behavior and experiment setting, which should be considered here, as **all of them are not required for the target solution, offered by this study**.

First of all, Black-Scholes-Merton model assumes <u>frictionless markets</u>, meanining that there are no transaction costs for hedging operations whatsoever — neither for the buying and selling the base asset, nor for the borrowing and lending of the funds. **The assumption is relaxed in the offered framework, as outlined in Section 3.5**.

Secondly, authors assumed <u>normality of returns</u> (equivalently, <u>log-normality of prices</u>) of the base asset. From such an assumption the derivation the formula that will be touched further on follows. However, while it allows for the simplification of the derivations, this premise lacks practical background to support it — many later authors (Officer, 1972; Stokie, 1982; Sheikh & Qiao, 2009) found out that the normal returns is not the case for the financial markets. More precisely, Black & Scholes outlined the pattern as stock prices, following Geometric Brownian Motion process, which is based on the assumption above. Thus, the fact that **presented research relaxes this assumption is quite crucial for the correctness of the market data modeling**, and provides us with the hope that the study will outperform this simple baseline.

Finally, the model works with the assumption that the standard deviation of underlying's returns throughout the lifetime of the derivative stays the same, which is definitely not the case for the market's random variables, as shown by (Engle & Ng, 1993). From this it follows that **relaxing this assumption too by the proposed framework also should bring positive improvements** to how the trading desks hedge their derivatives.

Jumping straight to the result of authors' paper's findings, the price function itself looks as following:

$$\mathscr{C}_t = N(d_1) \cdot S_t - N(d_2)e^{-r_t^{diff}T} \cdot K_{\%}S_t \tag{25}$$

$$\mathscr{P} = N(-d_2)e^{-r_t^{diff}T} \cdot K_{\%}S_t - N(-d_1) \cdot S_t \tag{26}$$

, where

$$d_1 = \frac{\log(\frac{1}{K_{\%}}) + (r_t^{diff} + \frac{\sigma_t^2}{2})T}{\sigma_t\sqrt{T}}$$

$$d_2 = d_1 - \sigma_t\sqrt{T}$$

$\mathscr{C}$ is the price of the European call option (see Section 3.7),

$\mathscr{P}$ is the price of the European put option (see Section 3.7),

S_t is the price of the base asset at time t,

T is time till maturity of the derivative,

r_t^{diff} is the rates difference at time t, as outlined in Section 3.9,

σ_t is the annualized assumed standard deviation of the underlying's returns (i.e., volatility),

$K_{\%}$ is strike of the option, expressed as the % of spot price (e.g., $K_{\%} = 100\,\%$ means $K = S$),

$N(\cdot)$ is cumulative distribution function for the random variable $n \sim N(0,\ 1)$

. As can be noted from formula, r_t^{diff} and σ_t do not change over time after the time of pricing t, being just scalar constants.

Accordingly, following the logic of Section 4.1, the hedging weights rule turns out to be quite simple too:

$$w_t^{\mathscr{C}} = \frac{\partial \mathscr{C}}{\partial S_t} = \Delta_{\mathscr{C}} = N(d_1) \tag{27}$$

$$w_t^{\mathscr{P}} = \frac{\partial \mathscr{P}}{\partial S_t} = \Delta_{\mathscr{P}} = N(d_1) - 1 = \Delta_{\mathscr{C}} - 1 \tag{28}$$

, noting that $\Delta_{\mathscr{C}}$, $\Delta_{\mathscr{P}}$ are the "deltas", as such first derivatives are usually called.

It is of great importance to note here that in practice most of the non-bulge-bracket banks are still sticking for this methodology to determine hedging strategy, thus, if the presented baseline is outperformed by the proposed framework, it already created value added to the practitioners.

Section 4.4. Weights calculation by Heston Model.

While the assumptions of the Black-Scholes-Merton model have been eased sequentially by different authors, in this paper, for the purposes of illustration, only the constant volatility and transaction costs will be tackled, as trading desks rarely go further in their practical applications, besides implementing the SOTA algorithms[1].

As was stated previously, the Black-Scholes-Merton model, outlined in Section 4.3, is derived from the assumption that the stock price follows the Geometric Brownian Motion with constant variance:

$$\frac{dS_t}{S_t} = r_t^{diff} dt + \sigma dW_t \tag{29}$$

, where

S_t is the price of the base asset at time t,

r_t^{diff} is the rates difference at time t, as outlined in Section 3.9,

σ is the annualized volatility,

W_t is the Wiener process

. The expected value of the stock price change (i.e., stock return) is set to be:

$$\mathbb{E}(\frac{dS_t}{S_t}) = r_t^{diff} \tag{30}$$

due to risk-neutrality of the measure. It can be noted that the process implies constant σ.

That being mentioned, the constant volatility issue can be easily tackled via Heston model (Heston, 1993), which assumes that during the lifetime of the derivative, the variance of returns follows the Ornstein-Uhlanbeck process:

$$\frac{dS_t}{S_t} = r_t^{diff} dt + \sqrt{v_t} dW_t^S \tag{31}$$

$$dv_t = \kappa(\sigma - v_t)dt + \xi\sqrt{v_t} dW_t^v \tag{32}$$

$$Cov(dW_t^S, dW_t^v) = \rho dt \tag{33}$$

, where

ξ is the volatility of volatility,

κ is the mean-reversion speed of the Ornstein-Uhlanbeck process,

σ is the long-term average annualized volatility (which is usually calibrated into market),

ρ is the correlation of the randomness between spot and volatility,

[1] https://www.risk.net/derivatives/6691696/jp-morgan-turns-to-machine-learning-for-options-hedging

W_t^S, W_t^v are the correlated Wiener processes of spot price and volatility, respectively

.

However, while the presented approach works in well the non-constant volatility, now the risk-neutral expectation for the derivative cannot be found directly. Therefore, here we can use the Monte-Carlo method — simulate the paths according to the Heston model, and then calculate the first derivative numerically simply by:

$$\frac{\partial \mathscr{C}}{\partial S_t} = \frac{\mathscr{MC}(S_t + \delta) - \mathscr{MC}(S_t - \delta)}{2\delta} \tag{34}$$

, where

$\mathscr{MC}$ is Monte-Carlo algorithm for calculation of the option price via simulation.

While such an approach is computationally expensive, the modeling by it will allow to match correctly the offered framework with usual algorithm, used by advanced investment banks.

Section 4.5. Gamma decision rule.

Last, but not the least topic in relation to the classical approach baseline is the attempt to tackle the transaction costs imperfection in the models, presented above.

In order to solve the issue of accounting for the transaction costs, we can use the simple approximation of the decision rule that will allow seamlessly adjust our hedging behavior via a heuristic. For that let's dive into the logic behind hedging an option.

When the spot price changes, if we were fully delta-hedged, our first-order loss is fully dispensed from by the definition of the delta-hedge. However, as the option is non-linear, its value change is goes beyond the first-order change, being defined as:

$$d\mathscr{C} = \Delta dS + \frac{1}{2}\gamma (dS)^2 \tag{35}$$

, where

$\gamma = \frac{\partial^2 \mathscr{C}}{\partial S^2}$ is an option "greek", called gamma — second-order derivative by spot price

.

Therefore, while the portfolio is delta-neutral, the gamma part remains unhedged — and **that is why we need to shift delta, dynamically rebalancing the portfolio — in order to constantly be protected as much as possible from the second-order convexity**:

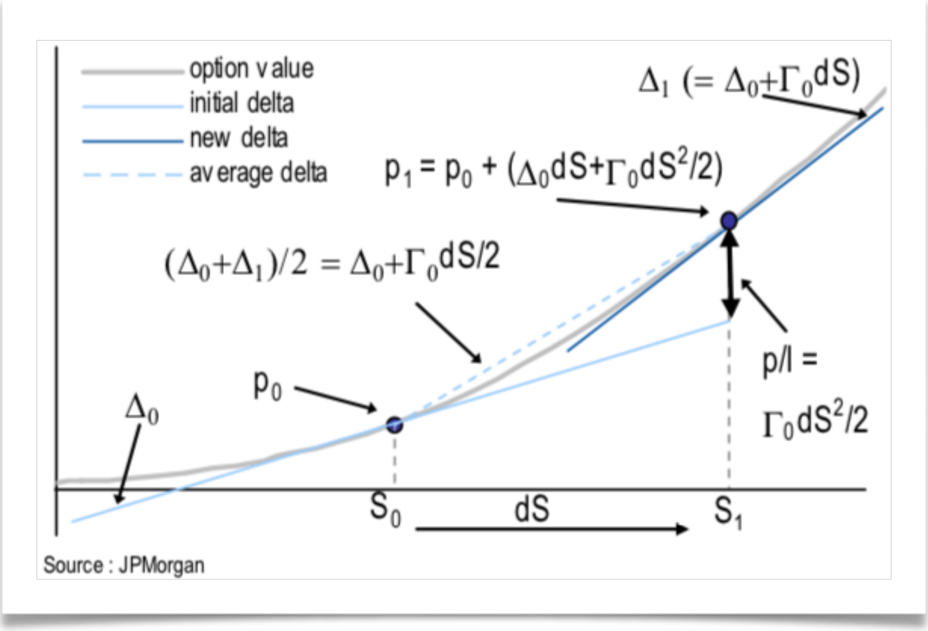

Figure 4: The gamma convexity effect. Source: JP Morgan.

.

Consequently, a after market data has changed **<u>delta-hedging step, only makes sense, if we are expected to lose more from it than the transaction costs paid</u>**. And what is our expected loss on the unhedged portfolio? It is the change from second-order movements of the option value, because due to delta-hedging process portfolio of the derivative plus hedging portfolio is always delta-neutral ($\Delta = 0$). Thus, the decision rule should look as following:

$$\begin{cases} w_{t+1} = \Delta_t, \zeta(dw_t) < \frac{1}{2}\gamma(dS)^2 \\ w_{t+1} = w_t, \; otherwise \end{cases} \tag{36}$$

, where

ζ is the function of transaction costs from making a deal at the market

.

Summarizing, such rule allows to control the hedging procedure from the simple heuristic in order to proceed with the delta-hedging step, only if it makes sense from the potential loss from unhedged position's perspective.

Chapter V. Reinforcement learning baseline.

Section 5.1. Reinforcement learning formalism.

As was outlined in the literature review, provided by Chapter II, the modern authors concentrate their statistical hedging findings in the reinforcement learning area. Therefore, as this paper aims at classical deep learning approach findings, the reinforcement learning baseline should be also accounted for.

Due to the fact that reinforcement learning is way more general approach to the statistical approximation, **it might be expected that a well-defined deep learning task with the loss that can have sustainable gradients outperforms the reinforcement learning baseline due to the excess generality and usual instability of the latter**.

Moreover, here we are working with the **continuous actions space**, as a weight can, in the overall task, can take any real number $w \in \mathbb{R}$, or, at least, number of weights is finite (due to the price and quantity of spot units), but very large.

Reward is the negative MSE between the derivative's payoff and the hedging portfolio PnL (see Section 3.5). The MSE is taken negative in order for the reward to be maximized, thus, deeming it to be actually a Cost.

But furthermore, there are the issues that concern Reward definition above. Indeed, as the Reward can be determined only at the end of the derivative's maturity, we are dealing with **sparse reward game**, which is a **great issue for the continuous-time SOTA models**. Moreover, as our Reward is actually a Cost, then another issue arises for the design of the optimization of RL algorithms — the datapoints without the signal are denoted by zero reward, which is the lower bound for Reward, but in case of the Cost it **will be an upper bound**.

Thus, let's reformulate the Cost into Reward by setting it positive, but increasing, when MsE declines by taking the ratio:

$$Reward = \frac{1}{MSE(PnL_{target}, PnL_{hedge})} \tag{37}$$

.

Now, let's transform our business task into the RL framework:

- Action space is the weights of the replicating portfolio that can be attained (see Sections 3.4-3.5) by our $Hedger$. It follows that $Actions = \{W \in \mathbb{R}^{A \times T}\}$;
- Reward is the inverse MSE, minimizing the distance between the payoff of the derivative and the hedging PnL by maximizing the reward.
- Observation is the market data and all other available features.

Let's discuss the ideas behind approach to environment creation, while trying to tackle the issues, outlined above, by the careful design of the environment implementation.

Section 5.2. Environment.

Undoubtedly, there are two basic ways to consider the task, outlined above. First of all, we can create environment, that is required to follow the Markov Decision Process (MDP) formalism, in a classical manner — it will work with single datapoints of the market data, returning a weight autoregressively step by step. Therefore, such implementation works with the following sets at each step:

$$(o_i, r_i, is\_done_i) \tag{38}$$

, where

o_i is the observation of the current state of the game,

r_i is the reward for transiting to the new state i,

is_done_i is the flag, whether the derivative have matured

.

However, due to the issues, outlined in Section 5.1, such approach would need to work with **sparse rewards**, because for all points before the maturity of the derivative in the sample, the actual reward will remain unknown, thus, most of the points will look like:

$$(o_i, 0, 0) \tag{39}$$

.

Consequently, we can reformulate MDP into the full sequence mode. In other words, the step of the game will now be defined not by one point in market data, but rather by one particular derivative, created by the Section 3.7 guidelines. Therefore, each step's observation will be a sequence of o_i with the reward of $\frac{1}{MSE}$ and flag $is\_done = 1$ always:

$$(\{o_t | t \in [T_0; T)\}, \frac{1}{MSE}, 1) \tag{40}$$

. Such an approach would **help to dispense from the issue of the sparse reward, and reformulate the learning task into sequence-based**.

Finally, the o_T data, which is not included into learning for both of the environments, is used to determine the reward of $\frac{1}{MSE}$ by the Section 3.5 procedure inside the step function, but not returned to the model for observation.

Section 5.3. Algorithms used.

As the paper does not target diving deep into reinforcement learning techniques, concentrating on deep learning solution, only basic algorithms will be applied. It should be noted that the approach can be definitely tuned in order to obtain more advanced reinforcement learning results, however, it is out of the scope of this study.

First of all, let's look into the formulation of the task, offered by Section 3.5's PnL calculation from the reinforcement learning perspective. If we consider the reward as negative MSE, as is actually done in deep learning approach (as our loss is positive MSE), then we can denote our reward-to-go G_t as:

$$G_t = \sum_t r_t = -(\mathbb{R}W - PnL_{target})^2 \tag{41}$$

, where

$\mathbb{R}$ is the matrix of asset universe returns from Section 3.1,

W is the hedging weights matrix (8)

.

Therefore, the deep learning gradient step is defined as simply:

$$\theta_{new} = \theta_{old} - \alpha \nabla_\theta (\mathbb{R}W - PnL_{target})^2 \tag{42}$$

, where

θ_i is the parameters of the neural network at optimization step i,

α is the learning rate

.

As such optimization depicts some resemblance of Monte Carlo policy gradient optimization, it actually looks like a REINFORCE (Williams, 1992) algorithm's optimization step, which is defined as:

$$\theta_{new} = \theta_{old} - \alpha \nabla_\theta \log[n_\theta(W)](\mathbb{R}W - PnL_{target})^2 \tag{43}$$

, where

$n_\theta(W)$ is the probability density function of our "policy", i.e., the neural net that outputs hedging weights

.

Thus, we can easily note that **our proposed approach is just a modified REINFORCE, but without the entropy adjustment** — consequently, in order to compare the approaches correctly, we can just take more advanced algorithms.

First of all, the study considers **Soft Actor-Critic** (Haarnoja, 2018), being the SOTA approach in the continuous control sphere.

However, as the **SAC is well-known to work poorly with the sparse reward**, and the reward signal design is out of the scope of current approach under consideration, the research compares the findings with the application of **Proximal Policy Optimization** (Schulman, 2017) by discretizing the weights into the finite space.

Therefore, such approaches will be used to compare the suggested approach of modified REINFORCE without the entropy adjustment, as i**n financial markets, due to the linearity of the utility function for the risk-neutral individual, there is no need to explore**, as, in our case, the minimization of PnL variation is a well-defined optimum.

Chapter VI. Model architecture.

Section 6.1. Model architecture.

By the business task defined above, the weights of the replicating portfolio are generated autoregressively, taking features autoregressively too. And important issue here is that **the features are market data-related, thus, are time sensitive and should be strictly protected from potential data leaks**, which is embedded in the model architecture, proposed in this paper.

Therefore, the nature of the task dictates two potential architectures — a **baseline in deep learning formulation will be implemented via Multilayer Perceptron** (Haykin, 1994) that will take previous weight and features as an input and return the next weight. This will undoubtedly protect the model from data leaks, but will lack ability to account for the step before $t - \epsilon$ to generate weight for the time moment t.

Another approach for such formulation, which is considered target solution in this article due to its robustness to the data is recurrent neural network that can support keeping "long-term memory". Therefore, **the Long Short-Term Memory (Hochreiter & Schmidhuber, 1997) architecture is used as the core solution that shows great results in this task**:

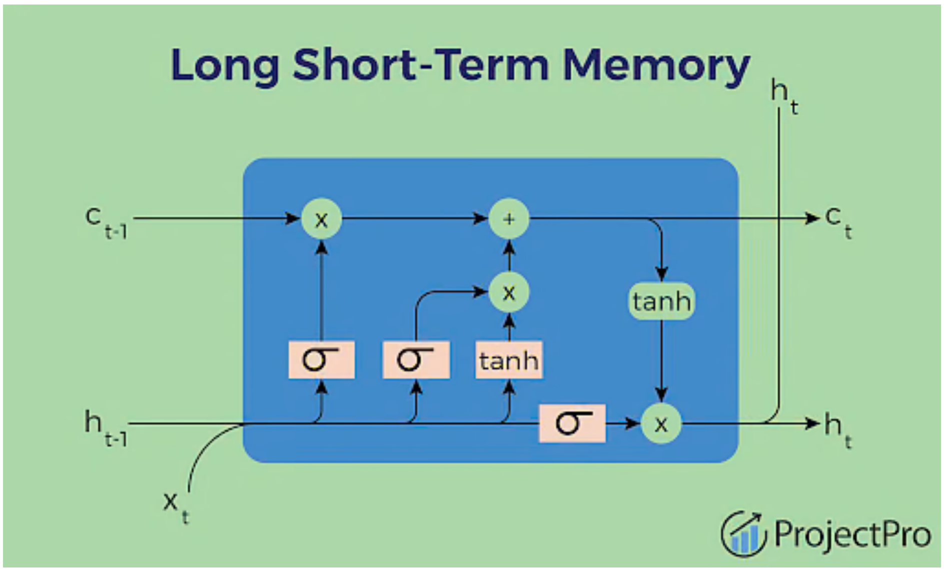

Figure 5: LSTM Architecture. Source: www.projectpro.io.

.

Consequently, at each step of **recurrent network procedure the optimization is protected from the data leaks due to the sequential nature** of the model's inference. The Self-Attention mechanism is not added in order to avoid look-ahead bias, and to support the maximum available datapoints, which brings significant results in the findings of this paper.

Thus, the **hedging weights of our interest is just the hidden states** at each step of recurrent inference:

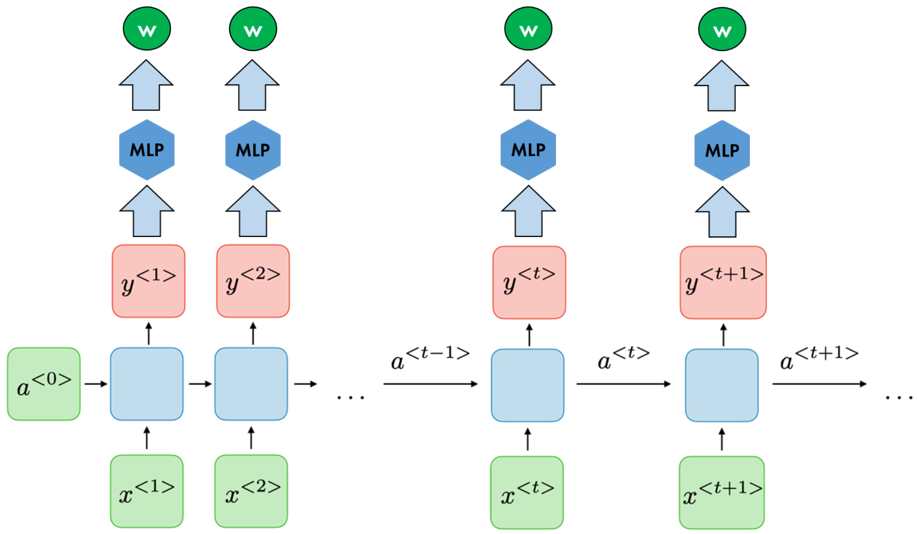

Figure 5: Weights output. Source: stanford.edu + sketch of the author.

.

However, better results are attained by imposing the 2-layer MLP above the LSTM's output, which consists of $hidden\_dim$ parameters, thus, such approach is used.

The model works as following:

- It receives the input features of the current market state;
- Further, the model outputs the hedging weight as the MLP's output above the hidden_state for the next step — **the $Hedger$ cannot trade at the price level that it has just received, this data is already realized, and the actual price, at which we are going to trade is unknown**. This is controlled by the PnL calculation procedure, as outlined in Section 3.5;
- The hidden_state (not the MLP output) is then inputted at the next step together with the features of the new realized market data — same ones that are used for the PnL calculation of the outlined above hedging weights output.

The final output matrix looks as following:

$$\begin{pmatrix} w_{t_0}^{a_1} & w_{t_1}^{a_1} & \dots \\ w_{t_0}^{a_2} & w_{t_1}^{a_2} & \dots \\ \vdots & \vdots & \vdots \\ w_{t_0}^{a_A} & w_{t_1}^{a_A} & \dots \end{pmatrix} \in A \times (T-2) \tag{44}$$

, because the remaining two vector-columns are zeroes, as defined by Section 3.5, and are appended later on in PnL calculation step, allowing to obtain (8).

Another important part of the architecture is an artificial feature that was generated in order to assist the model optimization — **the input layer also gets the time till maturity of the derivative** that helps the model with the prediction of the hedging weights, as the smaller the time period left, the less price uncertainty is left.

This concludes the architecture advancements in this paper, as it determines the framework and optimization task definition rather than experiments with particular advanced architectures. A potential improvement in this matter might be application of the Transformer with Self-Attention. The **more advanced model could work as the one that looks at the previous *N* datapoints with some attention weights, outputs a hedging weight and shifts the window by 1 datapoint, dropping the first one**. However, as the architecture would require the window to look at, the total number of samples, available for optimization, is expected to drop severely by such advancement.

Section 6.2. Optimization.

The model optimization is done in the classical deep learning method approach.

First of all, the PnL is calculated as per Section 3.5 guidelines, keeping all the gradients in the pytorch framework from the MLP / LSTM output. Then the $nn.MSELoss$ is applied in order to compute the MSE between the target, computed beforehand in the dataset creation (Section 3.4) and having no effect from the model parameters, and our **customized output of the neural network, which is fed through the special $get\_pnl(\cdot)$ method of the neural network class**.

Additionally to that, before the feed-forward step of the optimization, for the purposes of stabilizing the gradients, the input features are fed through the BatchNorm / LayerNorm layer. While the first seems to be a bit more reasonable from the financial task standpoint, the latter shows better practical results, which is outlined in the ablation study of the Chapter VIII.

Finally, the overall optimization is performed via gradient descent of $torch.optim.SGD$ or $torch.optim.Adam$ optimizers, having learning rate schedule control via Cosine Annealing.

Chapter VII. Results.

Section 7.1. Data collection.

For the purposes of this study, as it concentrates on the FX derivatives for the illustration of the proposed framework abilities, the data for **Moscow Exchange prices of USDRUB and EURRUB** currency pairs are used. The paper does not consider another price sources for this instruments in order to not include some additional "alphas" from the infrastructure of some market participant.

Overall, collected dataset includes **1 minute bid-ask data from 2017-01-03 07:00:00 to 2024-04-30 15:51:00, represented by the 1,488,719 (1.5 mio) datapoints**.

The FX instruments were chosen merely due to their simplicity of modeling in terms of data collection, as they do not produce any cash flows, do not have a maturity, and are easily obtainable for practically any time point in the past. However, as was mentioned before, framework does not require any adjustments in order to work with any other asset class.

Moreover, FX derivatives were chosen to be in the focus of this study also because of their dominance across all the derivatives traded by the Russian investment banks, according to the Central Bank of Russia data[2], as of 2022, especially in the area of FX options that are central to the current study:

Тип контракта	2021						2022					
	июль	август	сентябрь	октябрь	ноябрь	декабрь	январь	февраль	март	апрель	май	июнь
FX Forward, NDF	12 299	9 623	16 061	15 486	19 920	16 907	16 747	12 484	1 497	2 365	1 615	1 838
FX Swap	18 326	18 595	18 667	18 555	25 187	30 898	20 797	18 978	3 133	2 779	2 061	3 176
FX Option	2 064	1 937	2 088	2 098	1 978	2 079	1 630	2 071	80		37	56
Cross-Currency Swap (CCS)	85	54	35	65	113	62	56	101	10	16	6	6
Interest Rate Swap (IRS)	368	462	353	608	520	474	306	313	2 010	47	56	23
Interest Rate Option (Cap, Floor)	185	199	286	190	160	281	78	118		24	16	53
Forward Rate Agreement (FRA)	13	5	18	26	9	8	5	3				
Swaption	6	16	5	2	6	1						
Bond Forward	79	39	93	104	109	84	87	38	9	50		4
Bond Basket Option	6	12	20	44	19	528	376	504	25		3	59
Equity Forward	2 130	1 943	2 113	1 707	1 637	2 303	1 103	1 243	290	709	825	1 874
Equty Option	110	127	178	198	146	327	186	117	22	247	269	490
Commodity Swap	414	421	469	473	632	610	652	475		50		1
Commodity Forward	409	709	526	419	741	451	322	404	55	4		9
Commodity Option	197	201	207	171	65	53	623	128				
Credit Default Swap (CDS)		14	19	3	4	16	3					
По всем типам	**36 691**	**34 357**	**41 138**	**40 149**	**51 246**	**55 082**	**42 971**	**36 977**	**7 131**	**6 291**	**4 888**	**7 589**

Figure 6: The volume of OTC derivatives traded. Source: Central Bank of Russia.

.

However, as the study aims to account for the transaction costs, while determining the optimal hedging strategy, the usual price of trades inside the trading day is not enough. Moreover, as

[2] https://www.cbr.ru/Collection/Collection/File/46341/OTC_derivatives_market_2022.pdf

we want to simulate actual trading of a desk, the **correct backtest requires the prices that were actually available for making deals**. Thus, usually used and publicly available closing prices are not good enough — one cannot trade at the closing level, which is purely technical metric.

Therefore, for the purposes of this study the **actual bid and ask of the Volume-Weighted Average Price (VWAP)**[3] is used, which is sampled from the orderbooks of the Moscow Exchange. The VWAPs are sampled for 500k of the notional (e.g., for buying 500,000 USD for *ask* level of RUB), which is chosen as a representative delta-hedging volume for an average FX option flow for a corporate client. The VWAP price dynamics of USDRUB, resampled to 30 minutes for the purposed of illustration, looks as following:

Figure 7: Historical bid-ask prices for USDRUB (2017-2024). Source: MOEX.

.

More precisely, the bid-ask dynamics looks as outlined by the chart below, limited to the recent time from 2024-03-01 in order to highlight the actual bid-ask difference:

3 https://www.investopedia.com/terms/v/vwap.asp

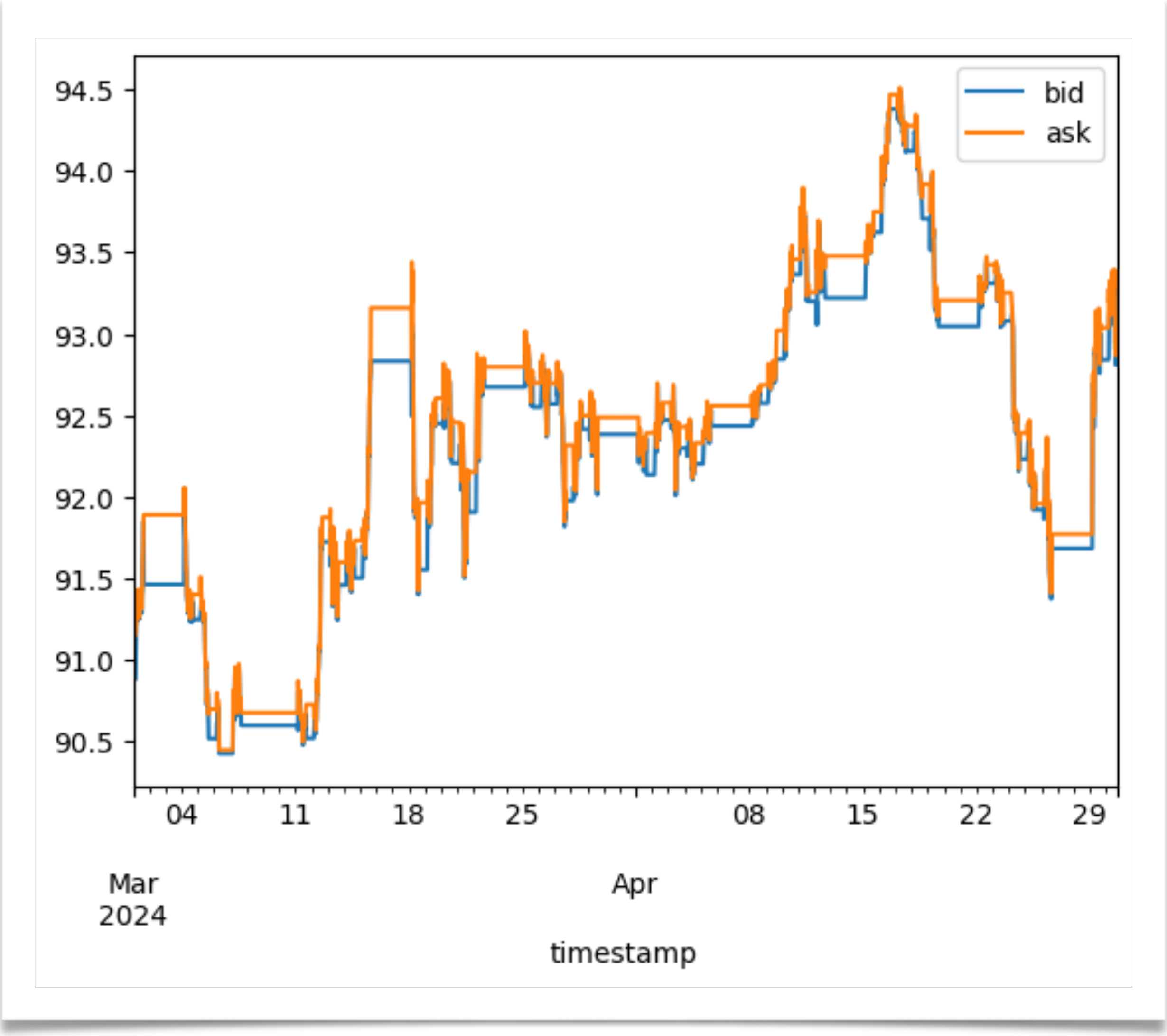

Figure 8: Historical bid-ask prices for USDRUB (03.2024-04.2024). Source: MOEX.

.

Overall, the **bid-ask spread has been pretty stable across the main parts of the dataset, however, diverging steeply during the unexpected events** (like 2020's and 2022's), which provides the additional source of uncertainty, which is not expected to be covered by the model though. The $ask - bid$ variable's change for USDRUB is depicted by the graph below:

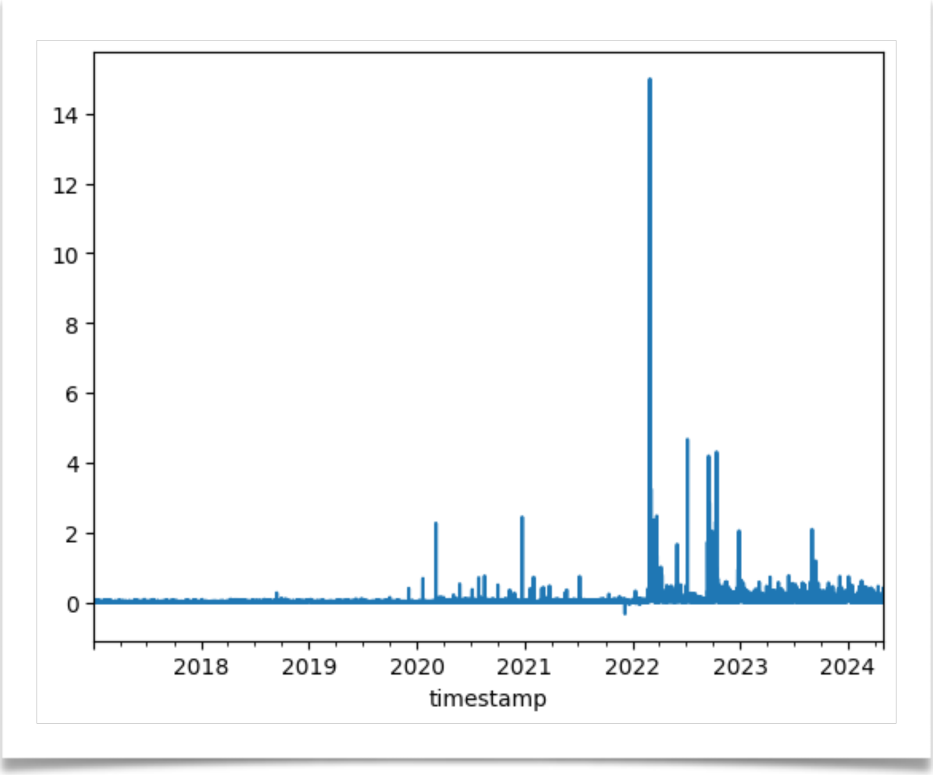

Figure 9: Historical bid-ask spread for USDRUB (03.2024-04.2024). Source: MOEX.

.

The average level of bid-ask spread was 0.0772 RUB (0.1%) across the considered time interval with the standard deviation of 0.2865 RUB (0.38%). The bid-ask spread's behavior across stable times (before 2019-10-01) can be seen below:

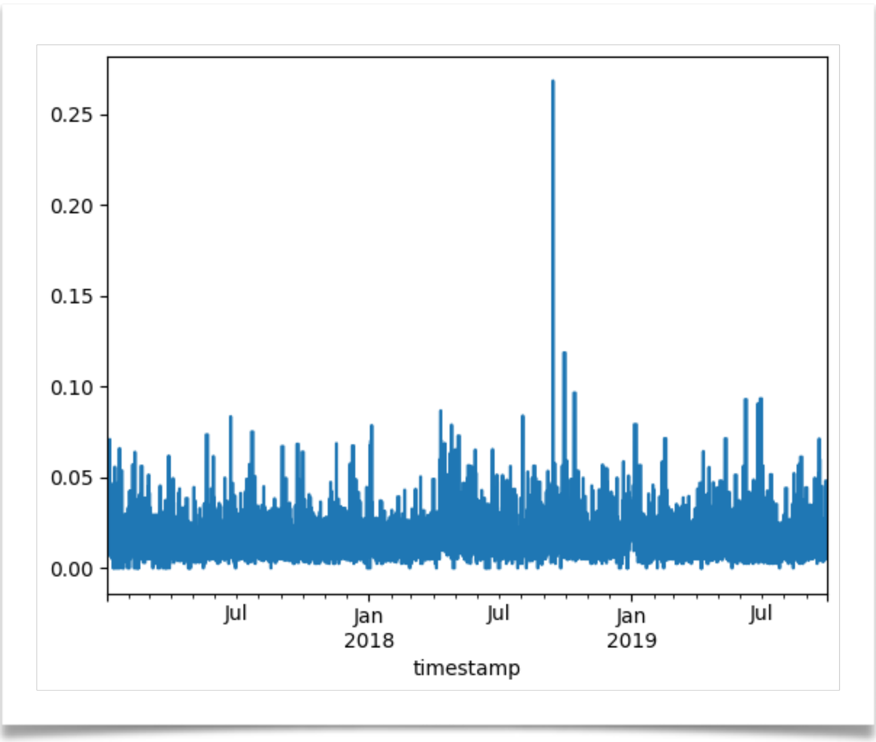

Figure 10: Historical bid-ask spread for USDRUB (01.2017-10.2019). Source: MOEX.

.

The overall calculations will be performed in RUB, thus, the PnL of the replicating portfolio will be denominated in RUB too. As it can be easily converted to USD, it makes little difference for the study's purposes, however, while the Russian investment banks' operating currency is RUB, such a calculation will be a bit more practically consistent.

As two assets are used, the correlation analysis might be of help to determine, if the quotes are collected correctly. We can note pretty high correlation of the mid prices, which is definitely an expected result:

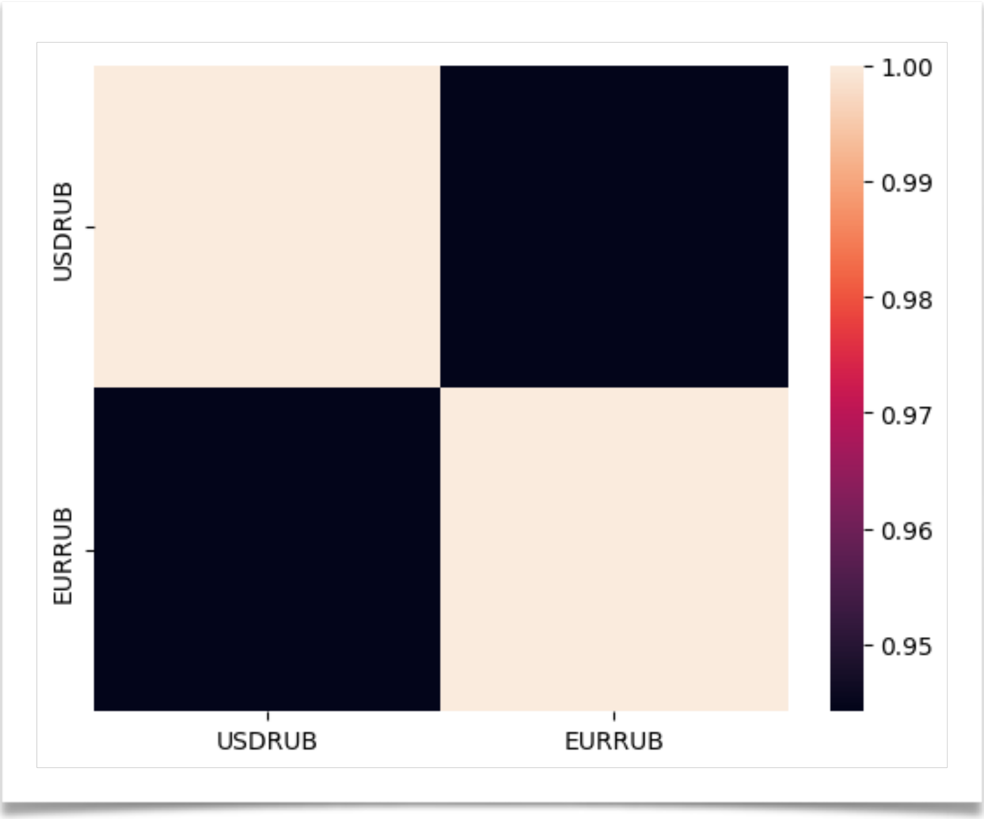

Figure 11: Correlation USDRUB/EURRUB (03.2024-04.2024). Source: Calculations of the author.

The collected datapoints constitute **1 minute quote frequency**, as diving deeper into the higher frequencies would drastically complicate the computational resources needed. As the trading desk is assumed to hedge at non-HFT frequencies — as they require reformulation of the task and orderbook analysis, which is usually done not at the hedging desk dividion — such a dataset is frequent enough. The **experiments with lower frequencies, attained by resampling, are outlined in Section 7.7**.

Section 7.2. Derivatives creation.

Following the methodology of Section 3.4, the derivative object is created inside the $Hedger$ service. We feed into the object's $payoff(\cdot)$ method the corresponding $spot\_path$, which is the dataset's point, created according to the Section 3.7.

For the sake of illustration, all of the experiments use $N\_DAYS = 5$ hyperparameter, thus, considering the **derivatives with maturity of 5 days**. In all of the experiments below, the strike for the forward is used at the level of fair forward price, which sets price of the forward to zero.

In regards to the European Call, **all experiments use** $K_{\%} = 100\,\%$, however, the presented framework allows to expand the experiments to any strike, needed, however, requiring the retraining of the model, which usually took 4 - 4.5 hours of wall clock time.

The example of the object's representation is given below:

```
EuropeanCall(strike=61.0165, term=0.0136986301369863, spot_ref=61.0165)
```

Figure 12: Example of the derivative object. Source: Implementation by the author.

.

Section 7.3. Texts processing.

In order to demonstrate the universality of the proposed framework, the text embeddings were added as an additional feature for the $NeuralHedger$ to work with.

The **texts were collected from 11 Telegram channels, related to the Russian market news for the period from January 2020 to October 2022**. The data sources are presented in the table below:

Source Name	Media Type	Link
War_Wealth_Wisdom	Asset Manager	@warwisdom
The Movchans	Asset Manager	@themovchans
VTs	Asset Manager	@v_tsuprov
Sky Bond	Asset Manager	@skybond
RSHB Invest	Asset Manager	@RSHB_Invest
Bitkogan	Blog / Asset Manager	@bitkogan
Headlines QUANTS	News (Global)	@headlines_quants
RSM Signals	News (Russia)	@cbrstocks
MMI	News (Russia)	@russianmacro
Cbonds	News (Russia)	@cbonds
Alfa Wealth	Private Banking	@alfawealth

Table 1. Source: Methodology of the author.

.

The texts were lemmatized, dropping the stopwords with Russian stopwords of nltk[4] library, and processing the words with morphology framework pymorphy2[5].

As the NLP analysis is out of the scope of this study, and the texts are added as merely a significant feature, deemed to outperform the results of the plain price-based learning, the pre-trained embeddings were used without the comparison with the training the model from scratch or using an advanced pre-trained network, which can be a potential improvement for the further research on the base of the presented one.

The **embeddings for the words are implemented via taking model "cc.ru.300" of fasttext[6] library**, which was pre-trained on the Russian tweets by the authors of this esteemed NLP models' source. Again, as NLP experiments are out of the scope of this paper, the model was taken arbitrary, however, showing great results, which will be presented in Section 7.7.

The texts are restricted by the **length of 128**, which is an arbitrary chosen hyperparameter due to the balance between the computational resources needed and the completeness of the market news representation.

Furthermore, chosen texts are joined by the datetime sequentially, meaning that to each price datapoint corresponds exactly one text, filled forward, until the new text is available. Such an approach was chosen due to the logic that each Telegram channel bears the information for the model, which is equivalent to the information from some other channel. This can also be tested in the further improvements on the presented study.

Finally, the obtained sequence of embeddings is stacked into a tensor, which is further **fed through the special $self.lstm\_text = nn.LSTM(\cdot)$ layer with 2-layer MLP on top** in order to produce a hidden state for each price datapoint. This hidden state is then concatenated to the orderbook features, and used in the usual course of actions, outlined in Chapter VI.

Section 7.4. Training process.

For the training, the resources of Google Colab T4 GPU were used. The average training time constituted 4 hours 43 minutes, varying due to the various experiments.

The total number of points was determined by the resampling of data, which was done for the frequency of 5 minutes, 30 minutes and 1 hour, additional to the original data of 1 minute, which was extensive for the avilable resources, requiring the learning for the wall clock time over 24 hours for 10+ epochs, which is why it was dropped from the study.

[4] https://www.nltk.org/

[5] https://pymorphy2.readthedocs.io/en/stable/

[6] https://fasttext.cc/

Section 7.5. Yield curves.

As was outlined in the Section 3.9, for the PnL calculation procedure the NSS curve is used. It is implemented as the child class $NelsonSiegelCurve$ of the abstract $YieldCurve$, which implements the curve construction and all the required methods for backtesting.

The parameters for RUB NSS curve are taken from the Moscow Exchange website's part of fitted Zero-Coupon Curve[7]. The data looks as following:

timestamp	B1	B2	B3	T1
2014-01-06 12:21:16	877.951361	-311.324633	51.105265	4.836731
2014-01-08 12:41:22	879.619947	-312.611788	51.560662	4.824178
2014-01-09 18:38:19	876.971884	-320.831620	56.936812	4.448947
2014-01-10 18:38:17	875.118031	-329.716005	61.089853	4.148542
2014-01-13 18:37:26	878.212981	-325.573772	60.784188	4.195013

Figure 13: NSS Curve parameters example, RUB. Source: MOEX.

.

The USD and EUR curves are taken from the open sources of the corresponding central banks data[8] [9].

The example of reconstructed curve for RUB with years till maturity as the x-axis, as of 2024-05-20, is presented below:

[7] https://www.moex.com/a3642

[8] https://www.federalreserve.gov/data/nominal-yield-curve.htm

[9] https://www.ecb.europa.eu/press/pr/date/2007/html/pr070710.en.html

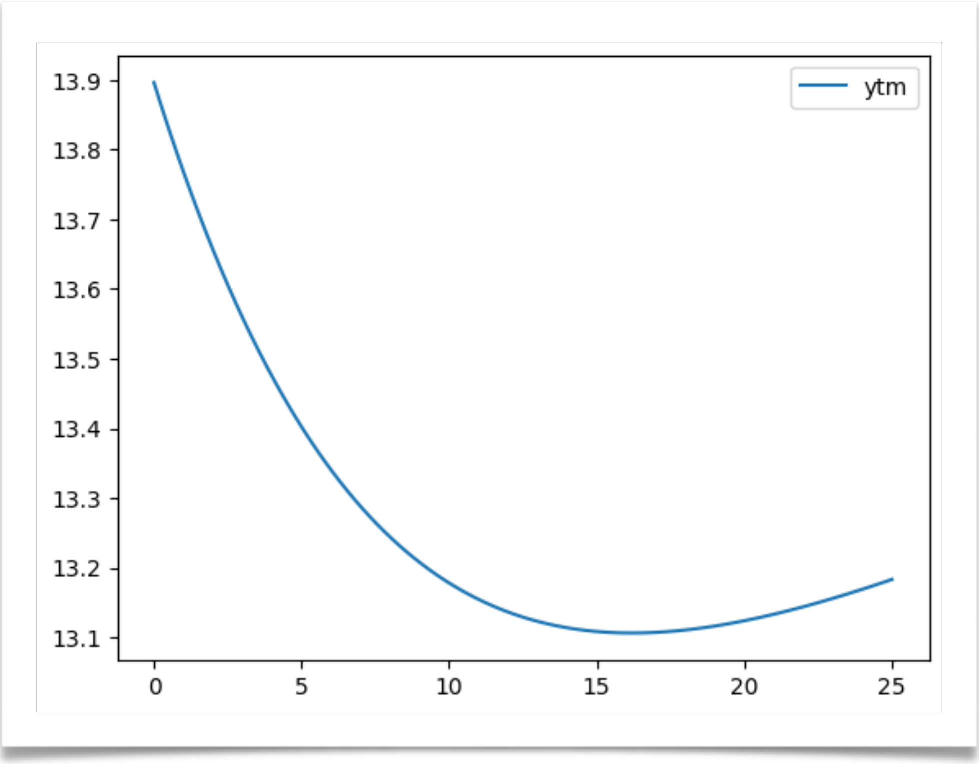

Figure 14: NSS RUB curve (20.05.2024). Source: Calculations by the author.

. We can note that the curve is inverted, which is an expected result for the current market expectations, thus, the implementation is correct.

The result instantaneous forward rate graph for RUB curve looks as following:

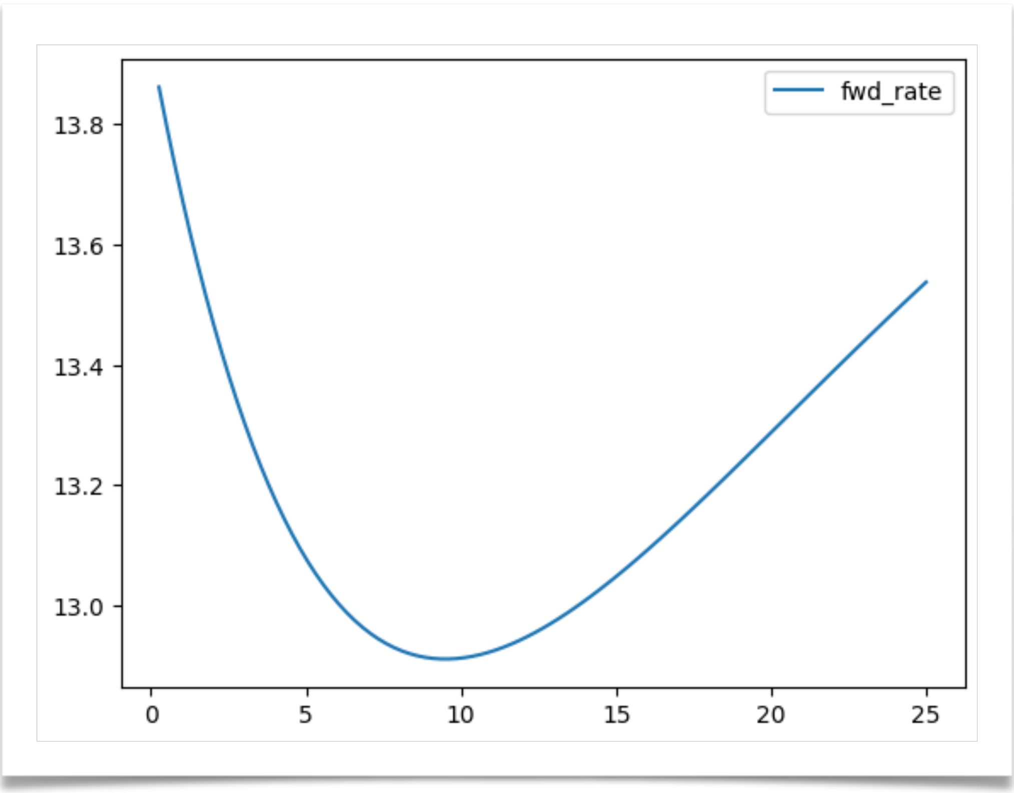

Figure 15: NSS RUB instant forward curve (20.05.2024). Source: Calculations by the author.

.

The correct modeling of interest rates, alongside with the transaction costs of bid-ask spreads was crucial for the correctness of the backtest of the presented framework, that is why this Section concentrates on this part with such rigor.

Section 7.6. Reinforcement learning training.

For the implementation of the reinforcement learning baseline, outlined in Chapter V, **the environment was created from scratch**, in accordance with the modeling framework chosen. Further, the baseline training was implemented, using the **stable-baselines3** library[10], which allows for seamless training of the most well-known algorithms, alongside which the ones that were chosen in Section 5.3.

The obtained training results are outlined in the tables below. The tables present the experiments data on the two approaches to environment creation, as discussed in Section 5.2 — the sparse reward with action for each step and the sequential processing with reward for the full sequence.

Section 7.7. Experiments design.

In all of the experiments, the final set of **features is presented by the past price and rates data** (without look-ahead bias, as discussed in Chapter VI) and nothing else as an input. The experiments that also use the **hidden state for the corresponding text's embeddings**, as discussed in Section 7.3, will be denoted additionally. Thus, the final set of features are outlined as:

Experiment	Bid Price	Ask Price	RUB Rate	USD Rate	Till Maturity	Texts
	FORWARD					
#1	√	√	√	√	√	𝕏
	EUROPEAN CALL					
#2	√	√	√	√	√	𝕏
#3	√	√	√	√	√	𝕏
#4	√	√	√	√	√	√
	ABLATION STUDY					
#5	√	√	√	√	√	𝕏
#6	√	√	√	√	√	𝕏
#7	√	√	√	√	√	𝕏
#8	√	√	√	√	√	𝕏

[10] https://stable-baselines3.readthedocs.io/en/master/

Experiment	Bid Price	Ask Price	RUB Rate	USD Rate	Till Maturity	Texts
#9	√	√	√	√	√	𝕏
#10	√	√	√	√	𝕏	𝕏

Table 2. Source: Methodology of the author.

.

The experiments are presented by the tables that should be read as following. The weights columns are related to the output of the hedger model — the weights of the replicating portfolio for the base asset (one asset case). The **mean PnL, standard deviation and Value-at-Risk**[11] at $\alpha\ \%$ (the measure of the tail risk, which is interpreted as "the portfolio is expected to lose not more than {Value-at-Risk} RUB in $(1-\alpha)\ \%$ cases") are calculated as RUB-nominated measure for the series of:

$$target\_pnl - model\_pnl \tag{44}$$

,
which is exactly the **RUB PnL that a trading desk would see in its books, while using the proposed framework for hedging**, thus, concluding the experiment's goals in terms of checking the viability of the framework.

Finally, each proposed model's difference series are **tested, using t-test**, due to the fact that the series (44) are assumed to correspond to the assumptions of 1) each point of difference being normally distributed; 2) having sample variances that are independent from the sample means, which allows for the Fisher's lemma to be satisfied, thus, qualifying the data to be distributed as **t-Student distribution**. Therefore, the t-statistic is calculated for the significance of means between the (44) for our proposed model versus the (44) for the baseline, outlined in the study by Chapter IV.

Section 7.8. Testing the strategies for forwards.

As was mentioned in the plan of the study presented, while the main part of the research concerns the hedging of the options, **the trivial case for checking the model's performance is hedging a forward, which's solution is expected to be the weights, equal to 1** for all the datapoints (see Section 4.2). Due to the fact that the solution is known, and is trivial, the reinforcement learning baseline is not checked for the forward. The results of the optimization are presented below.

Experiment 1. **Forward Hedging** (5 min frequency, 2017-01-03 to 2024-04-30).

[11] https://www.risk.net/definition/value-at-risk-var#:~:text=Value-at-risk is a,a pre-defined confidence level.

Data per instrument of notional 1 USD	Average weight	Weights Bounds	Mean PnL, RUB	Standard deviation of PnL, RUB	VaR at 5%, RUB	T-Stat
MLP Hedger	0.71317	[0.691; 0.886]	-0.017900	0.17155	0.30507	-4.32
LSTM Hedger	1.00359	[0.908; 1.163]	**-0.013432**	0.05398	0.10112	**3.51**
Baseline	1.00000	[1.000; 1.000]	-0.017169	**0.03894**	**0.07641**	-

Table 3. Source: Calculations of the author.

.

The distribution of the PnL obtained for the LSTM Hedger, which statistically significantly outperforms the baseline, can be found below:

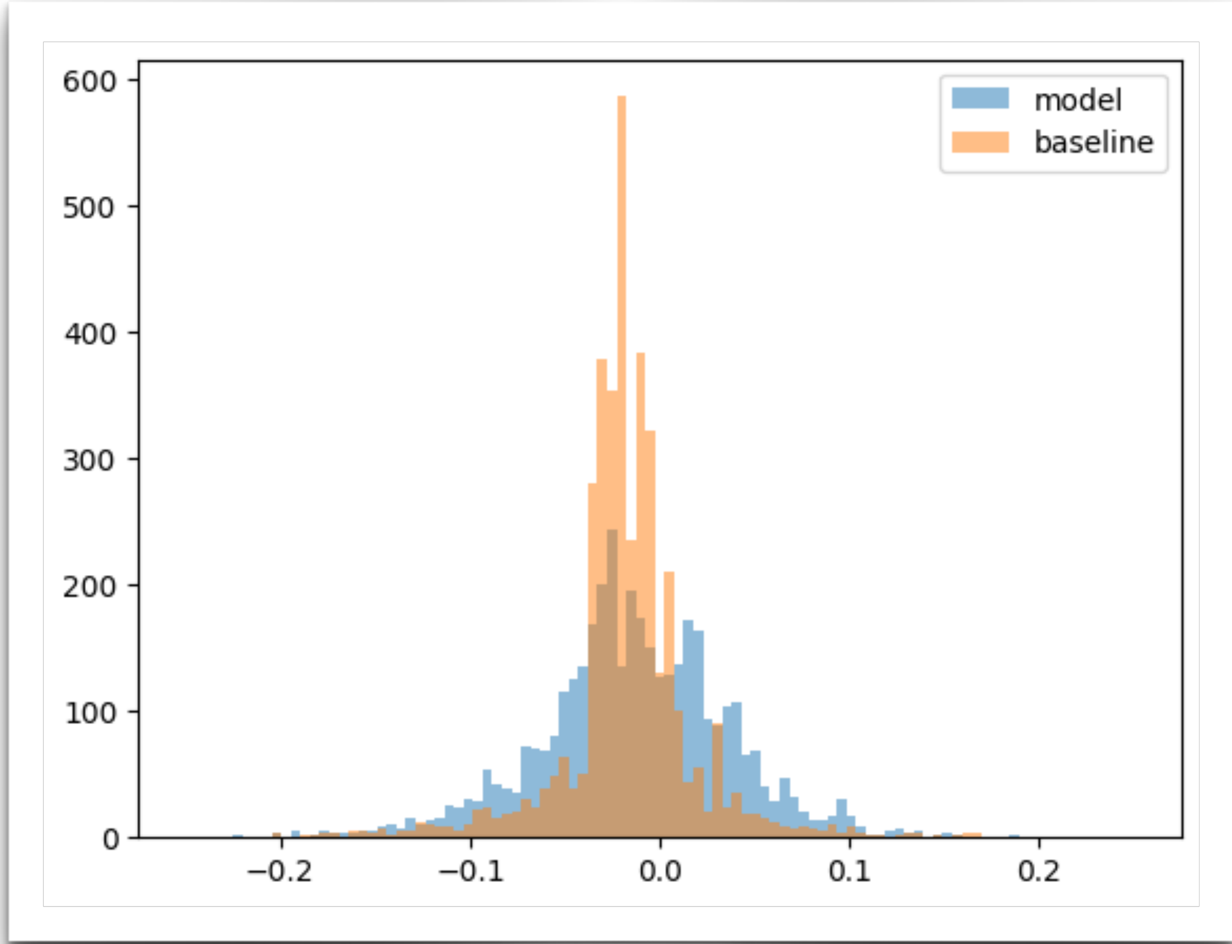

Figure 16: PnL distribution for forward hedging. Source: Calculations by the author.

.

Therefore, we can conclude that the hedging of the forward works well in terms of the optimization techniques, outlined in the paper, due to the fact that the **LSTM model showed statistically significant improvement on PnL at 1% level by the t-test statistic**, while MLP hedger lacks the ability to outperform the baseline.

We can see that the **weights, attained by the model are close to 1**, but deviate a little — this is due to the model's attempt to control transaction costs better, which actually provided us with the significantly better hedging result.

However, how can we explain the attained results? What is the source of such an outperformance, if there exists trivial solution? We can note that the outlined model shows statistically **significant PnL improvement, but has higher standard deviation and VaR**. Therefore, it means that while our results are stable in terms of the statistical learning definition, the financial logic dictates the conclusion that the model takes additional risks by trying to predict the market movements, which is expressed in the slight deviation of weights from the baselines' ones. It means that the model tries to gain (and does gain) additional improvement due to some prediction of the market movement, while deeming trading desk to take additional risks, which is not usually an expected behavior.

Therefore, while a trivial baseline shows great results, the model was not expected to outperform the baseline in the risk too, and such an effect is quite logical, as the actual improvement should be via riding the uncertainty of non-linear derivatives delta-hedging.

Section 7.9. Testing the strategies for options.

The results, presented in this Section are at the core of the overall study, as outperforming the outlined baselines in the area of option hedging would prove the value of the proposed framework beyond usability features.

This Section works with the final architecture and provides the experiments with the business logic of the hedging. The architecture ablation study are outlined in Chapter VIII.

The final architecture follows the guidelines of Chapter VI. It is represented by the **LSTM network with 2-layer MLP above it (with additional side LSTM for texts in Experiment 4). The LSTM and MLP use the same number of hidden layer neurons at the level of 32. The features are normalized during forward via LayerNorm (before passing into any other layers). The model is trained using Adam Optimizer with batch size of 32**.

Let's check the results of the optimization of the proposed model. We start off by examining relatively low-frequency hedging at 30 minutes.

<u>Experiment 2</u>. **Option Hedging** (30 min frequency, 2017-01-03 to 2024-04-30).

Data per instrument of notional 1 USD	Average weight	Weights Bounds	Mean PnL, RUB	Standard deviation of PnL, RUB	VaR at 5%, RUB	T-Stat
MLP Hedger	0.89231	[0.221; 0.568]	0.296510	0.34561	0.14721	-5.12
LSTM Hedger	0.28712	[0.246; 0.295]	**0.240436**	0.20865	**0.02009**	<u>**10.10**</u>

Data per instrument of notional 1 USD	Average weight	Weights Bounds	Mean PnL, RUB	Standard deviation of PnL, RUB	VaR at 5%, RUB	T-Stat
SAC MLP	2.15604	[-3.796; 3.591]	-135.39175	135.83582	401.6266	N/A
PPO MLP	0.56712	[0.5214; 1.0023]	0.321161	0.25672	0.18491	N/A
PPO Recurrent	0.43128	[0.128; 0.236]	0.261141	0.25112	0.23981	N/A
Baseline BSM	0.52347	[0.456; 1.000]	0.278402	**0.10796**	0.13268	-
Baseline Heston	0.36781	[0.331; 1.000]	0.257891	0.10954	0.12678	-

Table 4. Source: Calculations of the author.

. As the table highlights, the proposed model indeed shows the extremely positive results, **making the difference with baseline significant even at 0.01%** (t-statistic of 10.10). Moreover, the outlined model provides the hypothetical trading desk with **lower VaR** than both Black-Scholes-Merton and Heston hedging. But the model still exhibits higher standard deviation of the difference, supposing that the risk of such strategy is a bit higher, which, however, is compensated by the better hedging quality, as demonstrated by the t-statistic

.

Looking at the obtained results, a question might arise — **why is the hedging PnL is not close to zero, but constitutes RUB 0.25 per USD 1 notional (around 0.3%)**? This is definitely due to two variables — an imperfection of hedging rules (deviation of the hedging strategy from the theoretical behavior) and the transaction costs. As we can deduce from the table above, our model's confidence interval at 5% is given by $[-0.1686; 0.6494]$ RUB. Thus, as confidence interval's negative part is substantially large, we can conclude **that the hedging imperfection indeed plays significant role, as the model's inability to catch the market movements perfectly is expressed by two-sided error**.

Moreover, as the PnL is defined as (44), the positive skewness of the obtained results denote that the hedging strategy earns less than the target one, on average. And **this is definitely an expected results, as it is simply explained by the transaction costs effect, which take out the money from our hedging strategy's outcome, while being not present in the target PnL by definition**. Thus, the result is indeed expected by from the financial logic's standpoint.

Below the outlined the path of the training for our model. While the left graph demonstrates the optimized loss as per (5), the right depicts the analyzed difference, as per (44). We can note that the convergence of the model is pretty stable both at training, and at validation:

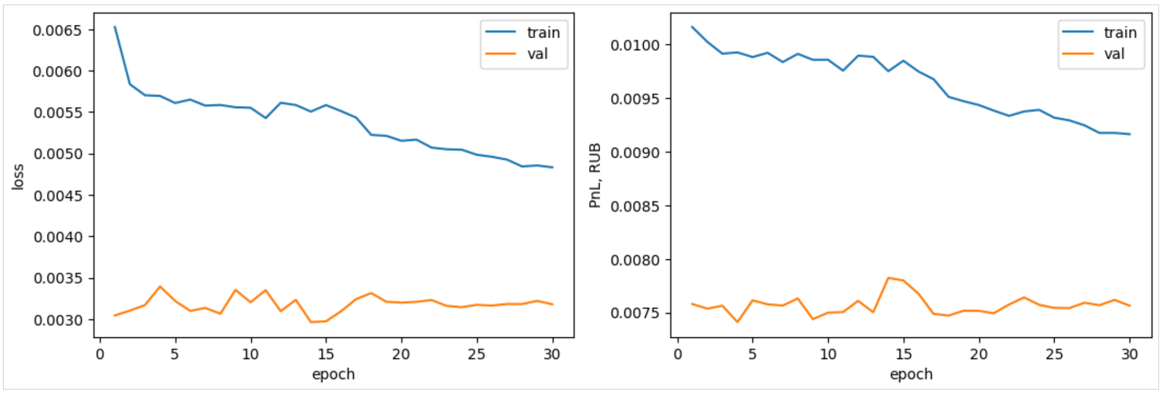

Figure 17: Losses by epoch of 30 min rebalance option hedging. Source: Calculations by the author.

.

Further, let's check the experiment, when we increase the hedging frequency (and, thus, inference frequency) to 5 min datapoints. In this experiment **the results of both the proposed model and the baseline are expected to improve drastically, as more frequent dynamic rebalancing is expected to provide us with better hedge of non-linear uncertainty**, as the lower the frequency, the lower the price change. And by (35) it means that the total risk of deviation undertaken should be less, thus, bringing the improvement in the hedging loss optimization. The result is outlined below.

<u>Experiment 3</u>. **Option Hedging** (5 min frequency, 2017-01-03 to 2024-04-30).

Data per instrument of notional 1 USD	Average weight	Weights Bounds	Mean PnL, RUB	Standard deviation of PnL, RUB	VaR at 5%, RUB	T-Stat
MLP Hedger	0.78129	[0.621; 0.889]	0.291129	0.18213	0.00772	0.07
LSTM Hedger	0.46917	[0.036; 0.473]	**0.251576**	0.17728	**0.03067**	**<u>33.83</u>**
Baseline BSM	0.52374	[0.371; 1.000]	0.296768	**0.10216**	0.16251	-
Baseline Heston	0.54356	[0.411; 1.000]	0.275183	0.11567	0.14129	-

Table 5. Source: Calculations of the author.

.

Indeed, the results above support the hypothesis of **better hedging for both the baseline and the model presented (standard deviation is lower)**. However, it **<u>comes at a price — more frequent hedging means more deals, and, consequently, more transaction costs paid</u>**, deeming the larger deviation of the target PnL (which is unchanged from the hedging frequency change) from the hedging strategy PnL.

Moreover, the weights, suggested by the model, have come closer to the ones that are usually used in the delta-hedging. An interesting note here is that **the model offers hedging by lower amount than the classical financial mathematics approach, deeming buying extra asset into the portfolio excessive from the transaction costs paid standpoint, thus, saving extra costs due to leaving some part of the derivative's exposure unhedged**.

However, an **extremely important result here that the model's outperformance in regards to the baselines have increased dramatically**, which is demonstrated by the significantly higher t-statistic. Therefore, not only we have fed more points in the model to provide it with larger samples for learning, but **the model could override the transaction costs dilemma**, gaining way better opportunity to hedge the derivative with significantly better quality. Thus, the model have attained **better hedging results, deeming the experiment extremely successful**.

We can observe the optimization path of 5 min training below. It can be noted that the convergence stability worsened, compared to 30 minutes case, while resulting in better out-of-sample results, commented above. The chart of the convergence is given below:

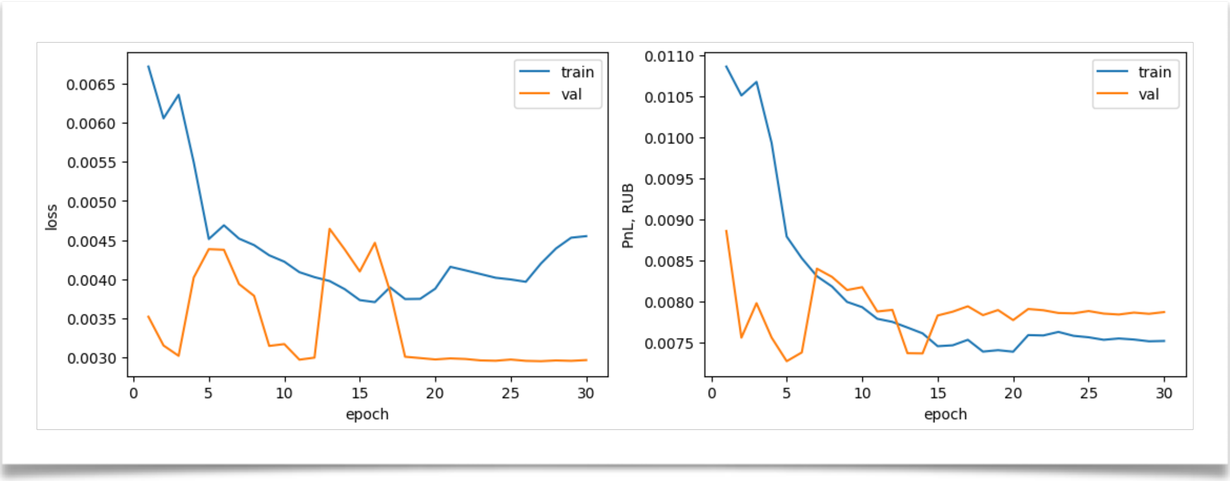

Figure 18: Losses by epoch of 5 min rebalance option hedging. Source: Calculations by the author.

.

The next logical step for improvement in this area is to take the highest frequency available — 1 minute data. However, due to the GPU resources restrictions, the optimization did not gain enough epochs for analysis. But the graph below demonstrates the losses paths, showing that **at 1 minute frequency the model starts to overfit** for training data, causing the validation loss to rise significantly. Therefore, **5 minute data becomes optimal, according to our experiments' results**.

Finally, the results for the optimization with the text features are presented. In order to save the computational time, the *embed_dim* is shortened from 300 to 128, matching the *max_len* of texts, providing the 128 x 128 embeddings for the texts of the news and analytics. The optimization results are presented below.

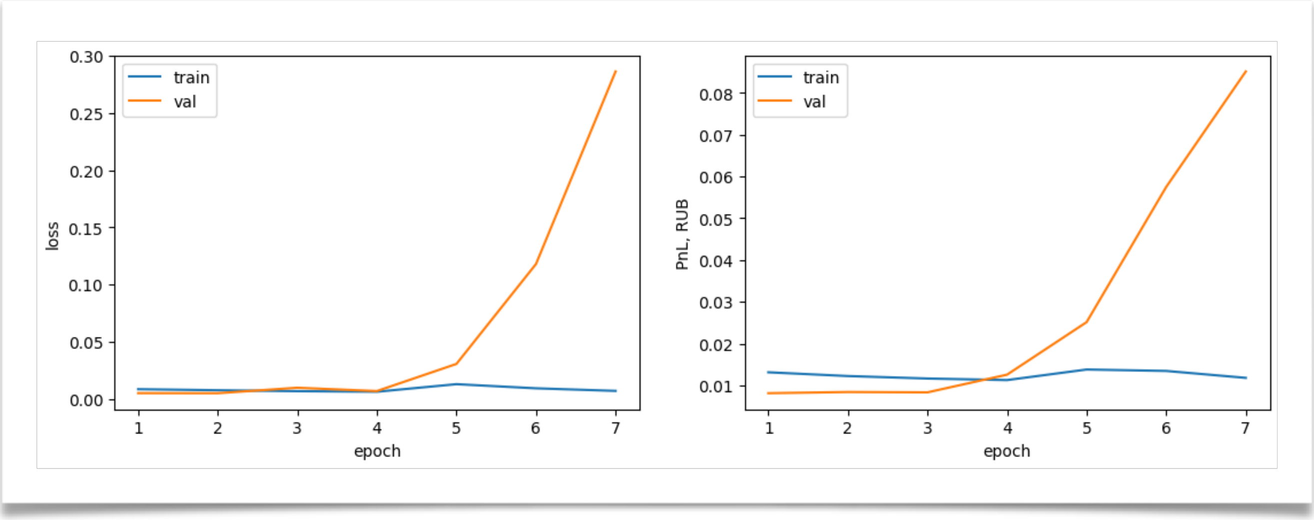

Figure 19: Losses by epoch of 1 min rebalance option hedging. Source: Calculations by the author.

Experiment 4. **Option Hedging with texts** (30 min frequency, 2020-01-01 to 2022-11-01).

Data per instrument of notional 1 USD	Average weight	Weights Bounds	Mean PnL, RUB	Standard deviation of PnL, RUB	VaR at 5%, RUB	T-Stat
LSTM Hedger	0.31085	[0.071; 1.950]	**0.129199**	**0.25407**	0.25626	**133.68**
Baseline BSM	0.51167	[0.422; 1.000]	0.450233	0.34132	0.14198	-
Baseline Heston	0.37891	[0.334; 1.000]	0.341120	0.23126	**0.11789**	-

Table 6. Source: Calculations of the author.

.

The table allows to conclude that **adding texts as the feature indeed allowed to drastically improve the hedging versus the baseline without such feature, attaining extremely significant result with the t-statistic of unprecedented 133.68**., thus not only proving the reliability of the proposed framework's opportunity to work with features, but also showing practically sustainable results of the hedging with such a feature.

The result turns out to be extremely more important from the practical point of view due to the fact that the analyzed timeframe includes also the "tail" events of February 2022, resulting in the significant market movements. Regardless of that, **the model reaches great result, with, however, worse VaR, which is probably due to this event-related movement: as model uses texts to predict the future returns — i.e. to decide, whether to hedge additionally, paying extra transaction costs, or not — the unexpected events cause model to overfit at the stable times, taking additional risk, which is, however, compensated well by the saved transaction costs**.

The optimization process of the model with texts is outlined below. We can note slightly less stable convergence, compared to 30 min rebalancing model, which is due to additional complex feature added:

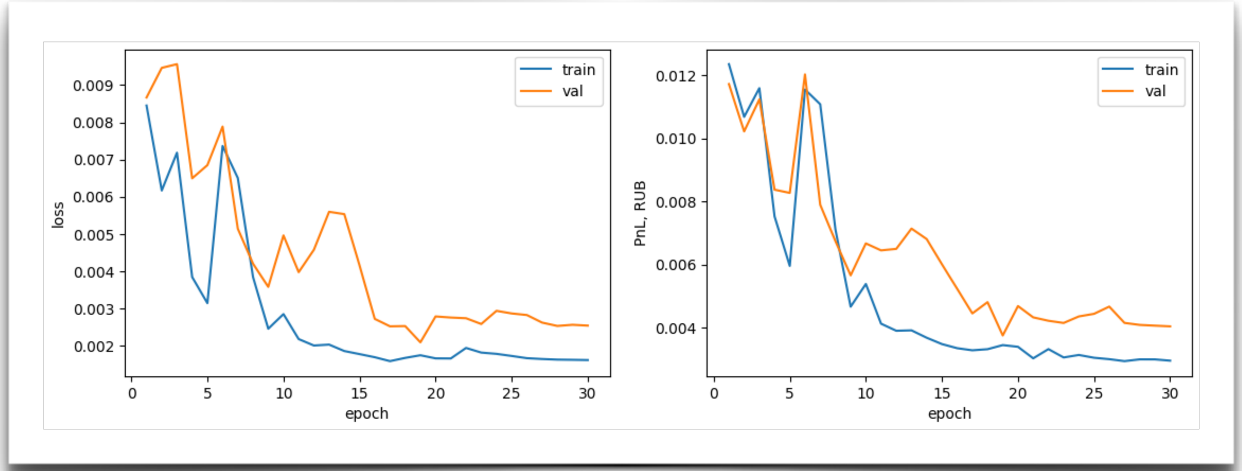

Figure 20: Losses by epoch of option hedging with text features. Source: Calculations by the author.

.

Furthermore, the distribution of PnL difference in RUB is presented below. We can definitely see the **divergence of two distributions, having the target model with texts provide us with extremely lower deviation from target**, as discussed above.

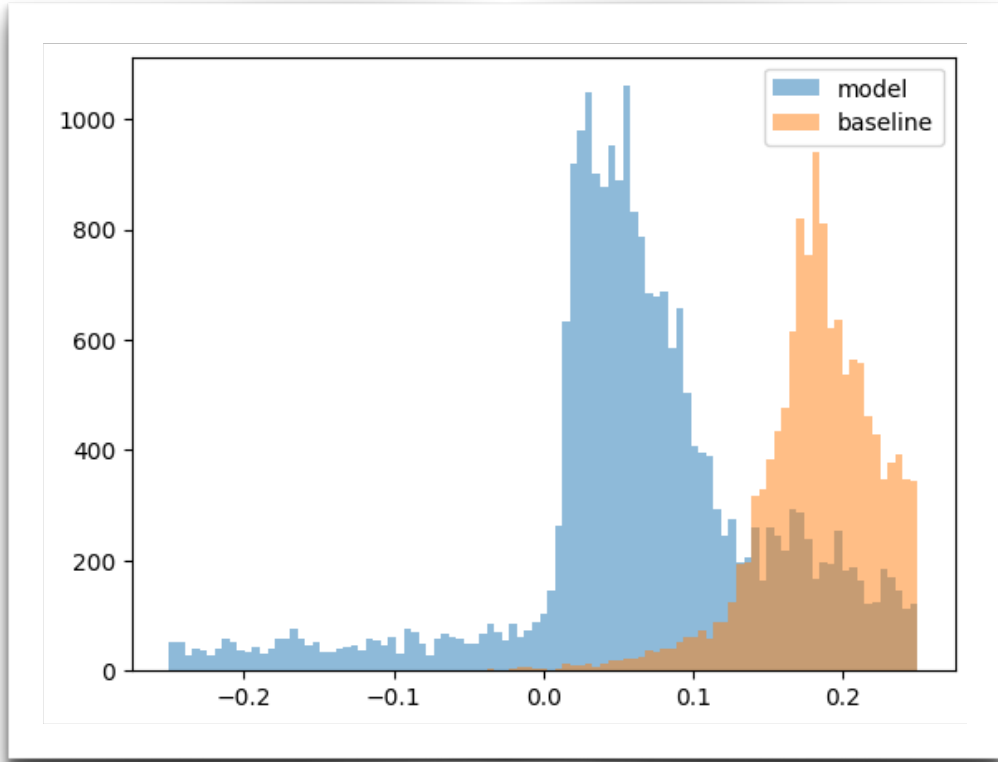

Figure 21: PnL distribution for option hedging with text features. Source: Calculations by the author.

.

Summarizing the findings above, **the proposed framework outperforms all of the stated baselines at 1% significance level, as the model attains substantially lower deviation from**

the derivative's payoff. The best result is attained at 5 minute rebalancing frequency, while adding texts as features allowed for unprecedented level of hedging quality.

The reason for such results of lower PnL divergence with, usually, higher standard deviation is simple — the model have learned to deal with the incomplete markets, balancing the transaction costs versus the risk that arises from leaving some position part unhedged. Therefore, **model actually trades certain amount of transaction costs that have zero volatility versus leaving some of the position unhedged**. Thus, it becomes a source of additional uncertainty for the PnL difference, increasing the volatility. However, **as the model supports such a balance well, according to the presented above results, the average PnL during testing becomes significantly (by the t-Student test) closer to the derivative's payoff**.

Chapter VIII. Ablation Study.

This Chapter provides the details on the logic behind the architecture chosen. As the results in Section 7.9 show, the LSTM *Hedger* outperforms significantly the MLP-based analog. Therefore, the ablation study considers LSTM as a target architecture, and studies the changes in the implementation of this deep learning model.

Moreover, as the NLP studies is out of the scope of this paper, the experiments are done without any exposure of the additional LSTM for texts layer, concentrating on the optimality of the price data-related structure.

First of all, the study checks the optimality of choosing LayerNorm. Surprisingly, while BatchNorm seems a more logical approach from the financial standpoint, as the prices should be normalized across the sample's dimension, LayerNorm shows better results across several runs, thus, it was chosen for the target implementation.

Experiment 5. **Ablation Study: LayerNorm vs BatchNorm** (30 min frequency).

Data per instrument of notional 1 USD	Average weight	Weights Bounds	Mean PnL, RUB	Standard deviation of PnL, RUB	VaR at 5%, RUB	T-Stat
LayerNorm	0.28712	[0.246; 0.295]	**0.240436**	0.20865	**0.02009**	**10.10**
BatchNorm	0.33483	[0.289; 0.343]	0.243174	0.19577	0.02263	8.64

Table 7. Source: Calculations of the author.

.

Further, the need for the additional MLP over the LSTM's output is checked. As expected, MLP improves the results of the model via providing additional non-linearity for modeling the weights change correctly.

Experiment 6. **Ablation Study: MLP over the LSTM's hidden layer** (30 min frequency).

Data per instrument of notional 1 USD	Average weight	Weights Bounds	Mean PnL, RUB	Standard deviation of PnL, RUB	VaR at 5%, RUB	T-Stat
MLP added	0.28712	[0.246; 0.295]	**0.240436**	**0.20865**	0.02009	**10.10**
No MLP	0.82380	[0.381; 0.878]	0.271759	0.27982	**0.01063**	1.38

Table 8. Source: Calculations of the author.

.

Moreover, the optimizer's quality is checked. As the distribution is likely not multimodal (only one optimum exists, as there always is the best available hedging strategy that outperforms significantly the others), Adam, most probably, does not suffer from the steep drops in loss landscape, thus, producing better results.

Experiment 7. **Ablation Study: SGD Optimizer vs Adam Optimizer** (30 min frequency).

Data per instrument of notional 1 USD	Average weight	Weights Bounds	Mean PnL, RUB	Standard deviation of PnL, RUB	VaR at 5%, RUB	T-Stat
Adam	0.28712	[0.246; 0.295]	**0.240436**	0.20865	**0.02009**	**10.10**
SGD	0.46587	[0.279; 0.503]	0.250424	0.17850	0.03530	8.38

Table 9. Source: Calculations of the author.

.

The tests for batch size optimality return the balanced variant of 32 datapoints in batch.

Experiment 8. **Ablation Study: Batch Size 1, 32, 64** (30 min frequency).

Data per instrument of notional 1 USD	Average weight	Weights Bounds	Mean PnL, RUB	Standard deviation of PnL, RUB	VaR at 5%, RUB	T-Stat
batch_size=1	0.23126	[0.017; 0.251]	0.267891	0.24127	0.02567	9.11
batch_size=32	0.28712	[0.246; 0.295]	0.240436	**0.20865**	0.02009	**10.10**
batch_size=64	0.13772	[0.005; 0.176]	**0.232195**	0.26603	**0.01164**	10.06

Table 10. Source: Calculations of the author.

.

The experiments with the size of hidden dimension, i.e., number of neurons in the hidden layer, show slight outperformance of 32 versus 64, thus, it was chosen for the final architecture, however, the result does not seem to have significant effect across several runs.

Experiment 9. **Ablation Study: Hidden Dimension 8, 16, 32, 64** (30 min frequency).

Data per instrument of notional 1 USD	Average weight	Weights Bounds	Mean PnL, RUB	Standard deviation of PnL, RUB	VaR at 5%, RUB	T-Stat
hidden_dim=8	0.59272	[0.483; 0.678]	0.258667	0.19523	0.03245	4.48
hidden_dim=16	0.34811	[0.327; 0.355]	0.245713	**0.19233**	0.02522	9.27
hidden_dim=32	0.28712	[0.246; 0.295]	0.240436	0.20865	0.02009	**10.10**
hidden_dim=64	0.06592	[0.053; 0.072]	**0.227525**	0.29798	**0.00515**	10.04

Table 11. Source: Calculations of the author.

.

Finally, the correctness of including the time till maturity into the model was checked. The experiments show that **such a feature, being crucial from the point of view of financial mathematics, indeed proves its value, being able to transform the result from insignificant (without the feature) to extremely significant (with the feature)**.

Experiment 10. **Ablation Study: Dropping time till maturity as a feature** (30 min frequency).

Data per instrument of notional 1 USD	Average weight	Weights Bounds	Mean PnL, RUB	Standard deviation of PnL, RUB	VaR at 5%, RUB	T-Stat
With time	0.28712	[0.246; 0.295]	**0.240436**	**0.20865**	**0.02009**	**10.10**
Without time	1.07331	[0.729; 1.079	0.283833	0.39499	0.04796	0.83

Table 12. Source: Calculations of the author.

.

Conclusion.

The hedging of the derivatives lies at the core of the modern financial system's functioning mechanics. Moreover, while the development of more and more complex financial products brings the unprecedented popularity of such instruments to our world, it causes requirements for rapid adjustments of the trading algorithms in order to much the speed and complexity of the changes in the financial markets. That is why this study focuses on the improvements in the area of statistical learning for hedging, implementing the easy-to-use hedging framework, which showed great results in terms of applicability for the real-world trading tasks.

Firstly, the thesis introduced a universal hedging framework that requires only definition of the derivative via a payoff formula that transforms the path of the underlying asset's price into the scalar PnL that a holder of said derivative will receive. After that, the framework trains the deep learning model that returns on inference the weights of the assets in the replicating portfolio, being able to work both with arbitrary universe of assets, and the base asset only. Therefore, the inference of introduced framework can be easily integrated with the order management system, creating a full end-to-end hedging algorithm.

Secondly, the proposed framework outperforms significantly the baselines from both classical financial models and advanced reinforcement learning algorithms, **allowing for the stable convergence by the simplicity of sequential-based task, if we account for the transaction costs and the real world borrowing and lending of funds**, recreating the actual trading patterns that an investment bank would use, if such a model is implemented.

The paper results show that the introduced model provides lower average deviation of the PnL from hedging strategy from the PnL of the derivative in the books of the trading desk, compared to the baselines introduced. The **real market data testing on USDRUB and EURRUB show 1% significance of the baseline outperformance for 30 minutes frequency of rebalancing the hedging portfolio, and 0.01% significance for 5 minutes frequency of rebalancing**.

Thirdly, the paper expands the framework to be modified in order to include arbitrary features in the neural hedger. Not only the implementation allows for the seamless inclusion of the virtually any feature, but also analyzes impact of adding the pre-trained embeddings of 2020-2022 texts from Telegram channels to the model's hedging weights prediction layers. **Such an approach results in t-statistic of 133.68, proving the significance at the level of $\alpha << 0.01\,\%$, creating a great starting point for the data science approaches in real world applications of statistical hedging**.

Finally, the research contributes to the relatively fresh field of neural hedging approaches, setting off a direction of classical deep learning gradient-based optimization, contrary to the already studied reinforcement learning implementations. However, the thesis has touched only the tip of the iceberg of statistical learning in the derivatives hedging, thus, leaving much to be studied in terms of both the presented approach improvement, and feature generation to empower the introduced model's capabilities.

Moreover, the paper might be of a great value for the practicians on financial markets, allowing the seamless integration in the existing hedging modules, which can drive the improvement in the area of client service on financial markets and trading divisions income from such an area.

Furthermore, it should be mentioned that there are numerous areas of potential research improvements on the base of the work, presented above. Such research, being an interconnection of the deep learning studies and financial mathematics, allows for contribution to both these areas. Concerning the latter, one can find virtually limitless improvements that can be done, which is provided by the broadness of the introduced framework, proving its value one more time.

First of all, there are several areas of financial while the existing study was limited to the most popular hedging cases, an interesting topic to cover would be running the framework on the dataset, where we do not have an opportunity to hedge via the base asset, deeming the hedging process to be done with other assets only, employing the correlation and non-linear dependance to implement tail-controlled hedge. Next to that, one can study the effect on hedging long-term options, contrary to the 5-day ones, presented here — such instruments with higher exposure to "vega" greek, thus, to the market expectations, rather than realizations, might require additional feature selection and architecture modification. Lastly, a potential vector of development on top of this study is application to hedging the exotic derivatives, adding the vanilla derivatives, like presented in this paper, to the tradable universe.

Besides that, there are viable research areas in the sphere of deep learning, which are directly related to this study. The researches may vary from application of more advanced frameworks, than LSTM, like Transformer, vector of development for which was outlined in Chapter VI, to Natural Language Processing modifications, improving the results of the text embeddings layer, which showed great results already, but still leaves much to be adjusted.

Ultimately, the study obtained strong results in terms of statistical hedging analysis, resulting in innovative approach to such a sphere with deep learning model that accounts for market frictions, being a perfect end-to-end solution in the incomplete markets. The paper sheds light on the area of gradient-based optimization in complex financial products, which is expected to be one of the leading research streams both in academia and investment banking industry, shaping the future economics developments all across the world.

List Of References:

1. Ilyinskiy K.N. and Buev M.V. (2020). V Zerkale Supermodeley. Nauka (2020).
2. Ahn Chang Mo (1992). Option Pricing When Jump Risk Is Systematic. *Mathematical Finance*, vol. 2, no. 4, pp. 299-308 (Oct., 1992).
3. Bates D.S. (1991). The Crash of '87: Was It Expected ? The Evidence From Option Markets. *Journal of Finance*, vol. 46, no. 3, pp. 1009-1044 (Jul., 1991).
4. Bergomi L. (2017). Local-Stochastic Volatility: Models and Non-Models (January 1, 2017).
5. Bessis J. (2010). Risk Management in Banking. Wiley (2010).
6. Black F. and Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy*, vol. 81, no. 3, pp. 637-654 (May/Jun., 1973).
7. Bridle J. (1989). Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters. *NIPS* (1989).
8. Burnett B. (2020). Hedging Valuation Adjustment: Fact and friction. *Risk Magazine* (Feb., 2021).
9. Bühler H., Gonon L., Teichmann J., Wood B. (2018). Deep Hedging. (February 8, 2018).
10. Bühler H., Murray P., Pakkanen M.S., Wood B. (2021). Deep Hedging: Learning to Remove the Drift under Trading Frictions with Minimal Equivalent Near-Martingale Measures (November 15, 2021).
11. Bühler H. (2019). Statistical Hedging (March 1, 2019).
12. Castagna A. (2010). FX Options and Smile Risk. John Wiley & Sons, Inc. (2010).
13. Corrado C.J. and Su T., (1996). Skewness and Kurtosis in S&P 500 Index Returns Implied by Option Prices. *Journal of Financial Research*, vol. 19, no. 2, pp. 175-192 (Summer, 1996).
14. Cox J.C., Ross S.A. and Rubinstein M., (1979). Option Pricing: A Simplified Approach. *Journal of Financial Economics*, vol. 7, pp. 229-263 (Jul., 1979).
15. Derman E. and Kani I. (1994). Riding on Smile. *Risk*, vol. 7, pp. 32-39 (Jan., 1994).
16. Dempster M.A.H. and Thorlacius A.E. (1998). Stochastic Simulation of International Economic Variables and Asset Returns: The Falcon Asset Model.
17. Du J., Jin M., Kolm P.N., Ritter G., Wang Y., Zhang B. (2020). Deep Reinforcement Learning for Option Replication and Hedging. *The Journal of Financial Data Science* 2(4):44-57 (2020).
18. Dupire B. (1994). Pricing with a Smile. *Risk*, vol. 7, pp. 18-20 (Jan., 1994).
19. Enders W. (2004). Applied Econometric Time Series. John Wiley & Sons, Inc. (2004).
20. Engle R.F. and Ng V.K. (1993). Measuring and Testing the Impact of News on Volatility. *The Journal of Finance*, vol. 48, no. 5, pp. 1749–1778 (Dec., 1993).
21. Galluccio S. and Le Cam Y. (2005). Implied Calibration of Stochastic Volatility Jump Diffusion Models.
22. Haarnoja T., Zhou A., Abbeel P., Levine S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *ICML* (2018).

23. Haykin, S. (1994). Neural networks: a comprehensive foundation. *Prentice Hall PTR* (1994).
24. Heston, S.L. (1993). A Closed Solution for Options with Stochastic Volatility, with Application to Bond and Currency Options. *The Review of Financial Studies*, 6, 327-343 (1993).
25. Hull J.C. (2014). Options, Futures, and Other Derivatives. Pearson. (2014).
26. Luenberger, D. G. (2014). Investment Science. *Oxford University Press USA* (2014).
27. Merton C.R. (1971). Optimal Consumption and Portfolio Rules in a Continuous Time Model. *Journal of Economic Theory*, vol. 3, no. 4, pp. 373–413 (Dec., 1971).
28. Merton C.R. (1973). The Theory of Rational Option Pricing. *The Bell Journal of Economics and Management Science*, vol. 4, no. 1, pp. 141-183 (Mar., 1973).
29. Merton C.R. (1974). On the Pricing of Corporate Debt: The Risk Structure of Interest Rates. *The Journal of Finance*, vol. 29, no. 2, pp. 449–470 (May, 1974).
30. Nelson, C. R., & Siegel, A. F. (1987). Parsimonious Modeling of Yield Curves. *The Journal of Business*, 60(4), 473–489.
31. Officer R.R. (1972). The Distribution of Stock Returns. *Journal of the American Statistical Association*, vol. 67, no. 340, pp. 807-812 (Dec., 1987).
32. Salih N. Neftci (2008). Principles of Financial Engineering. Elsevier (2008).
33. Samuelson P. (1965). Proof That Properly Anticipated Prices Fluctuate Randomly. *Industrial Management Review*, vol. 6 (Spring, 1965).
34. Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O. (2017). Proximal Policy Optimization Algorithms (July 20, 2017).
35. Scott L.O. (1987). Option Pricing When the Variance Changes Randomly: Theory, Estimation and an Application. *Journal of Financial and Quantitative Analysis*, vol. 22, pp. 419-438 (Dec., 1987).
36. Seelama P. and Thongtha D. (2021). Option Pricing Model with Transaction Costs and Jumps in Illiquid Markets. *Journal of Mathematical Finance*, vol. 11, no. 3, pp. 361-372 (Jan., 2021).
37. Sheikh A.Z. and Qiao H. (2009). Non-Normality of Market Returns: A Framework for Asset Allocation Decision Making. *The Journal of Alternative Investments*, vol. 12, no. 3, pp. 8-35 (Dec., 2009).
38. Son, G. & Kim J. (2021). Neural Networks for Delta Hedging (December 19, 2021).
39. Stokie M.D. (1982). The Distribution of Stock Market Returns: Tests of Normality. *Australian Journal of Management*, vol. 7, no. 2, pp. 159-178 (1982).
40. Svensson Lars E.O. (1994). Estimating and Interpreting Forward Interest Rates: Sweden 1992-1994 (September 1994). *IMF Working Paper* No. 94/114 (1994).
41. Trautmann S. and Beinert M. (1995). Stock Price Jumps and Their Impact on Option Valuation.

42. Volk-Makarewicz W., Borovkova S. and Heidergott B. (2022). Assessing the Impact of Jumps in an Option Pricing Model: A Gradient Estimation Approach. *Journal of Operational Research*, vol. 298, no. 2, pp. 740-751 (Apr., 2022).
43. West K.D., Edison H.J. and Cho D. (1993). A Utility-Based Comparison of Some Models of Exchange Rate Volatility. *Journal of International Economics*, vol. 35, no. 1-2, pp. 23-45 (Aug., 1993).
44. West K.D., Edison H.J. and Cho D. (1993). A Utility-Based Comparison of Some Models of Exchange Rate Volatility. *Journal of International Economics*, vol. 35, no. 1-2, pp. 23-45 (Aug., 1993).
45. Williams R. J. (1992). Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8, pp. 229-256 (1992).
46. cbr.ru.
47. deephedging.com.
48. ecb.europa.eu.
49. fasttext.cc.
50. federalreserve.gov.
51. home.treasury.gov.
52. investopedia.com.
53. moex.com.
54. nltk.org.
55. paperswithcode.com.
56. projectpro.io.
57. pymorphy2.radthedocs.io.
58. risk.net
59. stable-baselines3.readthedocs.io.
60. stanford.edu.

Appendix.

Appendix A. Code source and experiment data.

The Python implementation of the presented framework, runs of the experiments and the data, needed to recreate the results of the study can be found at https://github.com/v-buchkov/deep-hedging/tree/main. The README.md contains the information on how to run the code, which allows for everyone to use the implementation in order to achieve the claimed results.

Appendix B. An example of USDRUB orderbook.

As outlined in the Section 7.1, the study works with the Volume-Weighted Average Price, which is calculated for the snapshots of the orderbooks for each timestamp. The example of such orderbook is presented below.

		89.1900	300	2 865
		89.1925	150	2 565
		89.1950	50	2 415
		89.2075	50	2 365
		89.2100	75	2 315
		89.2125	150	2 240
		89.2150	150	2 090
		89.2175	50	1 940
		89.2200	100	1 890
		89.2250	250	1 790
		89.2275	50	1 540
		89.2300	150	1 490
		89.2325	100	1 340
		89.2350	150	1 240
		89.2375	100	1 090
		89.2400	500	990
		89.2425	100	490
		89.2525	150	390
		89.2550	190	240
		89.2575	50	50
150	150	89.2650		
200	50	89.2675		
270	70	89.2700		
469	199	89.2725		
689	220	89.2750		
889	200	89.2775		
1 989	1 100	89.2800		
3 125	1 136	89.2900		
3 225	100	89.2950		
3 275	50	89.2975		
4 179	904	89.3000		
4 279	100	89.3025		
4 329	50	89.3050		
4 479	150	89.3075		
4 539	60	89.3100		
4 589	50	89.3150		
4 639	50	89.3325		
4 789	150	89.3425		
4 989	200	89.3450		
4 992	3	89.3500		

.

Appendix C. Examples of Telegram channels posts.

As outlined in the methodology of Section 7.3, the Experiment 4 uses texts of Telegram channels to improve on the model's performance. Therefore, here the examples of such texts are outlined in order to check the logic behind such a feature.

Timestamp	Source Name	Text
2019-10-17T14:37:31	The Movchans	#Macro : Brexit со сделкой? Одна из крупнейших проблем для мировой экономики (ввиду остроты момента и приближения финальной даты), которую упоминают в последнее время все: от локальных ЦБ до ключевых центральных банков и МВФ – "Brexit без сделки"– возможно, разрешается благополучно. В четверг премьер-министр Великобритании Борис Джонсон и председатель Европейской Комиссии Жан-Клод Юнкер заявили о достижении соглашения, камнем преткновения которого был вопрос таможенной границы между Ирландией и Северной Ирландией. Скорее всего, Европарламент ратифицирует соглашение. Борис Джонсон надеется, что парламент Великобритании одобрит соглашение в субботу. В противном случае, Джонсон будет вынужден просить у ЕС новой отсрочки еще на три месяца , чтобы урегулировать спорные вопросы со всеми заинтересованными сторонами.Сообщение о достигнутом, но пока не ратифицированном соглашении стало причиной роста курса GBP, роста европейских акций и падения цен UST. Доходность 10Y UST поднялась на 3 бп до 1.78%. Инверсия кривой в длинном участке все больше исчезает – спрэд между 10Y ставками и 3-мес, став уже положительным, достиг 15 б.п. Ратификация парламентом Великобритании сделки может стать серьезным прорывом с 2016 г. и снизить неопределенность в отношении дальнейшего развития мировой экономики. Тем не менее, я не исключаю, что даже в этом случае Банк Англии, ближайшее совещание которого запланировано на 7 ноября, будет вынужден снизить ставку. Казначейские облигации США, скорее всего, смогут не только остаться в околонулевой зоне реальных процентных ставок, но, возможно, даже смогут стабилизироваться в положительной области.Итак, запасаемся попкорном и ждем итогов обсуждения текста соглашения в парламенте Великобритании в эту субботу и помним, что в случае провала Борис Джонсон обещал пойти на крайние меры, лишь бы разойтись с ЕС. Также ждем последние новости с торгового фронта, где Китай не решается пока пойти на Фазу #1 в соглашении с США.Александр Овчинников, управляющий ARGO SP.

Timestamp	Source Name	Text
2017-11-17T14:13:36	MMI	Фееричные у нас, конечно, новости в части экономической политики! То, что налоговая нагрузка на бизнес не будет повышаться, звучит как мантра со стороны правительства регулярно. Но чудес не бывает. Прав Кудрин – если Вы хотите жить при жёстком бюджетном правиле на $40, то неизбежно будете повышать налоговую нагрузку. Либо смягчайте это правило, хотя бы до $45, либо повышайте налоги. Выбирают второе. Ну и о каком экономическом росте мы говорим? Маразм ситуации ещё в том, что повышаемые налоги прямым образом бьют по инвестиционной активности. Налог на движимое имущество (автомобили, станки, оборудование) был отменён в 2012 году именно с целью подтолкнуть производителей к модернизации. Возвращение налога с 2018 года, по-видимому, можно рассматривать как завершение модернизации нашей экономики. Напомню, что Конгресс США принял вчера закон о налоговой реформе, снижающий налоговую нагрузку на экономику: https://t.me/russianmacro/399 Почитать подробнее про повышение налогов в РФ можно, например, здесь: https://www.bfm.ru/news/370197

Timestamp	Source Name	Text
2022-09-26T16:09:52	Alfa Wealth	ОФЗ обвалились. Не нужно паниковать. На прошлой неделе цены на 🇷🇺 рынке облигаций упали и сейчас продолжают снижаться. Новые доходности к погашению по ОФЗ достигают 10,5%. Коррекция началась на более жёсткой риторике ЦБ и усилилась на фоне «геополитической» напряжённости. Причём ситуация на рынке облигаций продолжает усугубляется. Когда это остановится? На этот вопрос ответа нет. Цены уже серьёзно упали, но распродажа продолжается. Понятно, что продавцы руководствуются собственными оценочными критериями и пытаются спасти деньги по своим соображениям. Весь вопрос в том, нужно ли вам брать с них пример?Если мы вспомним март, то тогда падение было ещё более существенным, но рынок облигаций достаточно быстро восстановился. Вообще, чем с точки зрения инвестора является падение на рынке рублёвые облигаций? Вызовом или возможностью? Напомню, серьёзные падения на рынке рублёвых облигаций были в 2008, в 2014 и в марте 2022. Если посмотреть, как себя вели облигационные фонды, то по историческим данным можно заключить, что падение на 10% выкупается в срок от 2 до 4 месяцев (ниже для примера наш фонд с тикером AKMB). Для кого-то это может быть слишком долго, но для большинства, по моему мнению, это совсем незначительный срок. Что могу порекомендовать? Если вам срочно не нужны деньги, не фиксируйтесь на коррекции. Да, невозможно спрогнозировать, когда она завершится, но исторически рублёвые облигации за 2-4 месяца выходят в плюс. Поэтому нет смысла выходить из облигаций, если ваши жизненные обстоятельства не просят вас самих на выход. Повторюсь — да, цены могут ещё снизиться, и невозможно определить, где будет дно, но всплытие от этого дна не займёт много времени. По крайней мере история предыдущих обвалов свидетельствует именно об этом.

.