

DOI: 10.17323/2587-814X.2024.3.56.69

# Оптимизация долгосрочного инвестирования на основе диверсификации Марковица

**А.В. Куликов** 

E-mail: avkulikov15@gmail.com

**Д.С. Полозов** 

E-mail: polozov.ds@phystech.edu

**Н.В. Волков** 

E-mail: nikita.v.volkov@phystech.edu

Московский физико-технический институт, Долгопрудный, Россия

## Аннотация

В работе рассмотрен алгоритм для долгосрочного инвестирования, позволяющий находить оптимальные решения в пространстве меньшей размерности. Снижение размерности достигается путем применения метода главных компонент или ядерного метода главных компонент. Подбор весов для портфеля осуществляется с помощью метода Марковица. В качестве гиперпараметров для модели рассмотрены размер окна, параметр сглаживания, период ребалансировки и доля объясненной дисперсии в методах уменьшения размерности. Представленный алгоритм содержит регуляризацию весов с учетом комиссии за ребалансировку портфеля. Подбор гиперпараметров осуществляется на основе коэффициента Мартина, что позволяет учитывать максимальную просадку для рассматриваемых алгоритмов. Показано, что предложенный алгоритм, оптимизированный на данных с 1990 по 2016 год, способен обеспечить более высокую доходность и значение коэффициента Шарпа, чем бенчмарк S&P 500 в период с 2017 по 2022 год. Это свидетельствует о том, что с помощью корректировки весов в портфеле можно улучшить производительность алгоритма.

**Ключевые слова:** PCA, Kernel PCA, размер окна, алгоритм Марковица, сеточный поиск, метод байесовской оптимизации

**Цитирование:** Куликов А.В., Полозов Д.С., Волков Н.В. Оптимизация долгосрочного инвестирования на основе диверсификации Марковица // Бизнес-информатика. 2024. Т. 18. № 3. С. 56–69.

DOI: 10.17323/2587-814X.2024.3.56.69

## Введение

**В** настоящее время одним из наиболее заметных глобальных трендов является динамичное развитие финансовых рынков. В первые два десятилетия XXI века рыночная капитализация увеличилась более чем в три раза [1]. При этом в США только за период с 2017 по 2021 год количество клиентских инвестиционных счетов в трех крупнейших брокерских фирмах Charles Schwab, Fidelity Investments и Robinhood увеличилось почти вдвое [2]. Стремление частных инвесторов активно искать перспективные инвестиционные возможности стимулировало интерес исследователей к поиску оптимального инвестиционного портфеля с целью максимизации прибыли при учете финансового риска.

Значительный прогресс в области оптимизации портфеля был достигнут с появлением портфельной теории Марковица [3]. Классическая задача оптимизации Марковица заключается в поиске такого распределения долей активов в портфеле, которое минимизирует риск при фиксированном уровне ожидаемой доходности. В более общем виде поиск оптимального распределения активов происходит при заданном коэффициенте баланса между минимизацией риска и максимизацией доходности. Данный коэффициент соответствует степени неприятия риска инвестором.

Алгоритм поиска оптимального распределения долей активов в портфеле для задачи Марковица основан на оценке средней доходности акций и их матрицы ковариаций на исторических данных. Однако данный алгоритм имеет ряд ограничений, в частности, он предполагает, что исторические значения доходности и риски акций сохраняют свое распределение в будущем, что не всегда справедливо [4]. Помимо этого, диверсификация Марковица уязвима к выбросам из-за неустойчивости средних доходностей акций [5]. В матрице ковариаций доходностей, используемой для оценки риска, также могут возникать выбросы.

Одним из способов решения указанных проблем, рассматриваемых в литературе, являются методы уменьшения размерности, такие как анализ главных компонент (PCA) и ядерный анализ главных компонент (KernelPCA). Данные методы помогают усовершенствовать алгоритм Марковица путем устранения выбросов в матрице ковариаций и выделения компонент, на которые стоит ориентироваться при сборе портфеля. PCA помогает выделить наиболее важные направления в пространстве данных (глав-

ные компоненты), которые объясняют наибольшую долю дисперсии. При понижении размерности выбросы имеют меньшее влияние на главные компоненты, так как PCA учитывает общую изменчивость данных. Эти подходы помогают снизить влияние шума в данных и улучшить качество оптимизации портфеля, как, например, в работах [6, 7]. Примером использования Kernel PCA к задаче Марковица является работа [8].

В работе рассмотрена обобщенная задача Марковица, учитывающая комиссию, которую платит инвестор за перебалансировку портфеля, и предложен алгоритм для решения этой оптимизационной задачи. Данный алгоритм предварительно обрабатывает данные о доходности активов, очищая их от шума, затем формирует портфель, используя экспоненциально сглаженные значения доходности акций, и оптимизирует линейную комбинацию риска и доходности с учетом комиссии.

Для того чтобы обойти ограничения классического решения для задачи Марковица, в работе рассмотрены четыре вариации предложенного алгоритма: без перехода в пространство меньшей размерности, переход с использованием PCA, а также переходы с полиномиальным и гауссовским ядрами KernelPCA.

В указанных выше работах исторический временной горизонт для определения ожидаемой доходности и матрицы ковариации определяются эмпирическим образом. В работах [9–17] рассмотрен подход к предсказанию ожидаемых доходностей и матрицы ковариации с помощью методов машинного обучения и статистических моделей типа GARCH. В данной работе для оптимизации работы алгоритма Марковица предлагается оптимизация гиперпараметров модели для определения оптимального портфеля по историческим данным: частота смены портфеля, размера окна, количество отбираемых компаний для дальнейшей работы и параметр уменьшения размерности в KernelPCA и Kernel PCA.

Ввиду большого числа гиперпараметров в алгоритме традиционные методы подбора параметров, — сеточный поиск (GridSearch) и случайный поиск (RandomSearch), — могут не привести к оптимальным результатам. Поэтому в данной работе для сравнения также используется метод байесовской оптимизации, основанный на гауссовских процессах [18]. Метод байесовской оптимизации — это алгоритм для нахождения оптимальных значений параметров функции, когда доступна толь-

ко ограниченная информация о функции и ее поведении. Суть метода заключается в итеративной последовательности выбора следующих пробных точек, основываясь на вероятностной модели, которая аппроксимирует неизвестную функцию. Метод позволяет находить оптимальное решение с использованием относительно небольшого числа итераций, поскольку алгоритм активно адаптируется к информации, полученной из предыдущих итераций. В контексте рассматриваемой задачи оптимизируемой функцией от весов портфеля является индекс Мартина, введенный в работе [19]. Таким образом, для каждой из четырех вариаций алгоритма подбор параметров проводится двумя способами: с помощью случайного поиска по сетке и байесовской оптимизации.

Обучение предложенных алгоритмов было проведено на данных доходности около 300 акций в период с 1990 по 2016 год. Для оценки эффективности каждой из восьми полученных вариаций алгоритма в работе проведено сравнение различных метрик доходности получившихся портфелей и сравнение портфелей с индексом S&P 500. Данный индекс является классическим бенчмарком для разрабатываемых алгоритмов, поскольку считается лучшим индикатором акций компаний с большой капитализацией [20] и отражает общее состояние экономики США [21, 22]. В работе [23] показано, что веса в классическом алгоритме Марковица также связаны с данным бенчмарком.

Результаты исследования показывают, что наиболее эффективным алгоритмом является вариация, использующая гауссовское ядро в KernelPCA и подобранная с помощью байесовской оптимизации. Портфель, оптимизированный с помощью данного метода, имеет наиболее высокую доходность и коэффициент Шарпа, во многом за счет наименьшей просадки в период вспышки пандемии COVID-19, и в то же время демонстрирует наиболее высокую волатильность по сравнению с остальными вариациями. Кроме того, показано, что за 2017–2022 гг. все предложенные алгоритмы показывают доходность выше индекса S&P 500.

Работа имеет следующую структуру: раздел 1 содержит постановку оптимизационной задачи, в разделе 2 представлено описание алгоритма для решения данной задачи и его вариаций, в разделе 3 приведены метрики для сравнения вариаций алгоритма, раздел 4 описывает используемые данные, анализ результатов исследования представлен в разделе 5.

## 1. Постановка оптимизационной задачи

Рассмотрим модель финансового рынка с  $N$  активами (акциями), доходности которых являются случайными величинами  $r_i$ ,  $i = 1, \dots, N$ . Вектор ожидаемых доходностей акций будем обозначать как  $\bar{\mu} = (\mathbf{E}r_1, \dots, \mathbf{E}r_N)^\top$ , а матрицу ковариаций доходностей акций как

$$\Sigma = (\text{cov}(r_i, r_j))_{(i,j=1)}^N.$$

Из данных активов инвестор формирует портфель с долями акций  $w_i \geq 0$ ,  $i = 1, \dots, N$ , где условие  $w_i \geq 0$  соответствует отсутствию возможности коротких продаж. Пусть  $\alpha$  – степень неприятия риска инвестором. Тогда преобразованная задача оптимизации Марковица для данного инвестора имеет вид:

$$\begin{aligned} \min_{\bar{w}} \quad & \frac{1}{2} \bar{w}^\top \Sigma \bar{w} - \alpha \bar{\mu}^\top \bar{w} \\ \text{s.t.} \quad & \bar{\mathbf{I}}^\top \bar{w} = 1, \\ & \bar{w} \geq \bar{0}. \end{aligned} \quad (1)$$

Согласно [24], данная задача эквивалентна классической задаче Марковица, в которой отсутствует коэффициент  $\alpha$ , с точностью до фиксированных параметров: в классической задаче фиксируется параметр ожидаемой доходности, а в преобразованной – коэффициент баланса между минимизацией риска и максимизацией доходности. В рассматриваемом алгоритме  $\alpha = 0,05$  однако в общем случае этот параметр можно изменять, больше опираясь на минимизацию риска при меньших  $\alpha$  и на максимизацию доходности при больших.

В данной работе мы рассмотрим более общую задачу оптимизации Марковица, в которой добавлена комиссия  $\lambda$  за перебалансировку портфеля. Таким образом, если  $P$  – период смены портфеля, а  $\bar{w}_p$  – распределение весов акций в предыдущий период времени, то задача оптимизации (1) принимает вид

$$\begin{aligned} \min_{\bar{w}, P} \quad & \frac{1}{2} \bar{w}^\top \Sigma \bar{w} - \alpha \bar{\mu}^\top \bar{w} - \frac{\lambda}{P} \left\| \bar{w} - \bar{w}_p \right\|^2 \\ \text{s.t.} \quad & \bar{\mathbf{I}}^\top \bar{w} = 1, \\ & \bar{w} \geq \bar{0}. \end{aligned} \quad (2)$$

Вектор распределения весов в момент времени 0 полагаем  $\bar{w}_p = \bar{0}$ .

## 2. Описание алгоритма

### 2.1. Принцип работы алгоритма

На вход оптимизационного алгоритма подаются данные о стоимости акций за определенный период времени. Результатом работы алгоритма являются веса  $w_i$  — доли соответствующих активов в портфеле инвестора, которые перебалансируются с определенной частотой, которая передается в алгоритм в качестве гиперпараметра.

Работа алгоритма, представленного ниже, условно состоит из двух частей: подбора гиперпараметров портфеля и выбора оптимальных весов, соответствующих этим гиперпараметрам. Из гиперпараметров производится подбор параметров предобработки данных (частота смены портфеля, фактор размера окна, количество отбираемых компаний для дальнейшей работы) и параметров уменьшения размерности (доля объясненной дисперсии и гиперпараметры ядер в KernelPCA).

Подбор весов осуществляется с помощью метода *fit* на обучающем датасете акций за определенный период времени и включает следующие этапы.

**Фильтрация данных.** На первом этапе происходит фильтрация данных, после которой для анализа остаются наблюдения только в рамках заданного временного окна.

Размер окна определяется двумя гиперпараметрами: *period\_change\_portfolio* — период смены портфеля, и *size\_of\_window\_rank* — фактор размера окна. Размер окна выражается как произведение указанных параметров. Такая функциональная зависимость позволяет варьировать размер окна вместе с изменением периода смены портфеля.

Обычно фактор размера окна принимает значения от двух до пяти, что соответствует окну примерно 2–5 периодов смены портфеля. Например, для анализа годовой беты актива в работе [25] используется окно размером 5 лет, а в работе [26] для оценки параметров смены портфеля каждые 250 дней используется скользящее окно размером 1000 дней. Применение оконной фильтрации позволяет учитывать ближайшие, наиболее актуальные данные, при этом исключая устаревшие.

**Выбор акций для формирования портфеля.** На данном этапе оценивается прибыльность акций для включения в портфель наиболее доходных из них.

Чтобы придать больший вес наиболее актуальным данным, применяется экспоненциальное сглаживание с параметром  $\alpha \in (0, 1)$ . В алгоритме пара-

метр экспоненциального сглаживания принимает значение  $\alpha = 0,99$ . Если применение фильтрации на предыдущем этапе эквивалентно умножению всех наблюдений в окне на 1, а за его пределами на 0, то экспоненциальное сглаживание можно представить как умножение доходностей на  $\alpha^{\Delta t}$ , где  $\Delta t$  представляет собой количество торговых дней, прошедших с рассматриваемой даты до текущего момента. Такой подход обеспечивает более высокий вес недавним данным, сохраняя влияние предыдущих периодов с уменьшающимся весом по мере удаления в прошлое.

Для отбора акций, включаемых в портфель, проводится вычисление взвешенной средней доходности для каждой акции с использованием экспоненциально сглаженных весов. Расчет осуществляется по формуле:

$$\bar{r}_{it} = \frac{\sum_{l=\tau-T+1}^{\tau} r_{it} \alpha^{\tau-l}}{\sum_{l=\tau-T+1}^{\tau} \alpha^{\tau-l}},$$

где  $T$  — величина окна;

$r_{it}$  — доходность акции  $i$  в момент  $t$ .

После вычисления взвешенных средних доходностей акции сортируются в порядке убывания и выбираются  $n\_top\_companies$  лучших акций компаний для включения в портфель. Количество отбираемых акций является гиперпараметром алгоритма.

**Уменьшение размерности.** Данный этап применяется только для вариаций алгоритма с PCA или KernelPCA. Как уже было отмечено, эти методы позволяют усовершенствовать алгоритм Марковица с помощью устранения выбросов в матрице ковариаций и выделения компонент, на которые впоследствии ориентируется алгоритм при формировании портфеля.

При применении этих методов снижения размерности, первоначально определяется количество компонент, включаемых в модель. Это количество определяется с учетом объясненной дисперсии первыми 1, 2, ...,  $n\_top\_companies$  компонентами на основе гиперпараметра *var\_ratio*, отвечающего за долю объясненной дисперсии. Доля объясненной дисперсии для  $d$  компонент — это величина  $\delta_d$ , равная отношению суммы квадратов отклонений наблюдаемых данных от их проекции на главные компоненты, к выборочной дисперсии данных. Большая остаточная дисперсия указывает на то, что главные компоненты не объясняют значительную часть изменчивости данных, что может свидетельствовать о наличии дополнительных

факторов, неучтенных в модели. Количество компонент  $d$  определяется как число, при котором  $\delta_d \geq \text{var\_ratio}$ , но  $\delta_{d-1} < \text{var\_ratio}$ .

Затем методу уменьшения размерности передается найденное число компонент и гиперпараметры, после чего он обучается на тренировочных данных, что позволяет снизить размерность данных, сохраняя ключевые характеристики и очищая их от выбросов.

Среди гиперпараметров KernelPCA отметим:

- ♦ *kernel* – тип ядра: в алгоритме рассматривались только гауссовское ядро «*rbf*»:

$$k(\bar{x}, \bar{y}) = \frac{e^{-\|\bar{x}-\bar{y}\|^2}}{\gamma},$$

и полиномиальное ядро «*poly*»:

$$k(\bar{x}, \bar{y}) = \frac{(\bar{x}^\top \bar{y} + c)^q}{\gamma};$$

- ♦ *kerneldegree* – степень полиномиального ядра ( $q$  в формуле полиномиального ядра);
- ♦ *kernelgamma* – параметр масштаба ( $\gamma$  в формулах выше);
- ♦ *kernelcoef0* – константа полиномиального ядра ( $c$  в формуле полиномиального ядра).

**Подбор долей акций в портфеле.** После формирования набора акций для включения в портфель, алгоритм определяет их веса.

Алгоритм решает задачу (2), в которой последнее слагаемое является по сути регуляризацией весов. Параметр  $\lambda$ , отвечающий за комиссию при ребалансировке портфеля, в работе принят равным 1%.

Помимо этого, если используются методы уменьшения размерности, то для решения задачи необходимо перейти из латентного пространства в исходное путем обратного преобразования  $\bar{w} = \phi^{-1}(\bar{w}_q)$ . Положим  $\bar{r}$  – вектор средних доходностей в латентном пространстве,  $C$  – выборочная матрица ковариаций в латентном пространстве. Тогда преобразованная задача оптимизации примет вид:

$$\begin{aligned} \min_{\bar{w}_q} \quad & \frac{1}{2} \phi^{-1}(\bar{w}_q)^\top C \phi^{-1}(\bar{w}_q) - \alpha \bar{r}^\top \phi^{-1}(\bar{w}_q) + \frac{\lambda}{P} \left\| \phi^{-1}(\bar{w}_q) - \bar{w}_p \right\|^2 \\ \text{s.t.} \quad & \bar{\Gamma}^\top \phi^{-1}(\bar{w}_q) = 1, \\ & \phi^{-1}(\bar{w}_q) \geq \bar{0}. \end{aligned} \quad (3)$$

В случае с PCA обратное преобразование осуществляется посредством домножения весов на ма-

трицу компонент:  $\phi^{-1}(\bar{w}_q) = Q \bar{w}_q$ . Получается задача квадратичного программирования, которая хорошо решается с помощью пакета *cvxpy* [27].

В методе KernelPCA вектор  $\phi^{-1}(\bar{w}_q)$  в исходном пространстве находится посредством поиска приближительного прообраза вектора  $\bar{w}_q$  решением задачи минимизации с помощью Ridge-регрессии [28]:

$$\phi^{-1}(\bar{w}_q) = \arg \min_{\bar{z} \in \mathbb{R}^d} \left\| \phi(\bar{z}) - \bar{w}_q \right\|^2.$$

Данная задача не является задачей квадратичного программирования, и поэтому решается менее точно с помощью библиотеки *scipy.optimize* [29].

Обратное преобразование весов  $\bar{w} = \phi^{-1}(\bar{w}_q)$  происходит по формулам выше вместе с отсечением весов по пороговому значению: если  $w_i < \text{threshold}$ , то  $w_i = 0$ . Пороговое значение установлено порядка  $10^{-6}$ , чтобы отсекал слишком маленькие веса, которые в реальности невозможно получить, так как необходимо покупать целые количества лотов акций.

**Оценка доходности портфеля.** После обучения алгоритма на тренировочном датасете мы можем собрать портфель, используя полученные веса, и оценить его работу на тестовом датасете. Однако портфель с постоянными весами показывает себя хуже, так как распределение доходностей акций меняется со временем. Поэтому периодически необходимо ребалансировать портфель: через *period\_change\_portfolio* торговых дней алгоритм обучается на тренировочных данных и уже известной ему тестовой части данных. Период смены портфеля в алгоритме выбирался довольно длительным: он мог варьироваться от трех–четырёх месяцев до нескольких лет. Это связано с тем, что метод Марковица неплохо работает на больших промежутках времени с редкой сменой портфеля. Например, в [30] портфели с минимальным риском, т.е. классические портфели Марковица, подвергались ребалансировке 2–3 раза в год.

Для оценки доходности портфеля используется метод *predict*, который принимает на вход обученный алгоритм и тестовый датасет. Результатом работы метода является датафрейм с индексом – временем и колонкой – рассчитанным доходом. После этого возможна визуализация результатов в виде временного ряда доходностей портфеля, а также расчет экономических показателей посредством метода *score*.

## 2.2. Подбор гиперпараметров

В работе рассмотрено четыре вариации исходного алгоритма: без перехода в пространство меньшей размерности, переход с использованием PCA и KernelPCA с полиномиальным и гауссовским ядром. В каждом из методов проводился подбор параметров двумя способами: с помощью случайного поиска по сетке *RandomizedSearchCV* [31] и байесовской оптимизации. Таким образом, получилось восемь алгоритмов с различными наборами гиперпараметров.

В рассматриваемых способах нахождения оптимальных весов производился подбор следующих гиперпараметров:

- ♦ в методе, который не включает переход в пространство меньшей размерности, проводился только выбор гиперпараметров предобработки матрицы доходностей. Эти параметры включают период смены портфеля (в промежутке от трех месяцев до двух с половиной лет), фактор размера окна (от 2 до 5) и количество топ-компаний, отбираемых на основе их средней доходности (от 50 до 200);
- ♦ в методе PCA, помимо вышеназванных параметров, была подобрана также доля объясненной дисперсии (от 80% до 95%);
- ♦ в KernelPCA с полиномиальным ядром были подобраны гиперпараметры, указанные выше, плюс параметры полиномиального ядра – степень (от 2 до 4), свободный член (от 0,5 до 1) и параметр порядка (от 0,02 до 0,04).
- ♦ в методе KernelPCA с гауссовским ядром, помимо параметров предобработки матрицы доходностей и доли объяснимой дисперсии, был подобран параметр порядка (от 0,001 до 0,1). Стоит также отметить, что согласно статье [8], портфель, сформированный с помощью преобразования KernelPCA с гауссовским ядром, является более рискованным, поэтому диверсифицировать риск в данном случае было решено меньше: количество *n\_top\_companies* по доходности было оптимизировано в диапазоне от 5 до 30 (для gbf и Bayes rbf).

## 3. Метрики сравнения портфелей

Для сравнения методов оптимизации портфелей в работе рассмотрены несколько метрик.

**Доходность портфеля (portfolio value).** Отношение текущей цены портфеля к начальной:

$$V_p(\tau) = \prod_{t=1}^{\tau} \left(1 + \bar{r}_t^{\top} \bar{w}_t\right).$$

**Среднегодовая доходность портфеля (average rate of return, AR).** Средняя инвестиционная прибыль, получаемая за год. Этот показатель используется для сравнения доходности различных инвестиционных инструментов. Пусть  $L$  – суммарный срок инвестиций в годах,  $T$  – суммарное число торговых дней. Тогда среднегодовую доходность можно вычислить следующим образом:

$$AR_p = \left(V_p(T)\right)^{\frac{1}{L}} - 1.$$

**Стоимость под риском (value at risk, VaR) портфеля уровня  $\alpha$ .** Обратный выборочный  $\alpha$ -квантиль доходностей портфеля, вычисляемый по формуле:

$$VaR_p(\alpha) = -\hat{q}_{\alpha}^{\top} \left(\bar{r}_t^{\top} \bar{w}_t\right).$$

**Премия портфеля за риск (избыточная доходность).** Выигрыш от держания портфеля по сравнению с безрисковым активом. Для инвестиционного портфеля  $\mathcal{P}$  с весами активов  $\bar{w}_t$ , относительной доходностью  $\bar{r}_t$  и безрисковой процентной ставкой  $rf_t$  в момент  $t = 1, \dots, T$ :

$$\mu_p = \frac{1}{T} \sum_{t=1}^T \left(\bar{r}_t^{\top} \bar{w}_t - rf_t\right).$$

**Стандартное отклонение портфеля.** Выборочное стандартное отклонение избыточной доходности, вычисляемое по формуле:

$$\sigma_p = \frac{1}{T-1} \sqrt{\sum_{t=1}^T \left(\bar{r}_t^{\top} \bar{w}_t - rf_t - \mu_p\right)^2}.$$

**Коэффициент Шарпа (Sharpe ratio).** Отношение премии за риск к стандартному отклонению портфеля:

$$ShR_p = \frac{\mu_p}{\sigma_p}.$$

**Относительная просадка (relative drawdown).** Относительная разница максимальной цены портфеля до настоящего момента и текущей его цены. Формально, если  $V_p(t)$  – отношение цены портфеля в момент  $t$  к начальной, то

$$RD_p(t) = 1 - \frac{V_p(t)}{\max_{\tau=1, \dots, t} V_p(\tau)}.$$

**Коэффициент Мартина (Martin Ratio, Ulcer performance index, UPI) [10].** Отношение избыточной до-

ходности портфеля к среднеквадратичной относительной просадке:

$$PF_p = \frac{\mu_p}{\sqrt{RD_p^2}}$$

Коэффициент Мартина используется для подбора гиперпараметров на этапе валидации.

#### 4. Описание данных

Для проведения исследования использовались данные, полученные из базы Yahoo с использованием библиотеки *yfinance* [32]. В перечень анализируемых активов были включены акции компаний, входящих в индекс S&P 500, а также 500 случайно отобранных акций из базы данных Yahoo. Таким образом, выбор активов обеспечивает независимость весов, полученных в результате работы алгоритма, от весов, построенных только на основе бенчмарка. Сформированный датасет включает дневную динамику стоимости по 1000 акциям за период с 1990 по 2022 год.

Далее была проведена предварительная фильтрация данных, исключающая акции, где более чем в 10% наблюдений отсутствовали значения. Для остальных наблюдений, где отсутствовали значения, доходность была определена как среднее значение за весь предыдущий исторический период. Также из рассмотрения были исключены акции с высокой волатильностью, стоимость которых изменялась более чем в два раза по сравнению с предыдущим днем хотя бы один раз.

После проведенной фильтрации было отобрано 300 акций, представляющих различные секторы. Среди них оказались компании из ИТ-сектора, такие как Apple и Microsoft, из финансовой сферы – JPMorgan

Chase и Citigroup, а также представители медицинской отрасли, включая Johnson & Johnson и Pfizer. Кроме того, в список вошли компании из сектора потребительских товаров: PepsiCo, The Coca-Cola Company, McDonald's, Procter & Gamble и Walmart.

Полученные данные были разделены на два периода: обучающий (1990–2016 гг.) и тестовый (2017–2022 гг.). Для подбора гиперпараметров использовалась кросс-валидация для временных рядов. Методика такой кросс-валидации предполагает разделение данных на блоки, идущие последовательно по времени. Обучение модели происходит на всех предыдущих блоках данных, а проверка – на последующем (*рис. 1*). Данный процесс повторяется несколько раз и может быть реализован методом *TimeSeriesSplit* из *sklearn* [33].

#### 5. Результаты

Конфигурации подобранных гиперпараметров для всех методов представлены в *таблице 1*. Стоит отметить, что у большинства методов оптимальный период смены портфеля составил примерно полгода (180 дней), что может указывать на глобальную оптимальность данного временного интервала. Интересно отметить, что для ядерных моделей фактор масштаба ядра приблизительно одинаков и составляет около 0,04.

После проведения процессов подбора гиперпараметров с использованием байесовской оптимизации и метода поиска по случайной сетке для четырех методов была вычислена доходность на тестовой выборке. Графики доходностей на тестовом наборе данных для всех восьми алгоритмов и базового бенчмарка S&P 500 приведены на *рис. 2*, а также доступны на интерактивном HTML-графике [34].

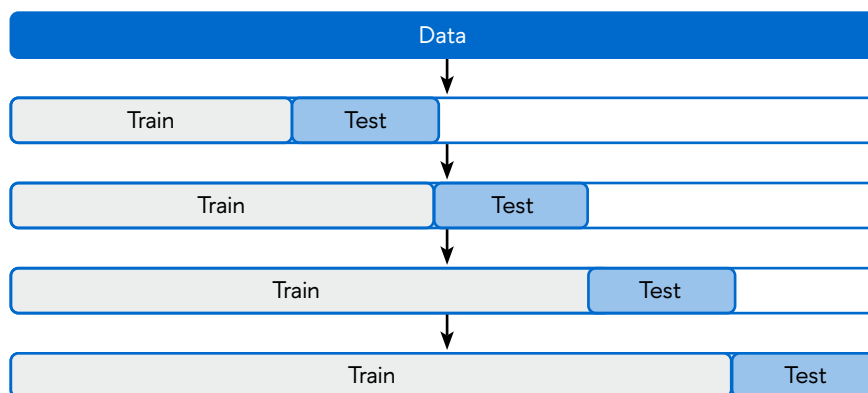


Рис. 1. Визуализация кросс-валидации *TimeSeriesSplit* из *sklearn*.

Таблица 1.

## Подобранные гиперпараметры для всех 8 алгоритмов

	size_of_window_rank	period_change_portfolio	n_top_companies	var_ratio	kernel	kernelgamma	kerneldegree	kernelcoef0
<b>Без оптимизации</b>	3,20	190	97	–	–	–	–	–
<b>РСА</b>	4,90	192	150	0,935	–	–	–	–
<b>Poly KPCA</b>	2,10	249	86	0,86	poly	0,04	4	0,77
<b>rbf</b>	4,90	170	27	0,91	rbf	0,04	–	–
<b>Bayes</b>	4,79	190	100	–	–	–	–	–
<b>Bayes PCA</b>	3,45	189	71	0,891	–	–	–	–
<b>Bayes Poly KPCA</b>	3,37	190	66	0,921	poly	0,034	2	0,844
<b>Bayes rbf</b>	2,44	188	17	0,764	rbf	0,044	–	–

Помимо графиков, были рассчитаны экономические показатели из раздела 3. Для каждого из портфеля и S&P 500 они представлены в *таблице 2*.

Результаты исследования показывают, что наиболее эффективным алгоритмом является метод Bayes rbf, использующий гауссовское ядро в KernelPCA, подобранный с помощью байесовской

оптимизации. Портфель, оптимизированный с помощью данного метода, в период февраля – апреля 2020 г., связанного с первой вспышкой пандемии COVID-19, имеет наименьшую просадку 28,5%, что на 5 п.п. ниже уровня просадки S&P 500. Данное наблюдение подтверждает результат исследования [8], демонстрирующего, что алгоритмы с га-



Рис. 2. Значения доходности на тестовом датасете для всех 8 алгоритмов и S&amp;P 500.



Таблица 2.

## Экономические показатели на тесте для всех 8 алгоритмов и S&amp;P 500

	S&P 500	Без оптимизации	PCA	Poly KPCA	rbf	Bayes	Bayes PCA	Bayes Poly KPCA	Bayes rbf
Профит фактор	1,09	1,1	1,13	1,09	1,09	1,1	1,1	1,09	1,15
Кoeffициент Шарпа	0,41	0,57	0,65	0,46	0,63	0,58	0,53	0,45	0,85
Кoeffициент Мартина	0,26	1,05	1,9	0,71	1,47	1,11	1,09	0,54	3,64
Годовая доходность	10,87%	18,75%	18,11%	12,9%	17,17%	18,71%	16,54%	13,01%	26,98%
Годовая волатильность	20,26%	28,14%	23,97%	22,5%	27,71%	27,83%	26,08%	23,13%	28,51%
5% VaR	1,935%	2,815%	2,4%	2,113%	2,715%	2,817%	2,442%	2,151%	2,897%
Максимальная просадка	33,92%	44,41%	38,04%	43,51%	37,03%	42,99%	43,29%	34,1%	28,27%
Winning days	54,5%	54,2%	55,3%	53,6%	53,6%	54,1%	54,1%	53,4%	54,0%

уссовским ядром показывают лучшие результаты на рынке кризисных ситуаций. В то же время алгоритмы с гауссовским ядром rbf и Bayes rbf демонстрируют наиболее высокую волатильность (27,7% и 28,5% соответственно). Этот результат обусловлен ограниченным числом рассматриваемых акций и, следовательно, меньшей диверсификацией риска портфеля.

Наименьшую волатильность, помимо индекса S&P 500, показали портфели с полиномиальным ядром Poly KPCA и Bayes Poly KPCA: 22,5% и 23,1% соответственно. Это свидетельствует о способности полиномиального ядра эффективно фильтровать шумы. Кроме того, данные портфели характеризуются наилучшими значениями 5% VaR (около 2,1%), что немного больше значения для S&P 500. Однако среднегодовые значения доходности этих портфелей одни из самых низких – 12,9% и 13,0%.

Портфели PCA и Bayes PCA, основанные на методе главных компонент, показали средние результаты по риску и доходности. Это свидетельствует о том, что даже такие классические алгоритмы с дополнительной оптимизацией превзошли бенчмарки по большинству метрик.

В то же время результаты исследования не позволяют установить, что байесовская оптимизация

или случайный поиск по сетке более эффективны при отборе гиперпараметров.

Хотя S&P 500 характеризуется наименьшим риском, его доходность также низка, что отразилось на показателях отношений. Таким образом, построенные портфели, хоть и более волатильны, чем S&P 500, показывают лучшие результаты по остальным метрикам, демонстрируя, что оптимальная перебалансировка портфеля может обеспечить дополнительную доходность за счет риска.

Чтобы привести полученные нами портфели в соответствие с уровнем риска S&P, т.е. среднегодовой волатильностью индекса S&P рассмотрим результаты алгоритма Bayes rbf в случае, если часть денег вкладывается в данный алгоритм, а оставшаяся – в безрисковый актив (таблица 3). Заметим, что если 50% капитала вкладывается в лучший алгоритм, а 50% в безрисковый актив, то доходность и риск портфеля лучше по сравнению с индексом S&P. Если ограничить риск риском, соответствующим индексу, то среднегодовая доходность вырастет с 10,9% до 19,9%. Соответственно, рассматриваемый алгоритм на протяжении длительного периода времени и получать доходность выше S&P при том же уровне риска.

Таблица 3.

## Сравнение лучшего алгоритма и S&amp;P 500

	S&P 500	Bayes rbf	S&P-risk Bayes rbf	0% Bayes rbf	50% Bayes rbf	75% Bayes rbf	90% Bayes rbf
Годовая доходность	10,87%	26,98%	19,93%	2,61%	14,8%	20,89%	24,54%
Годовая волатильность	20,26%	28,51%	20,26%	0,0%	14,26%	21,38%	25,66%
Коэффициент Шарпа	0,41	0,85	0,85	0,0	0,85	0,85	0,85

## Заключение

В работе был исследован алгоритм долгосрочного инвестирования, основанный на решении задачи Марковица с предварительным переходом в пространство меньшей размерности при помощи различных вариаций метода главных компонент.

Периодическая перебалансировка портфеля позволяет гибко реагировать на изменения на рынке, что подтверждает сравнимая с бенчмарком просадка некоторых моделей, несмотря на более высокий показатель риска. Однако перебалансировка портфеля проводилась не слишком часто, чтобы метод Марковица смог показать себя на достаточно больших промежутках времени. Наконец, оптимизация коэффициента Мартина при подборе гиперпараметров показывает высокую доходность портфелей и не слишком резкую просадку.

Помимо этого, были получены портфели, показывающие более доходные траектории, чем бенчмарк, однако немного более рискованные. Важно

отметить, что риск портфелей можно регулировать посредством коэффициента risk aversion, что позволяет подбирать портфели, исходя из предпочтений инвестора относительно риска. Риск можно регулировать и с помощью выбираемых методов: метод с ядерным PCA с полиномиальным ядром подбирает менее рискованный, но и менее доходный портфель, в то время как метод без уменьшения размерности позволяет больше рисковать, но и при этом больше получать. Золотой серединой в этой дилемме может оказаться простой метод главных компонент, который позволяет соблюдать баланс между риском и доходностью. Отметим также, что лучшими экономическими показателями обладает алгоритм, использующий ядерный анализ главных компонент с гауссовским ядром в основном за счет незначительной просадки в пандемийный период в сравнении с другими алгоритмами. Можно сделать вывод, что в кризисные моменты наиболее эффективно применение гауссовского ядра к доходностям акций в качестве перехода в пространство меньшей размерности. ■

## Литература

1. Market capitalization of listed domestic companies. World Federation of Exchanges database // The World Bank. [Электронный ресурс]: <https://data.worldbank.org/indicator/CM.MKT.LCAP.CD> (дата обращения 20.11.2023).
2. Абрамов А.Е., Косырев А.Г., Радыгин А.Д., Чернова М.И. Поведение частных инвесторов на фондовых рынках России и США // Мониторинг экономической ситуации в России. 2021. Вып. 18 С. 11–16.
3. Markowitz H. Portfolio Selection // The Journal of Finance. 1952. Vol. 7. No. 1. P. 77–91. <https://doi.org/10.2307/2975974>
4. Durall R. Asset allocation: From Markowitz to deep reinforcement learning // arXiv:2208.07158. 2022. <https://doi.org/10.48550/arXiv.2208.07158>
5. Best M.J., Grauer R.R. On the sensitivity of mean–variance-efficient portfolios to changes in asset means: Some analytical and computational results // The Review of Financial Studies. 1991. Vol. 4. No. 2. P. 315–342. <https://doi.org/10.1093/rfs/4.2.315>
6. Stefatos G., Hamza A.B. Cluster PCA for outliers detection in high-dimensional data // Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics, Montreal, QC, Canada. 2007. P. 3961–3966. <https://doi.org/10.1109/ICSMC.2007.4414244>
7. Saha B.N., Ray N., Zhang H. Snake validation: A PCA-based outlier detection method // IEEE Signal Processing Letters. 2009. Vol. 16. No. 6. P. 549–552. <https://doi.org/10.1109/LSP.2009.2017477>

8. Peng Y., Albuquerque P.H.M., do Nascimento I.F., Machado J.V.F. Between nonlinearities, complexity, and noises: An application on portfolio selection using kernel principal component analysis // *Entropy*. 2019. Vol. 21. No. 4. Article 376. <https://doi.org/10.3390/e21040376>
9. Ma Y., Han R., Wang W. Portfolio optimization with return prediction using deep learning and machine learning // *Expert Systems with Applications*. 2021. Vol. 165. Article 113973. <https://doi.org/10.1016/j.eswa.2020.113973>
10. Heaton J.B., Polson N., Witte J. Deep learning for finance: deep portfolios // *Applied Stochastic Models in Business and Industry*. 2016. Vol. 33. No. 1. P. 3–12. <https://dx.doi.org/10.2139/ssrn.2838013>
11. Chen K., Zhou Y., Dai F. A LSTM-based method for stock returns prediction: A case study of china stock market // *Proceedings of the 2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, USA. 2015. P. 2823–2824.
12. Yun H., Lee M., Kang Y.S., Seok J. Portfolio management via two-stage deep learning with a joint cost // *Expert Systems with Applications*. 2020. Vol. 143. Article 113041. <https://doi.org/10.1016/j.eswa.2019.113041>
13. Chong E., Han C., Park F. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies // *Expert Systems with Applications*. 2017. Vol. 83. P. 187–205. <https://doi.org/10.1016/j.eswa.2017.04.030>
14. Fischer T., Krauss C. Deep learning with long short-term memory networks for financial market predictions // *European Journal of Operational Research*. 2018. Vol. 270. No. 2. P. 654–669. <https://doi.org/10.1016/j.ejor.2017.11.054>
15. Hoseinzade E., Haratizadeh S. CNNpred: Cnn-based stock market prediction using a diverse set of variables // *Expert Systems with Applications*. 2019. Vol. 129. P. 273–285. <https://doi.org/10.1016/j.eswa.2019.03.029>
16. Kim J., Lee M. Portfolio optimization using predictive auxiliary classifier generative adversarial networks // *Engineering Applications of Artificial Intelligence*. 2023. Vol. 125. Article 106739. <https://doi.org/10.1016/j.engappai.2023.106739>
17. Siaw R., Ofosu-Hene E., Tee E. Investment Portfolio Optimization with GARCH Models // *Elk Asia Pacific Journal of Finance and Risk Management*. 2017. Vol. 8. No. 2. [Электронный ресурс]: <https://ssrn.com/abstract=2987932> (дата обращения 4.08.2024).
18. Bardenet R., Bengio Y., Bergstra J., Kégl B. Algorithms for hyper-parameter optimization // *Proceedings of the Advances in Neural Information Processing Systems 24 (NIPS 2011)*. 2011.
19. Martin P.G., McCann B.B. The investor's guide to fidelity funds // *John Wiley & Sons*, 1989.
20. S&P 500. S&P Dow Jones Indices // S&P Global. [Электронный ресурс]: <https://www.spglobal.com/spdji/en/indices/equity/sp-500/#overview> (дата обращения 20.11.2023).
21. Beneish M.D., Whaley R.E. A scorecard from the S&P game // *Journal of Portfolio Management*. 1997. Vol. 16. No. 2. Article 23.
22. Latham S., Braun M. Does short-termism influence firm innovation? An examination of S&P 500 firms 1990–2003 // *Journal of Managerial Issues*. 2010. Vol. 22. No. 3. P. 368–382.
23. Zhang Z. Study of portfolio performance under certain restraint comparison: Markowitz Model and Single Index Model on S&P 500 // *Proceedings of the 2022 7th International Conference on Social Sciences and Economic Development*. 2022. P. 1930–1938. <https://doi.org/10.2991/aebmr.k.220405.323>
24. Lien G. Non-parametric estimation of decision makers' risk aversion // *Agricultural Economics*. 2002. Vol. 27. No. 1. P. 75–83. [https://doi.org/10.1016/S0169-5150\(01\)00063-9](https://doi.org/10.1016/S0169-5150(01)00063-9)
25. Fama E.F., MacBeth J. Risk, return, and equilibrium: Empirical tests // *Journal of Political Economy*. 1973. Vol. 71. P. 607–636.
26. Paoletta M.S. The univariate collapsing method for portfolio optimization // *Econometrics*. 2017. Vol. 5. No. 2. Article 18. <https://doi.org/10.3390/econometrics5020018>
27. CVXPY 1.4 Manual // The CVXPY authors. [Электронный ресурс]: <https://www.cvxpy.org/index.html> (дата обращения 20.11.2023).
28. Bakir G., Weston J., Schölkopf B. Learning to find pre-images // *Proceedings of the Advances in Neural Information Processing Systems 16 (NIPS 2003)*. 2003.
29. SciPy v1.11.4 Manual // The SciPy community. [Электронный ресурс]: <https://docs.scipy.org/doc/scipy/tutorial/optimize.html> (дата обращения 20.11.2023).
30. Drenovak M., Rankovic V. Markowitz portfolio rebalancing with turnover monitoring // *Economic Horizons*. 2014. Vol. 16. No. 3. P. 207–217. <https://doi.org/10.5937/ekonhor1403211D>
31. API Reference. sklearn.model\_selection.RandomizedSearchCV // scikit-learn. Machine Learning in Python. [Электронный ресурс]: [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.RandomizedSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html) (дата обращения 20.11.2023).
32. yfinance documentation // GitHub. [Электронный ресурс]: <https://yfinance.readthedocs.io/en/documentation/> (дата обращения 20.11.2023).
33. API Reference. sklearn.model\_selection.TimeSeriesSplit // scikit-learn. Machine Learning in Python. [Электронный ресурс]: [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.TimeSeriesSplit.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html) (дата обращения 20.11.2023).
34. Интерактивный график результатов исследования. [Электронный ресурс]: <https://disk.yandex.ru/d/5nF7RiXKIWp1ig> (дата обращения 20.11.2023).

## Об авторах

### Куликов Александр Владимирович

к.ф.-м.н.;

доцент, кафедра высшей математики, Московский физико-технический институт, Россия, 141701, Московская область, г. Долгопрудный, Институтский пер., д. 9;

E-mail: avkulikov15@gmail.com

ORCID: 0000-0002-3963-0814

### Полозов Дмитрий Сергеевич

магистрант, Московский физико-технический институт, Россия, 141701, Московская область, г. Долгопрудный, Институтский пер., д. 9;

E-mail: polozov.ds@phystech.edu

ORCID: 0009-0008-1108-4955

### Волков Никита Васильевич

аспирант, Московский физико-технический институт, Россия, 141701, Московская область, г. Долгопрудный, Институтский пер., д. 9;

E-mail: nikita.v.volkov@phystech.edu

ORCID: 0009-0007-8434-9822

# Long-term investment optimization based on Markowitz diversification

## Alexander V. Kulikov

E-mail: avkulikov15@gmail.com

## Dmitriy S. Polozov

E-mail: polozov.ds@phystech.edu

## Nikita V. Volkov

E-mail: nikita.v.volkov@phystech.edu

Moscow Institute of Physics and Technology, Dolgoprudny, Russia

## Abstract

The article introduces a long-term investment algorithm that identifies optimal solutions in lower dimensional spaces constructed through principal component analysis or kernel principal component analysis. Portfolio weights optimization is carried out using the Markowitz method. Hyperparameters of the model include window size, smoothing parameter, rebalancing period and the fraction of explained variance in dimensionality reduction methods. The algorithm presented incorporates weights regularization taking into account portfolio rebalancing transaction costs. Hyperparameters' selection is based on the Martin coefficient, which allows us to consider the maximum drawdown for the suggested algorithms. The results demonstrate that the proposed algorithm, trained from 1990 to 2016, shows higher returns and Sharpe ratios compared to the S&P 500 benchmark from 2017 to 2022. This indicates that weights optimization can improve the algorithm's performance through rebalancing.

**Keywords:** PCA, Kernel PCA, window size, Markowitz algorithm, Grid Search, Bayesian optimization

**Citation:** Kulikov A.V., Polozov D.S., Volkov N.V. (2024) Long-term investment optimization based on Markowitz diversification. *Business Informatics*, vol. 18, no. 3, pp. 56–69. DOI: 10.17323/2587-814X.2024.3.56.69

## References

1. The World Bank Group (2023) *Market capitalization of listed domestic companies*. *World Federation of Exchanges database*. Available at: <https://data.worldbank.org/indicator/CM.MKT.LCAP.CD> (accessed 20 November 2023).
2. Abramov A.E., Kosyrev A.G., Radygin A.D., Chernova M.I. (2021) Behavior of private investors in the stock markets of Russia and the US. *Russian Economic Developments*, vol. 18, pp. 11–16 (in Russian).
3. Markowitz H. (1952) Portfolio selection. *The Journal of Finance*, vol. 7, no. 1, pp. 77–91. <https://doi.org/10.2307/2975974>
4. Durall R. (2022) Asset allocation: From Markowitz to deep reinforcement learning. *arXiv:2208.07158*. <https://doi.org/10.48550/arXiv.2208.07158>
5. Best M.J., Grauer R.R. (1991) On the sensitivity of mean–variance-efficient portfolios to changes in asset means: Some analytical and computational results. *The Review of Financial Studies*, vol. 4, no. 2, pp. 315–342. <https://doi.org/10.1093/rfs/4.2.315>
6. Stefatos G., Hamza A.B. (2007) Cluster PCA for outliers detection in high-dimensional data. *Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics, Montreal, QC, Canada*, pp. 3961–3966. <https://doi.org/10.1109/ICSMC.2007.4414244>.
7. Saha B.N., Ray N., Zhang H. (2009) Snake validation: A PCA-based outlier detection method. *IEEE Signal Processing Letters*, vol. 16, no. 6, pp. 549–552. <https://doi.org/10.1109/LSP.2009.2017477>
8. Peng Y., Albuquerque P.H.M., do Nascimento I.F., Machado J.V.F. (2019) Between nonlinearities, complexity, and noises: An application on portfolio selection using kernel principal component analysis. *Entropy*, vol. 21, no. 4, 376. <https://doi.org/10.3390/e21040376>
9. Ma Y., Han R., Wang W. (2021) Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, vol. 165, 113973. <https://doi.org/10.1016/j.eswa.2020.113973>
10. Heaton J.B., Polson N., Witte J. (2016) Deep learning for finance: Deep portfolios. *Applied Stochastic Models in Business and Industry*, vol. 33, no. 1, pp. 3–12. <https://doi.org/10.2139/ssrn.2838013>
11. Chen K., Zhou Y., Dai F. (2015) A LSTM-based method for stock returns prediction: A case study of China stock market. *Proceedings of the 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, USA*, pp. 2823–2824. <https://doi.org/10.1109/BigData.2015.7364089>
12. Yun H., Lee M., Kang Y.S., Seok J. (2020) Portfolio management via two-stage deep learning with a joint cost. *Expert Systems with Applications*, vol. 143, 113041. <https://doi.org/10.1016/j.eswa.2019.113041>
13. Chong E., Han C., Park F. (2017) Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, vol. 83, pp. 187–205. <https://doi.org/10.1016/j.eswa.2017.04.030>
14. Fischer T., Krauss C. (2018) Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669. <https://doi.org/10.1016/j.ejor.2017.11.054>
15. Hoseinzade E., Haratizadeh S. (2019) CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, vol. 129, pp. 273–285. <https://doi.org/10.1016/j.eswa.2019.03.029>
16. Kim J., Lee M. (2023) Portfolio optimization using predictive auxiliary classifier generative adversarial networks. *Engineering Applications of Artificial Intelligence*, vol. 125, 106739. <https://doi.org/10.1016/j.engappai.2023.106739>
17. Siaw R., Ofosu-Hene E., Tee E. (2017) Investment portfolio optimization with GARCH models. *Elk Asia Pacific Journal of Finance and Risk Management*, vol. 8, no. 2. Available at: <https://ssrn.com/abstract=2987932> (accessed 04 August 2024).
18. Bardenet R., Bengio Y., Bergstra J., Kégl B. (2011) Algorithms for hyper-parameter optimization. *Proceedings of the Advances in Neural Information Processing Systems 24 (NIPS 2011)*.
19. Martin P.G., McCann B.B. (1989) *The investor's guide to fidelity funds*. John Wiley & Sons.
20. S&P Global (2023) *S&P 500. S&P Dow Jones Indices*. Available at: <https://www.spglobal.com/spdji/en/indices/equity/sp-500/#overview> (accessed 20 November 2023).
21. Beneish M.D., Whaley R.E. (1997) A scorecard from the S&P game. *Journal of Portfolio Management*, vol. 16, no. 2, 23.
22. Latham S., Braun M. (2010) Does short-termism influence firm innovation? An examination of S&P 500 firms 1990–2003. *Journal of Managerial Issues*, vol. 22, no. 3, pp. 368–382.
23. Zhang Z. (2022) Study of portfolio performance under certain restraint comparison: Markowitz Model and Single Index Model on S&P 500. *Proceedings of the 2022 7th International Conference on Social Sciences and Economic Development*, pp. 1930–1938. <https://doi.org/10.2991/aebmr.k.220405.323>
24. Lien G. (2002) Non-parametric estimation of decision makers' risk aversion. *Agricultural Economics*, vol. 27, no. 1, pp. 75–83. [https://doi.org/10.1016/S0169-5150\(01\)00063-9](https://doi.org/10.1016/S0169-5150(01)00063-9)
25. Fama E.F., MacBeth J. (1973) Risk, return, and equilibrium: Empirical tests. *Journal of Political Economy*, vol. 71, pp. 607–636.

26. Paoletta M.S. (2017) The univariate collapsing method for portfolio optimization. *Econometrics*, vol. 5, no. 2, 18. <https://doi.org/10.3390/econometrics5020018>
27. The CVXPY authors (2023) *CVXPY 1.4 Manual*. Available at: <https://www.cvxpy.org/index.html> (accessed 20 November 2023).
28. Bakir G., Weston J., Schölkopf B. (2003) Learning to find pre-images. Proceedings of the *Advances in Neural Information Processing Systems 16 (NIPS 2003)*.
29. The SciPy community (2023) *SciPy v1.11.4 Manual*. Available at: <https://docs.scipy.org/doc/scipy/tutorial/optimize.html> (accessed 20 November 2023).
30. Drenovak M., Rankovic V. (2014) Markowitz portfolio rebalancing with turnover monitoring. *Economic Horizons*, vol. 16, no. 3, pp. 207–217. <https://doi.org/10.5937/ekonhor1403211D>
31. scikit-learn. Machine Learning in Python (2023) *API Reference. sklearn.model\_selection. RandomizedSearchCV*. Available at: [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.RandomizedSearchCV.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html) (accessed 20 November 2023).
32. GitHub (2023) *yfinance documentation*. Available at: <https://yfinance.readthedocs.io/en/documentation/> (accessed 20 November 2023).
33. scikit-learn. Machine Learning in Python (2023) *API Reference. sklearn.model\_selection. TimeSeriesSplit*. Available at: [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.TimeSeriesSplit.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html) (accessed 20 November 2023).
34. *Interactive chart of research results*. Available at: <https://disk.yandex.ru/d/5nF7RiXKIWp1ig> (accessed 20 November 2023).

### About the authors

#### Alexander V. Kulikov

Cand. Sci. (Phys.-Math.);

Associate Professor, Moscow Institute of Physics and Technology, 9, Institutskiy Ln., Dolgoprudny 141701, Moscow Region, Russia;

E-mail: [avkulikov15@gmail.com](mailto:avkulikov15@gmail.com)

ORCID: 0000-0002-3963-0814

#### Dmitriy S. Polozov

Master's Student, Moscow Institute of Physics and Technology, 9, Institutskiy Ln., Dolgoprudny 141701, Moscow Region, Russia;

E-mail: [polozov.ds@phystech.edu](mailto:polozov.ds@phystech.edu)

ORCID: 0009-0008-1108-4955

#### Nikita V. Volkov

Doctoral Student, Moscow Institute of Physics and Technology, 9, Institutskiy Ln., Dolgoprudny 141701, Moscow Region, Russia;

E-mail: [nikita.v.volkov@phystech.edu](mailto:nikita.v.volkov@phystech.edu)

ORCID: 0009-0007-8434-9822