



**Разработка инвестиционной программы-помощника на базе
математических моделей формирования цены инвестиционного портфеля
пользователя**

Вакуленко Р.Я., д.э.н., профессор,

ФГБОУ ВО «Нижегородский государственный лингвистический университет
им. Н.А. Добролюбова», Нижний Новгород, Россия

Лапаев Д.Н., д.э.н., профессор, ФГБОУ ВО «Нижегородский государственный
технический университет им. Р.Е. Алексеева», заместитель директора по
научной работе Института экономики и управления, Нижний Новгород, Россия

Виноградова О.В., к.э.н., доцент, ФГБОУ ВО Национальный
исследовательский университет «Высшая школа экономики», Москва, Россия

Хилал-Шпиро Мухаммед Рияд, бакалавр

ФГБОУ ВО «Нижегородский государственный лингвистический университет
им. Н.А. Добролюбова», Нижний Новгород, Россия

Аннотация. В настоящее время всё больше наблюдается тенденция разработки высокоточных трейдер-ботов, способных выполнять по сто, а то и десятки тысяч операций в день. Данная работа содержит описательное объяснение основ инвестирования в акции и торговли ими, а также некоторых популярных торговых стратегий, в том числе с использованием моделей машинного обучения для прогнозирования движения цен на определенные цены акций, чтобы читатели, не имеющие опыта в инвестировании или Data Science, могли проследить, как программа была разработана определенным образом. А также показывает эффективность разработанного трейдер-бота на примере акций компаний Microsoft, Apple и Goldman Sachs.

Ключевые слова: трейдер-бот, предсказание цен, торговая биржа

Development of an investment assistant program based on mathematical models of the formation of the price of the user's investment portfolio

Vakulenko R.Ya., Doctor of Economics, Professor, Nizhny Novgorod State Linguistic University named after N.A. Dobrolyubov, Nizhny Novgorod, Russia

Lapaev D.N., Doctor of Economics, Professor, Nizhny Novgorod State Technical University named after R.E. Alekseev, Deputy Director for Scientific Work of the Institute of Economics and Management, Nizhny Novgorod, Russia

Vinogradova O.V., Candidate of Economics, Associate Professor, National Research University Higher School of Economics, Moscow, Russia

Hilal-Shpiro Mohammed Riyad, Bachelor, Nizhny Novgorod State Linguistic University named after N.A. Dobrolyubov, Nizhny Novgorod, Russia

Annotation. Currently, there is an increasing tendency to develop high-precision trading bots capable of performing one hundred or even tens of thousands of operations per day. This paper contains a descriptive explanation of the basics of investing in stocks and trading them, as well as some popular trading strategies, including using machine learning models to predict price movements for certain stock prices, so that readers without experience in investing or Data Science can trace how the program was developed in a certain way. And also shows the effectiveness of the developed trader bot on the example of shares of Microsoft, Apple and Goldman Sachs.

Key words: trader bot, price prediction, trading exchange

Широко распространено мнение, что цену акций предсказать невозможно, поэтому в данном проекте мы не стали углубляться в оптимизацию результатов, а представили предложение о том, как использование Machine Learning может вписаться в общую картину.

Работая на фондовом рынке, трейдеру необходимо понимать, что проследить все изменения стоимости акций в мире не только невозможно, но и очень затратно по времени и моральной энергии. Именно поэтому в настоящее время всё больше разрабатываются и совершенствуются боты-трейдеры,

которые могли бы выполнять тысячи, а то и миллионы операций в день и экономить время и деньги всем инвесторам [1, 6].

Развитие темы начнём с определения наиболее приемлемой модели бот-трейдера, представленной на рис. 1. Структура была разработана таким образом, чтобы каждый компонент выполнял определенную задачу, что оказалось выгодным, поскольку код и ход проекта стали гораздо более управляемыми. Таким образом, требование ко всей программе может быть легко переведено в группу конкретных задач. Хорошей демонстрацией этого является то, что из требования, что бот должен быть способен торговать позициями, можно понять, что это обычно включает вычисление индикатора, покупать или продавать, и вызов этой вычислительной задачи ежедневно в определенный момент.

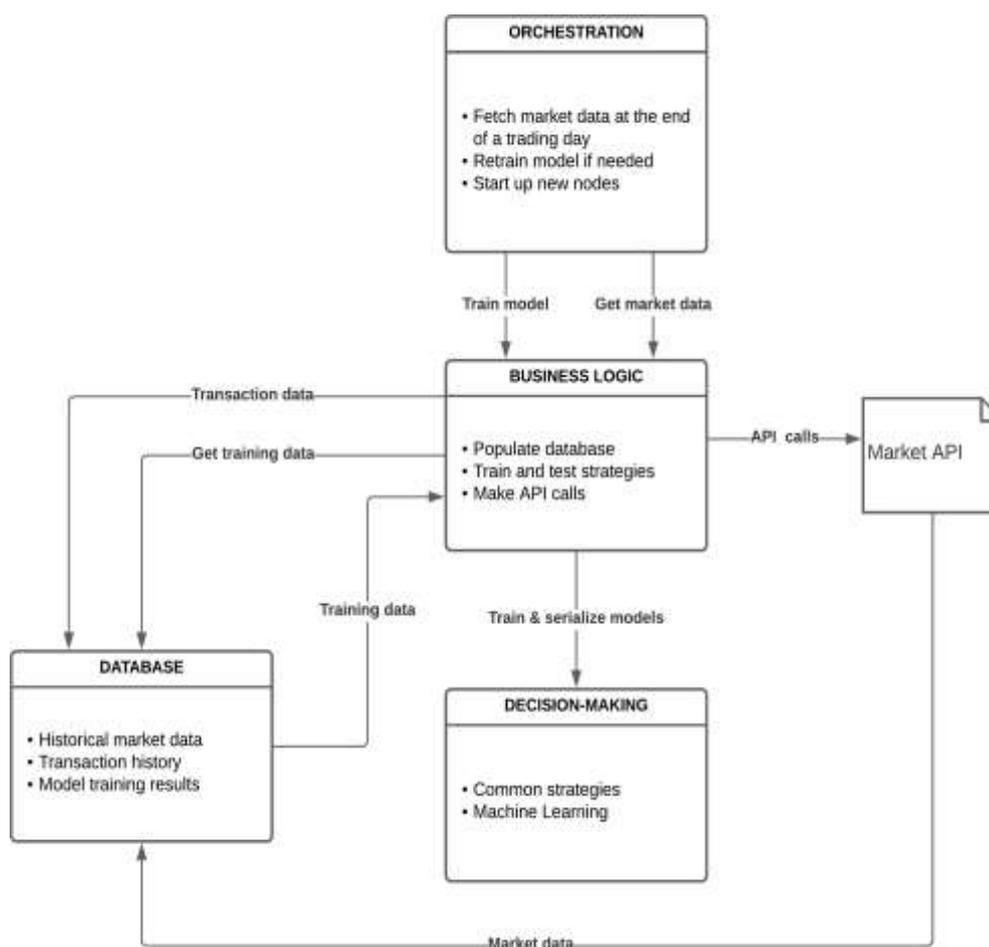


Рис. 1 – Основные взаимодействия между компонентами программы (авторская разработка)

Следовательно, код, который необходимо изменить или написать, должен находиться в модулях оркестровки и принятия решений, и в данном примере он включает функцию для вычисления торгового решения и выражение cron для планирования вызова этой функции.

Согласно рисунку, модуль бизнес-логики является основой этой программы. Он вызывает методы из других модулей для выполнения таких задач, как вставка данных в базу данных или обучение и оценка моделей Machine Learning из модуля принятия решений. Поскольку каждый модуль был разработан для одной конкретной цели, кодовая база может быть легко дополнена другими модулями, перечисленными позже в разделе предложений. Более подробное объяснение каждого компонента будет описано в последующих разделах.

Инфраструктура развертывания представлена на рис. 2.

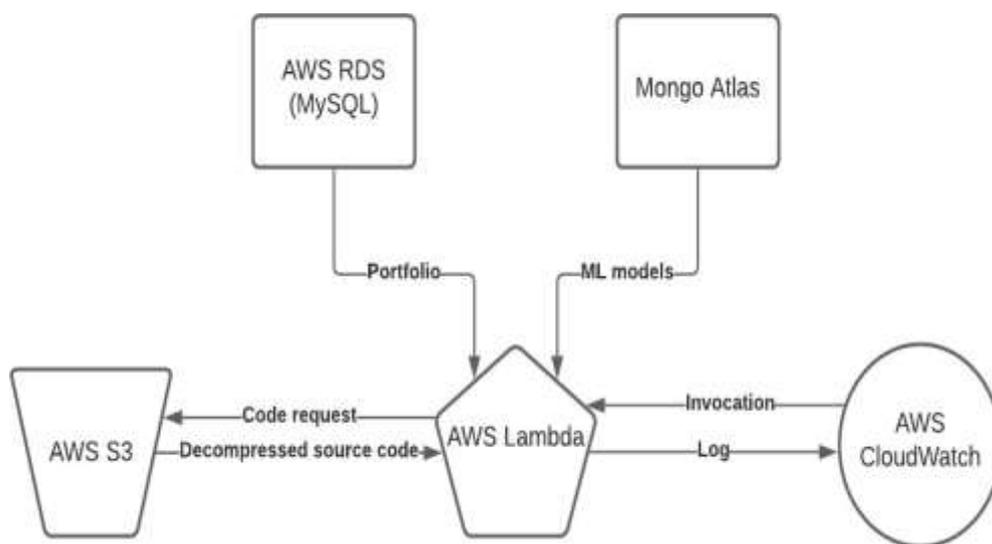


Рис. 2 – Взаимодействие между облачными сервисами для позиционной торговли (авторская разработка)

Хранение информации

Для хранения данных было выбрано решение использовать службу баз данных MySQL, предоставляемую AWS для табличных данных, таких как рыночная цена и исторические транзакции, а Mongo Atlas является платформой для развертывания MongoDB, которая способна хранить нетепличные данные, такие как модели машинного обучения.

В данном проекте были использованы преимущества бесплатных услуг AWS Free Tier, предоставляемых вновь созданным учетным записям. Поскольку вычислительные мощности на этом уровне весьма ограничены, услуги следует использовать только для позиционной торговли, поскольку она не требует значительных ресурсов в небольших масштабах и не должна справляться с обработкой данных в режиме реального времени, что присуще стратегиям дневной торговли.

При задействовании фреймворка Serverless большая часть обеспечения и обслуживания инфраструктуры выполняется автоматически. В целом исходный код и его зависимости были упакованы, затем заархивированы и загружены в AWS S3 с помощью фреймворка Serverless. Затем было создано правило AWS CloudWatch, также с помощью фреймворка Serverless на основе predefined конфигураций разработчика, для вызова функции AWS Lambda за некоторое время до закрытия рынка. После активации функция Lambda сначала загружает исходный код и его зависимости из указанной корзины S3, затем выполняет код для получения торгового профиля, хранящегося в базах данных, а также рыночных данных за этот день и рассчитывает последующие действия на основе этих данных. Если есть какие-либо новые транзакции, они записываются в файл журнала, а также отправляются в базу данных для сохранения для возможного анализа в будущем.

Приведенный далее рис. 3 позволяет как можно ближе имитировать реальный пример, сохраняя при этом упрощенность. Таблица `daily_price` может показаться излишней, поскольку такие данные можно получить практически из любого места. Однако, если необходимо улучшить производительность модели машинного обучения. Это также, может быть, резервный набор данных для обучения новых моделей в случае, если исторические рыночные данные не могут быть получены из оригинального источника из-за простоя сервера или ошибок соединения.

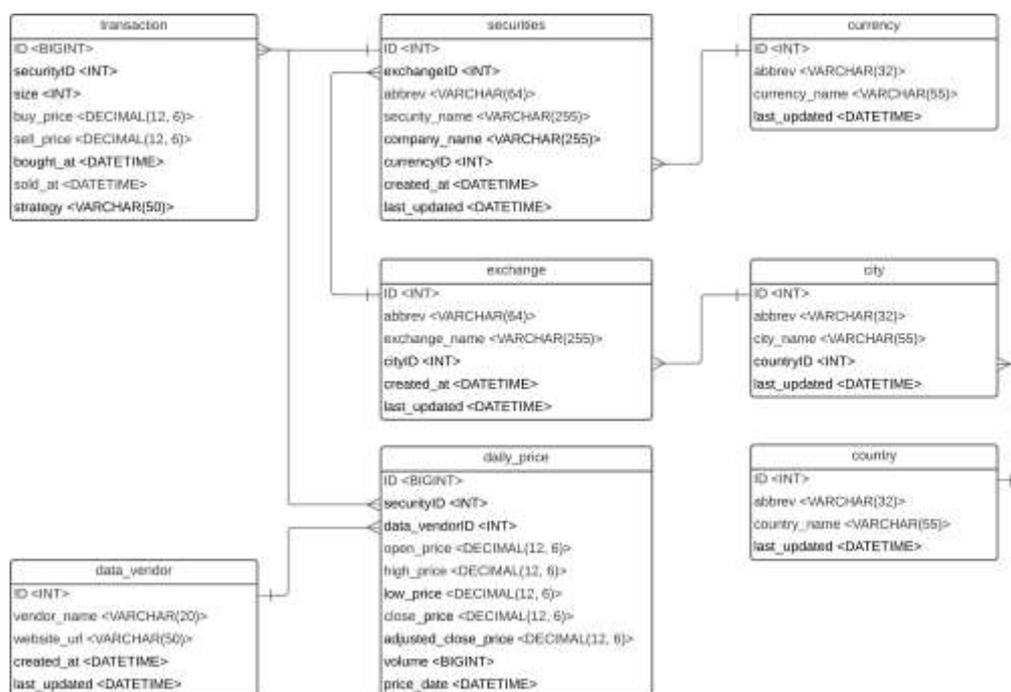


Рис. 3 – Имитация реального примера (авторская разработка)

Текущий торговый профиль хранится в таблице метаданных, которая содержит следующую информацию.

1. **Баланс:** количество денег, не вложенных ни в какие акции. По умолчанию это значение установлено равным 4000.
2. **CurrencyID:** целочисленное значение, относящееся к строке в таблице валют. Изначально валютой был USD.
3. **Current_strategy:** строковое значение, обозначающее название текущей торговой стратегии. Существует две группы: те, которые начинаются с «ML_», и те, которые не начинаются. Как можно было догадаться, первая указывает на то, что используется модель машинного обучения, поэтому скрипт вызовет функцию для ее получения из базы данных MongoDB. Второй относится к стратегиям, которые полагаются на технические индикаторы, поэтому основной скрипт просто загрузит предварительно созданный класс.

Кроме того, список экземпляров модели класса Stock был бинаризован и сохранен в коллекции под названием `owned_stocks` в MongoDB.

Для хранения моделей машинного обучения не существует жестких схем, которым необходимо следовать, поэтому в базе данных Mongo под названием

algotrading_models была создана коллекция с названием models, в которой хранились документы, содержащие как минимум следующие поля.

1. Model: фактический объект ML в Python, который был «замаринован», или, другими словами, превращен в массив байтов и сохранен в базе данных в виде строки.

2. Model_name: имя модели, которое было гарантированно уникальным для облегчения поиска.

3. Created_at & last_trained: время, когда модель была создана и обучена соответственно.

Языком программирования, использованным в данном проекте, был Python. Это было обусловлено следующими причинами.

1. Большая поддержка сообщества: Python есть многоцелевой язык программирования, но благодаря своему минималистичному синтаксису он обычно используется для обработки данных и автоматизации, что также является темой данного проекта. Поэтому в случае возникновения ошибок было гораздо проще получить помощь от других.

2. Простота: синтаксис Python отличается минимализмом. Он не является сильно типизированным и не включает в себя чрезмерное количество классов, что упрощает такие задачи, как преобразование типов данных.

3. Библиотеки сторонних разработчиков: Язык программирования Python имеет множество пакетов для обработки данных, и они могут помочь еще больше абстрагировать код, сохраняя его чистым. Таким образом, количество фактически написанного кода может быть уменьшено, а его читабельность повышена.

Функциональные требования

Данный модуль должен отвечать за принятие торговых решений, которые включают в себя покупку или продажу, а также количество акций. Поэтому, получив набор рыночных данных, состоящий из данных OHLC и определенных технических индикаторов, он должен уметь выполнять вычисления на них для оценки ситуации и получения вышеупомянутого значения.

Нефункциональные требования

Наиболее важным критерием при оценке хода разработки данного модуля является прибыльность используемой стратегии. Она будет считаться успешной и приемлемой, если стратегия приносит более 30% дохода или, по крайней мере, безубыточна соответственно.

Метод, используемый для оценки стратегии, заключается в том, чтобы посмотреть, как она работает на исторических рыночных данных. Для этой цели существуют два популярных претендента под названием `backtesting.py` и `Zipline`. После нескольких дней исследований и экспериментов был выбран `backtesting.py` за его открытый API, простую установку и настройку тестов.

При использовании скальпинговой стратегии действуют те же требования, но методы тестирования и оценки результатов отличаются незначительно. Поскольку фреймворк `backtesting.py` не поддерживает внутрисуточную торговлю, вместо использования готовых инструментов необходимо разработать симуляционный тест, который циклически просматривает все точки данных. По сути, он прокручивает каждую строку таблицы рыночных данных, вызывает функцию принятия решений бота и вычисляет сумму оставшегося баланса. Скальпинговая стратегия считается успешной, даже если она приносит маргинальную прибыль или не несет убытков.

Как уже упоминалось ранее, в данном проекте предпринимается попытка получить прибыль с помощью двух подходов: размещение ордеров на основе оценки рынка с помощью комбинации определенных технических индикаторов или на основе прогноза модели машинного обучения.

Чтобы обеспечить равенство при сравнении стратегий, крайне важно, чтобы при их тестировании по отдельности применялись одинаковые условия тестирования. Поэтому начальный инвестиционный капитал был установлен в размере 4000 долларов США, размер комиссии – 2%, а период времени, в течение которого тестировались стратегии, – с апреля 2018 года по март 2021 года, или примерно 3 года.

Стратегией, тестируемой в рамках эксперимента, была скользящая средняя. Используя комбинацию значений скользящих средних за 10 и 15 дней, результат оказался весьма впечатляющим, как и показано на рис. 4.

```

bt = Backtest(data, SmaCross, cash=4000, commission=.002)
stats = bt.run()
stats

Start                2018-04-02 00:00:00
End                  2021-03-30 00:00:00
Duration             1093 days 00:00:00
Exposure Time [%]    96.423841
Equity Final [$]     13847.028388
Equity Peak [$]      17039.512152
Return [%]           246.17571
Buy & Hold Return [%] 198.235923
Return (Ann.) [%]    51.35739
Volatility (Ann.) [%] 50.010715
Sharpe Ratio         1.026928
Sortino Ratio        2.455041
Calmar Ratio         2.085638
Max. Drawdown [%]   -24.624303
Avg. Drawdown [%]   -3.818092
Max. Drawdown Duration 210 days 00:00:00
Avg. Drawdown Duration 20 days 00:00:00
# Trades             23
Win Rate [%]         52.173913
Best Trade [%]       80.59984
Worst Trade [%]      -10.005939
Avg. Trade [%]       5.582115
Max. Trade Duration  188 days 00:00:00
Avg. Trade Duration  46 days 00:00:00
Profit Factor         4.031158
Expectancy [%]       6.987999
SQN                   1.297649
_strategy            SmaCross
_equity_curve        ...
_trades              Size EntryB...
dtype: object

```

Рис. 4 – Результат тестирования стратегии, скользящей средней с апреля 2018 года по март 2021 года

В конце тестового периода на руках было почти 14000 долларов США, что составляет около 250% прибыли. Однако при более глубоком изучении истории торговли есть несколько моментов, на которые следует обратить внимание.

Прежде всего, поскольку это симуляционный тест, основанный на исторических данных, стратегия (см. рис. 5) должна была быть успешной,

особенно при экспериментировании на цене акций такой прибыльной компании, как Apple.

	Size	EntryBar	ExitBar	EntryPrice	ExitPrice	PnL	ReturnPct	EntryTime	ExitTime	Duration
0	88	27	58	45.085687	45.065083	-1.813209	-0.000457	2018-05-09	2018-06-22	44 days
1	-88	58	72	44.974953	46.266052	-113.616706	-0.028707	2018-06-22	2018-07-13	21 days
2	83	72	120	46.358584	53.513555	593.862647	0.154340	2018-07-13	2018-09-20	69 days
3	-83	120	129	53.406528	55.897168	-206.723098	-0.046635	2018-09-20	2018-10-03	13 days
4	76	129	140	56.008962	52.935265	-233.600980	-0.054879	2018-10-03	2018-10-18	15 days
5	-76	140	204	52.829395	37.585803	1158.512957	0.288544	2018-10-18	2019-01-23	97 days
6	137	204	281	37.660975	45.822351	1118.108609	0.216706	2019-01-23	2019-05-14	111 days
7	-138	281	302	45.730707	47.860155	-293.863926	-0.046565	2019-05-14	2019-06-13	30 days
8	125	302	341	47.955876	49.212142	157.033221	0.026196	2019-06-13	2019-08-08	56 days
9	-125	341	351	49.113717	52.604389	-436.333921	-0.071073	2019-08-08	2019-08-22	14 days
10	108	351	479	52.709597	71.081509	1984.166444	0.348550	2019-08-22	2020-02-26	188 days
11	-108	479	509	70.939346	65.179751	622.036241	0.081190	2020-02-26	2020-04-08	42 days
12	127	509	619	65.310111	117.949955	6685.260232	0.805998	2020-04-08	2020-09-15	160 days
13	-127	619	634	117.714055	115.328390	302.979473	0.020267	2020-09-15	2020-10-06	21 days
14	132	634	651	115.559047	112.009093	-468.593839	-0.030720	2020-10-06	2020-10-29	23 days
15	-132	651	663	111.785075	118.742560	-918.388066	-0.062240	2020-10-29	2020-11-16	18 days
16	117	663	675	118.980046	123.335698	509.611299	0.036608	2020-11-16	2020-12-03	17 days
17	-117	675	677	123.089026	122.127500	112.498631	0.007812	2020-12-03	2020-12-07	4 days
18	119	677	703	122.371755	130.604846	979.737891	0.067279	2020-12-07	2021-01-14	38 days
19	-119	703	710	130.343636	143.385742	-1552.010542	-0.100059	2021-01-14	2021-01-26	12 days
20	97	710	723	143.672513	134.350006	-904.283196	-0.064887	2021-01-26	2021-02-12	17 days
21	-97	723	750	134.081306	122.820000	1092.346720	0.083989	2021-02-12	2021-03-24	40 days
22	115	750	754	123.065640	120.110001	-339.898495	-0.024017	2021-03-24	2021-03-30	6 days

Рис. 5 – Подробная торговая история стратегии, скользящей средней с 2018 по 2021 год

При более тщательном рассмотрении можно увидеть, что большая часть прибыли получена от длинной позиции летом 2020 года, которая вернула чуть более 80% от вложенного бюджета. Как и ожидалось, этот взрыв в цене акций Apple произошел как раз перед тем, как Apple выпустила линейку компьютеров MacOS, работающих на чипе Apple M1.

Во-вторых, эта стратегия не очень хорошо проявила себя в случае, когда цена часто колеблется, что демонстрирует ряд с 15 по 20. Цена акций колебалась от примерно 111,78 долларов США в конце октября 2020 года до более чем 143 USD к концу января 2021 года, что могло бы принести прибыль в размере 28,82%, если бы бот купил 132 акции и держал их в течение 3 месяцев. Однако бот неправильно истолковал ситуацию и продолжал покупать и держать акции в

течение более коротких периодов времени, что привело к потере более 10%, когда он попытался совершить короткую сделку по цене 130,34 USD и закрыл позицию, когда цена акций подскочила до уровня выше 143 USD за акцию менее чем через две недели. За вышеупомянутый период бот также пытался совершить две другие короткие сделки, но они не принесли значительной прибыли.

По-видимому, бот не очень хорошо сработал в описанном выше случае, потому что он воспользовался только парой запаздывающих индикаторов, или, другими словами, техническими индикаторами, которые подтверждают тренд только на основе движений в прошлом, или, иногда, импульса роста цены. Это не так эффективно при использовании на волатильном рынке, поскольку текущее движение цены гораздо меньше коррелирует с ее историческими записями, что усложняет для программы принятие торгового решения. Понятно, что следующим шагом для улучшения работы этой программы было создание модели машинного обучения для прогнозирования цены акций и принятия соответствующих решений. Теоретически, это может компенсировать слабость запаздывающих индикаторов, упомянутых выше.

В качестве иллюстрации на рис. 6 и 7 представлены фактические и прогнозные данные с апреля 2018 года по март 2021 года. Каждая выборка состояла из ежедневной информации, включая низкую цену, высокую цену, цену открытия, цену закрытия и объем торгов. В качестве обучающего набора данных были выбраны первые 80% образцов, а тестовые данные составили оставшиеся 20% образцов. Весь анализ был выполнен исключительно на техническом анализе, без использования модели машинного обучения. Все имеющиеся данные использовались для оценки подходящего размера сети, увеличивая количество нейронов скрытого слоя до тех пор, пока эффективное количество параметров не сходилось к постоянному значению.

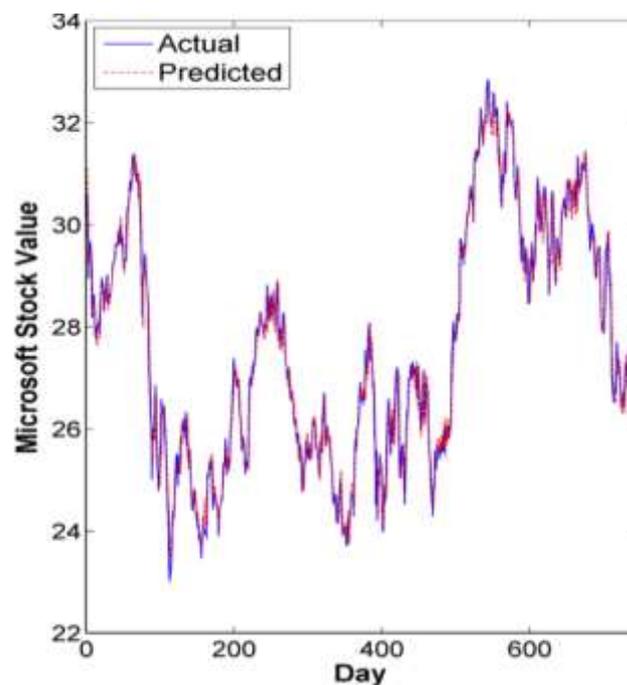


Рис. 6 – Прогноз модели и фактическая цена акций

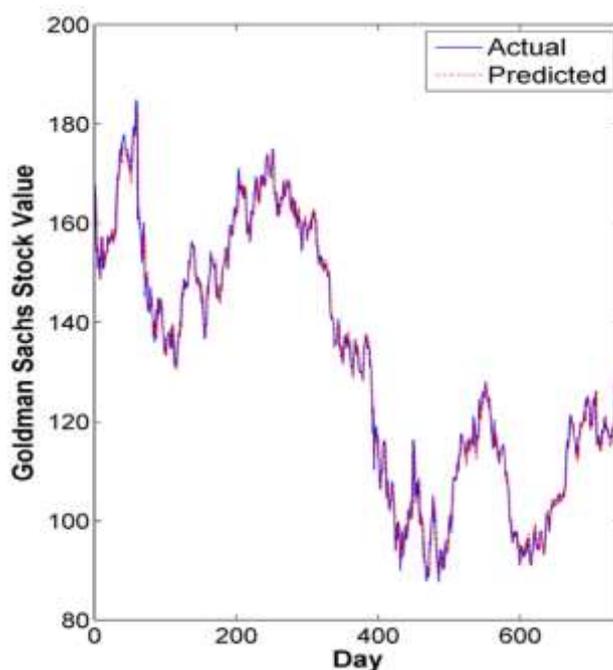


Рис. 7 – Прогноз модели и фактическая цена акций

Также было проведено сравнительное исследование для проверки эффективности сети в данном исследовании с усовершенствованной гибридной моделью Хассана и др. Для этого эксперимента были собраны данные по компаниям Apple Inc. (AAPL) и International Business Machines Corp. (IBM). Этот результат свидетельствует о том, что модель может успешно обрабатывать большие массивы данных со значительным шумом и волатильностью, сохраняя

при этом обобщающую способность. На заключительных рисунках показаны результаты этого эксперимента в виде графика зависимости целевой цены акций от прогнозируемой цены.

Пунктирная линия обозначает линию с наклоном, равным единице, что свидетельствует об идеальном соответствии. Уменьшение ошибки в данных обучения ожидаемо, однако ограниченный разброс в данных тестирования свидетельствует о том, что модель обеспечивает надлежащее обобщение.

Таким образом, была сформулирована модель бота-трейдера, далее был представлен общий вид базы данных, и всех инструментов, которые будут полезными для построения бота-трейдера. Основным этапом явилось построение стратегии торговли на основании скользящей средней, что позволило достичь результатов в 250% к текущему счёту, что весьма достойно. Это дало уверенность в правильности наших научных предположений и расчётов.

Библиографический список:

1. J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh and A. H. Byers, «Big data: The next frontier for innovation, competition and productivity», 2011. [Online PDF] Available: https://www.mckinsey.com/~media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Big%20data%20The%20next%20frontier%20for%20innovation/MGI_big_data_exec_summary.pdf (дата обращения: 17.10.2021).
2. The Forbes, «The Forbes 400», 2020. [Online]. Available: <https://www.forbes.com/forbes-400/> (дата обращения: 17.10.2021).
3. W. Kenton, «Financial Instrument» in Investing Essentials. Investopedia, [online document] 2020. Available: Investopedia, <https://www.investopedia.com/terms/f/financialinstrument.asp> (дата обращения: 17.10.2021).
4. W. Kenton, «Security» in Investing Essentials. Investopedia, [online document] 2020. Available: <https://www.investopedia.com/terms/s/security.asp> (дата обращения: 15.11.2021).

5. A. Hayes, «Trade», in Economics. Investopedia, [online document] 2021. Available: <https://www.investopedia.com/terms/t/trade.asp> (дата обращения: 15.11.2021).

6. W. Kenton, «Exchange» in Investing Essentials. Investopedia, [online document] 2020. Available: <https://www.investopedia.com/terms/e/exchange.asp> (дата обращения: 15.11.2021).

7. Банк С.В. Финансовые инструменты как важнейшее условие регулирования хозяйственной деятельности в рыночной экономике // Финансы и кредит. – 2005. – № 5 (173). – С. 45-51.

8. Адамов Н.А., Банк С.В., Суглобов А.Е., Древинг С.Р., Музалев С.В., Орлова Е.А., Агеева О.А., Донцов И.А. Финансовый менеджмент. — Москва: ИД «Экономическая газета», 2010. – 528 с.

References:

1. J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh and A. H. Byers, «Big data: The next frontier for innovation, competition and productivity», 2011. [Online PDF] Available: https://www.mckinsey.com/~media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Big%20data%20The%20next%20frontier%20for%20innovation/MGI_big_data_exec_summary.pdf (accessed date:17.10.2021).

2. The Forbes, «The Forbes 400», 2020. [Online]. Available: <https://www.forbes.com/forbes-400/> (accessed date:17.10.2021).

3. W. Kenton, «Financial Instrument» in Investing Essentials. Investopedia, [online document] 2020. Available: Investopedia, <https://www.investopedia.com/terms/f/financialinstrument.asp> (accessed date:17.10.2021).

4. W. Kenton, «Security» in Investing Essentials. Investopedia, [online document] 2020. Available: <https://www.investopedia.com/terms/s/security.asp> (accessed date:15.11.2021).

5. A. Hayes, «Trade», in Economics. Investopedia, [online document] 2021. Available: <https://www.investopedia.com/terms/t/trade.asp> (accessed date:15.11.2021).
6. W. Kenton, «Exchange» in Investing Essentials. Investopedia, [online document] 2020. Available: <https://www.investopedia.com/terms/e/exchange.asp> (accessed date:15.11.2021).
7. Bank S.V. Financial instruments as the most important condition for regulating economic activity in a market economy // Finance and Credit. – 2005. – № 5 (173). – Pp. 45-51.
8. Adamov N.A., Bank S.V., Suglobov A.E., Dreving S.R., Muzalev S.V., Orlova E.A., Ageeva O.A., Dontsov I.A. Financial management. – Moscow: Publishing house «Ekonomicheskaya Gazeta», 2010. – 528 p.

Для цитирования: Вакуленко Р.Я., Разработка инвестиционной программы-помощника на базе математических моделей формирования цены инвестиционного портфеля пользователя/ Вакуленко Р.Я., Лапаев Д.Н., Виноградова О.В., Хилал-Шпиро Мухаммед Рияд, //Российский экономический интернет-журнал. – 2022. – № 3. URL:
© Вакуленко Р.Я., Лапаев Д.Н., Виноградова О.В., Хилал-Шпиро Мухаммед Рияд, Российский экономический интернет-журнал 2022, № 3.