

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра _____ Прикладная математика и информатика
(наименование)

09.04.03 Прикладная информатика

(код и наименование направления подготовки)

Прикладной анализ данных

(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему «Методы и модели прогнозирования курса криптовалютных активов на основе анализа временных рядов и факторных данных»

Обучающийся

_____ А.А. Щекин _____

(Инициалы Фамилия)

(личная подпись)

Научный
руководитель

_____ Д.Т.Н., доцент, С.В. Мкртычев _____

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2025

Оглавление

Введение	3
Глава 1 Текущее состояние исследований в области прогнозирования котировок криптовалют	7
Глава 2 Разработка подходов анализа криптовалютных рынков с использованием нейронных сетей.....	13
2.1 Технические, фундаментальные и макроэкономические показатели при прогнозировании.....	13
2.2 Нейронные сети при прогнозировании	17
Глава 3 Анализ данных котировок криптовалют	26
3.1 Методика предобработка данных.....	26
3.2 Апробация предложенной методики предобработка данных	30
3.3 Сравнение эффективности различных типов нейронных сетей ..	33
Глава 4 Программная реализация анализа данных	38
4.1 Программный модуль сбора данных.....	38
4.2 Программный модуль для анализа данных.....	40
4.3 Программный модуль для визуализации данных.....	55
4.4 Тестирование предложенных подходов	60
Заключение	64
Список используемой литературы и используемых источников	67

Введение

Применение технологии blockchain в финансовом секторе привело к появлению децентрализованных цифровых валют, называемых также криптовалютами. Рост популярности криптовалют обусловлен: низкими транзакционными издержками при осуществлении переводов, отсутствие необходимости раскрытия своей личности для осуществления переводов, отсутствие возможности блокирования криптовалютных переводов.

Причисленные выше достоинства цифровых валют привели к приходу крупных участников торгов на криптовалютные биржи. По данным сервиса coinmarketcap.com общая капитализация рынка криптовалют на конец 2023 года составляет 123,78 трлн. рублей. К крупным участникам торгов относятся финансовые компании, инвестиционные банки, а также представители государственного сектора.

Из-за роста популярности криптовалют актуальным становится вопрос прогнозирования котировок криптовалют. Не смотря на сходство природы всех криптовалют, заключающееся в анонимности и децентрализации они различаются между собой по фундаментальным показателям, начиная от индекса волатильности и заканчивая рыночной капитализацией. Поэтому прогнозирование криптовалютных рынков является сложной задачей, требующей применения технологий анализа данных.

Существующие подходы для анализа других рынков, например, фондового не подходят для использования на криптовалютном рынке, так как не учитывают высокую волатильность криптовалют, а также отличия в наборе фундаментальных показателей. Однако нейронные сети, за счет возможности обучения могут подстроиться под данную задачу прогнозирования котировок криптовалют. По этому в данном исследовании предполагается разработать принципы прогнозирования криптовалют с использованием нейронных сетей.

Объект исследования – рынок криптовалют.

Цель исследования – выбор методов и разработка на их основе моделей прогнозирования котировок криптовалютных рынков с использованием нейронных сетей.

Гипотеза исследования состоит в том, что использование нейронных сетей позволяет повысить точность прогнозирования котировок криптовалют, если:

- учтены особенности технических, фундаментальных и макроэкономических показателей;
- применены методы нормализации и снижения размерности данных;
- использованы архитектуры нейронных сетей, адаптированные для анализа временных рядов;
- проведена апробация предложенных подходов на различных типах данных.

Задачи исследования:

- провести анализ существующих подходов к прогнозированию криптовалютных рынков.
- разработать подходы к анализу данных криптовалютных рынков с учетом технических, фундаментальных и макроэкономических показателей.
- провести сравнительный анализ эффективности различных типов нейронных сетей для прогнозирования котировок криптовалют.
- разработать и апробировать программный модуль для анализа и прогнозирования котировок криптовалют.

Теоретико-методологическая основа исследования включает анализ существующих научных работ по использованию нейронных сетей для обработки временных рядов, методы нормализации данных и методы снижения размерности на основе значений SHAP.

Методы исследования: изучение литературных источников, математическое моделирование, программная реализация нейронных сетей и проведение вычислительных экспериментов.

Опытнo-экспериментальная база исследования состоит из данных котировок криптовалют, предоставленных сервисом yahoo!finance.

Научная новизна исследования заключается в:

- адаптации методов анализа данных к специфике криптовалютных рынков;
- применении методов нормализации и снижения размерности данных на основе SHAP;
- сравнительном анализе различных архитектур нейронных сетей для задач прогнозирования криптовалют.

Теоретическая значимость исследования заключается в:

- систематизации подходов к анализу временных рядов криптовалют;
- описанию влияния различных типов данных на точность прогнозирования;
- разработке рекомендаций по выбору нейронных сетей для анализа криптовалютных рынков.

Практическая значимость исследования заключается в разработке программного модуля, который может быть использован для прогнозирования криптовалютных котировок и принятия инвестиционных решений.

Достоверность и обоснованность результатов исследования обеспечивались использованием верифицированных данных, проверкой результатов на нескольких типах данных и сравнением предложенных методов с существующими.

Личное участие автора включало разработку методик анализа данных, программного модуля и проведение вычислительных экспериментов.

По теме исследования опубликованы статья [13].

На защиту выносятся:

- методика предобработки данных криптовалют, включающая нормализацию и снижение размерности.
- результаты сравнительного анализа нейронных сетей для прогнозирования котировок криптовалют.

– программный модуль для анализа данных криптовалютных рынков.

Структура магистерской диссертации состоит из введения, четырёх глав, заключения, списка литературы из 30 источников и приложений. Основной текст изложен на 75 страницах.

Во введении обоснованы актуальность исследования, определены объект, предмет, цель, гипотеза и задачи работы. Также изложены научная новизна и практическая значимость исследования. В первой главе «Текущее состояние исследований в области прогнозирования криптовалют» проведён обзор литературы. Рассмотрены работы, посвящённые использованию нейронных сетей для прогнозирования временных рядов, и выявлены ограничения существующих подходов. Во второй главе «Разработка подходов анализа криптовалютных рынков» описаны используемые технические, фундаментальные и макроэкономические показатели, а также методы нормализации и снижения размерности данных. В третьей главе «Анализ данных котировок криптовалют» приведены результаты экспериментов по влиянию снижения размерности данных на точность прогнозирования. Проведён сравнительный анализ нейронных сетей, таких как LSTM, GRU. В четвёртой главе «Программная реализация анализа данных» представлена разработка программного модуля, включая алгоритмы предобработки данных, моделирования и прогнозирования.

В заключении представлены основные результаты исследования.

Работа изложена на 71 странице и включает 38 рисунков, 10 таблиц и 30 источников.

Глава 1 Текущее состояние исследований в области прогнозирования котировок криптовалют

В работе [24] под авторством Huang освещаются вопросы применения нейронных сетей для прогнозирования временных рядов в финансовом секторе. В данном исследовании делается вывод, что нейронные сети подходят для моделирования нелинейных зависимостей содержащихся в данных временных рядов. Также говорится, что такие типы зависимостей преобладают в задачах прогнозирования будущей цены финансового актива по косвенным показателям. В работе также демонстрируется не линейный характер зависимостей между ценой актива, его дивидендами и объемами торгов. Кроме того, описывается зависимость между индексом акций и различными индикаторами.

В работе [30] под авторством Zoumprekas приведены результаты исследования по предсказанию цены криптовалюты ETH за счет использования рекуррентных нейронных сетей. В исследование предлагается в качестве входных сигналов использовать по дням: цену криптовалюты на момент закрытия торгов и средневзвешенную цену криптовалюты за день. Автор сравнивает эффективность различных архитектур нейронных сетей, среди которых сети CNN, LSTM и ее вариации, а также GRU. Сделан вывод, что LSTM сети точнее работают в задаче прогнозирования цены на криптовалюту ETH. При этом в нейронах сети применяется функция активации tanh, а также L2 регуляризация.

В работе [14] под авторством Alonso-Monsalve исследуется возможность определения тренда криптовалют на основе технических индикаторов. Для предсказания тренда криптовалют в исследовании используются «1 min» котировки. Также стоит отметить, что предсказания строятся на основе котировок 6 наиболее популярных криптовалют (рисунок 1).

		Bitcoin	Dash	Ether	Litecoin	Monero	Ripple
Train	0	49.86%	70.95%	54.68%	60.47%	74.44%	62.81%
	1	50.14%	29.05%	45.32%	38.53%	25.56%	37.19%
Validation	0	49.64%	70.79%	54.55%	60.65%	74.45%	62.74%
	1	50.36%	29.21%	44.45%	39.35%	25.55%	37.36%
Test	<i>Baseline</i> 0	50.00%	70.62%	54.67%	60.88%	74.69%	62.81%
	1	50.00%	29.38%	45.33%	39.12%	25.31%	37.19%

Рисунок 1 – Таблица с результатами прогнозирования котировок, приведенных в исследовании [14]

В качестве входных сигналов используются около 20 технических индикаторов. В работе показаны результаты тестирования двух нейронных сетей прямого распространения и двух рекуррентных нейронных сетей. Авторы докладывают, что наилучшей точностью прогнозирования обладает сеть Contextual LSTM (рисунок 2). В работе приведены детальные параметры данной нейронной сети.

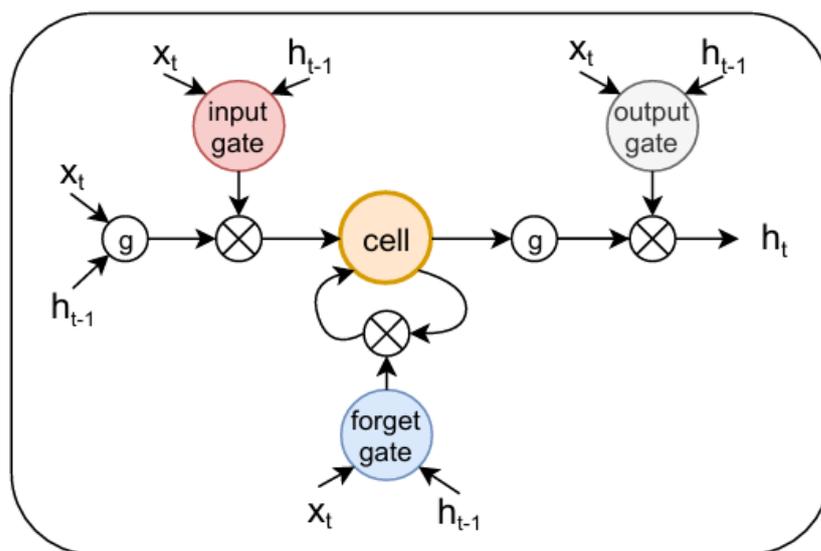


Рисунок 2 – Предложенная в исследовании [14] структура LSTM блока

В работе [26] под авторством Neely рассматривается вопрос эффективности прогнозирования котировок на основе анализа технических индикаторов. В исследовании сделан вывод, что прогнозирование котировок возможно как при отдельном использовании технических индикаторов, так и при их использовании совместно с макроэкономическими индикаторами (рисунок 3). В последнем случае точность прогнозирования котировок выше, так как эти индикаторы дополняют друг друга.

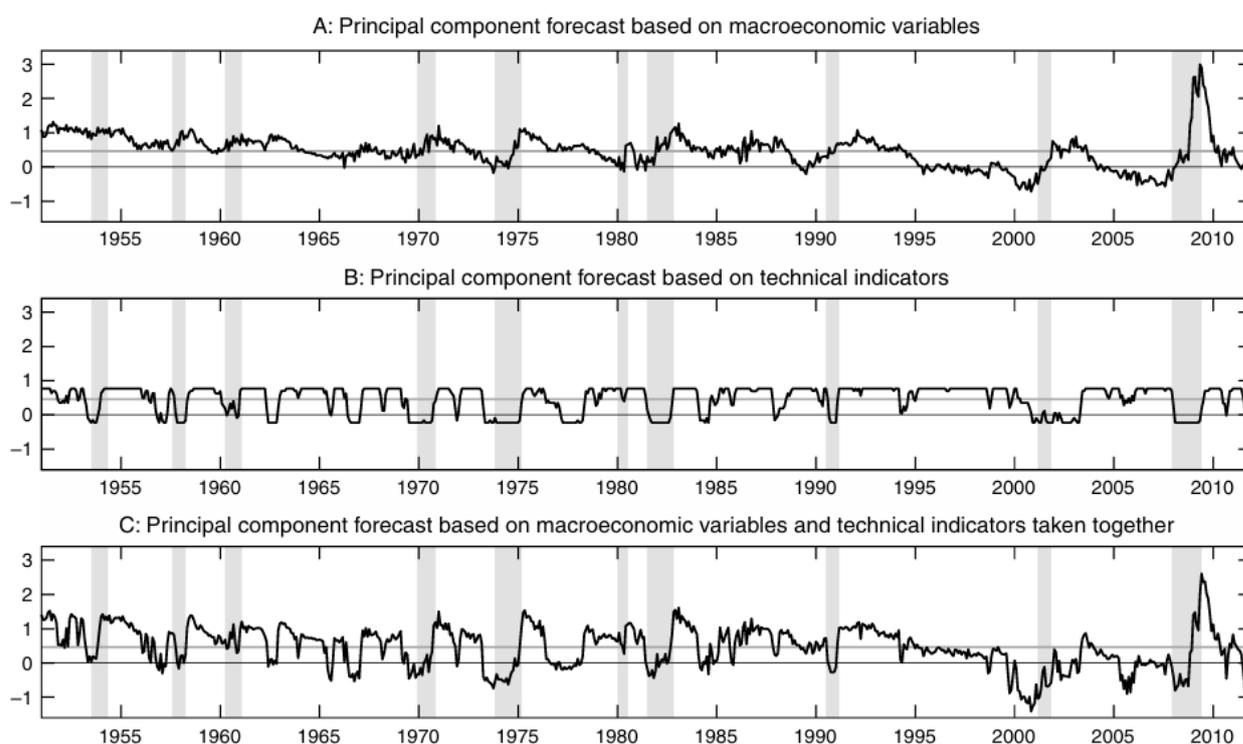


Рисунок 3 – Пример из работы [26], прогнозирование на основе макроэкономических переменных (график А), технических индикаторов (график В) и суммирование двух графиков (график С)

В работе [22] под авторством Hansson исследовался вопрос представления задачи прогнозирования котировок в виде задач классификации и регрессии. При представлении прогнозирования в виде задачи классификации определяется направление изменения доходности и ее относительное изменение в следующем периоде. В работе представлены результаты прогнозирования индекса акций таких стран как Швеция и США.

В качестве инструмента прогнозирования используются нейронные сети LSTM. В результатах работы докладывается о возможности использования LSTM сетей в задаче прогнозирования котировок финансовых активов (рисунок 4).

FIGURE 8. Softmax deep LSTM trading strategy.

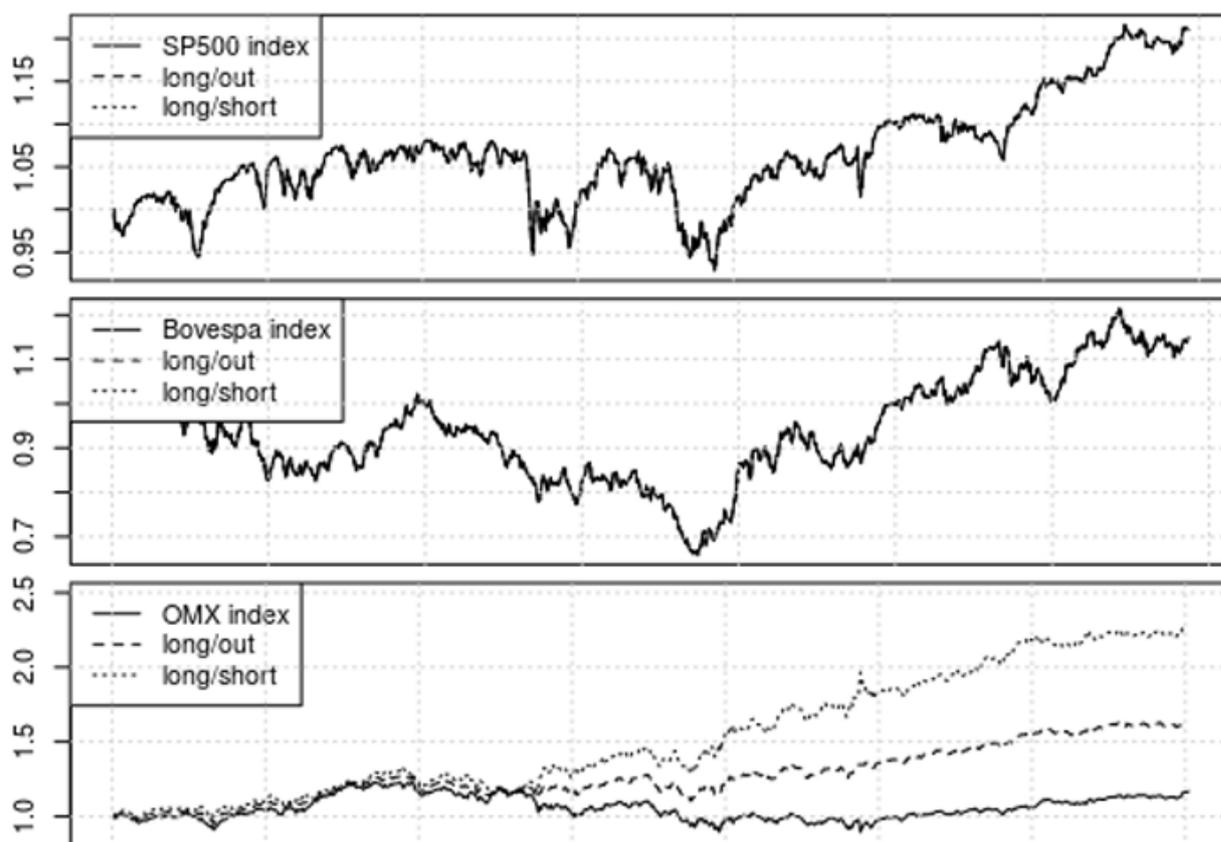


Рисунок 4 – Стратегия трейдинга на основе LSTM сети из работы [22]

В работе [20] под авторством Dutta исследуется вопрос прогнозирования котировок криптовалюты BTC с использованием рекуррентных нейронных сетей. В исследовании сравнивается работа двух нейронных сетей – GRU и LSTM. В качестве входных сигналов используется около 20 фундаментальных признаков. В работе предложена методика отсеивание незначимых признаков для снижения размерности обучающей выборки более чем на 20%. Авторы докладывают, что снижение размерности обучающей выборки позволило

повысить скорость обучений нейронной сети. В результатах исследования говорится о том, что сеть GRU позволила достичь более высокой точности прогнозирования котировок BTC, чем при использовании сети LSTM (рисунок 5).

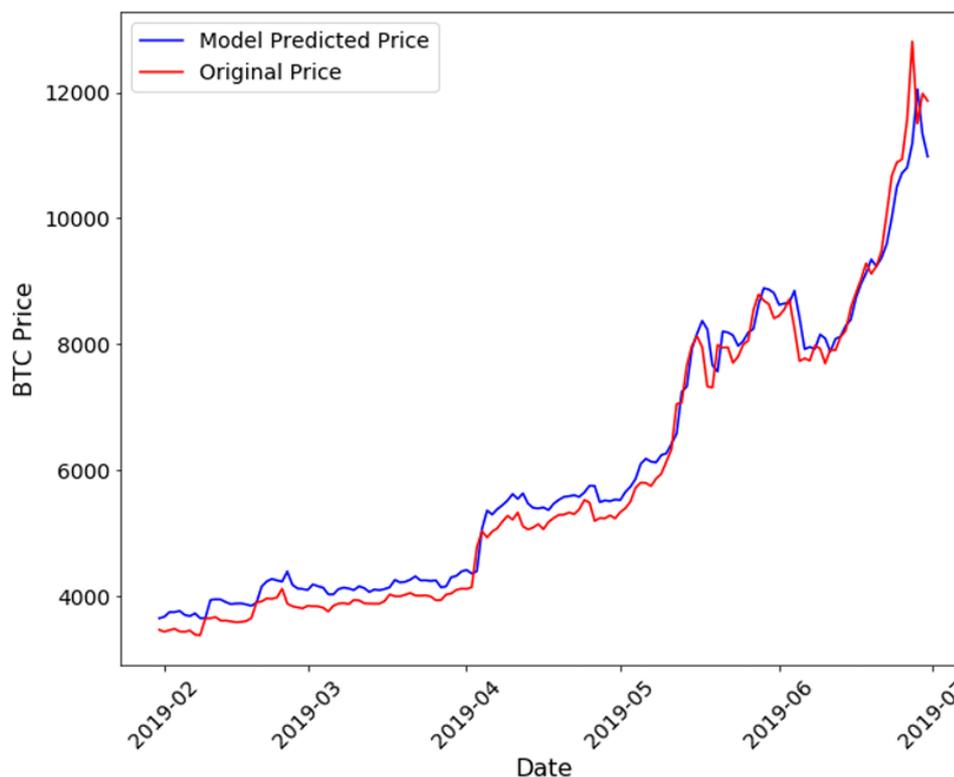


Рисунок 5 – Использование GRU сети для прогнозирования цены BTC из работы [20]

В работе [29] под авторством Yildirim исследуется вопрос совместного использования технических и макроэкономических индикаторов для прогнозирования направления изменения котировок на рынке Forex. В исследовании приведено сравнения двух подходов к построению системы прогнозирования котировок. В первом подходе используется нейронная сеть LSTM, которая обучается на всем наборе признаков (рисунок 6). Во втором подходе используются две LSTM сети, объединенные в ансамбль: первая обучается на технических индикаторах, а вторая на обучается на

макроэкономических индикаторах.

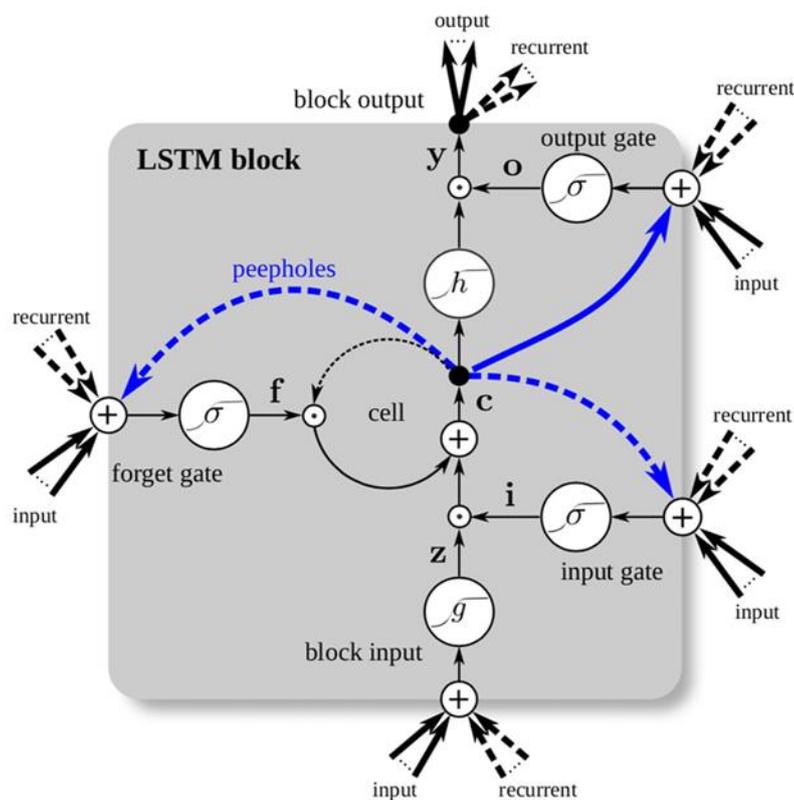


Рисунок 6 – LSTM блок сети из исследования [29]

Авторы докладывают, что ансамбль сетей LSTM показывает лучшие результаты прогнозирования, чем отдельная LSTM сеть, обученная на всех имеющихся данных.

Выводы по главе 1

Обобщим результаты проведенный анализ литературных источников по теме прогнозирования котировок криптовалют. Существующие исследования имеют следующие недостатки: рассматривается вопрос прогнозирования одной криптовалюты без обобщения полученных результатов на другие, не используется весь потенциал технологий машинного обучения (например, снижение размерности датасета и кросс-валидация), при прогнозировании осуществляется анализ признаков только из одного типа индикаторов (или фундаментальных или технических).

Глава 2 Разработка подходов анализа криптовалютных рынков с использованием нейронных сетей

2.1 Технические, фундаментальные и макроэкономические показатели при прогнозировании

Технические индикаторы – это функции, которые на основе значений параметров торгов оценивает будущее направление изменения котировок. Алгоритм расчета любого технического индикатора заранее определен и не меняется со временем. Технический индикатор нельзя рассматривать, как полноценный прогноз изменения котировок по причине того, что каждый индикатор оперирует малым набором признаков и сконцентрирован на оценке только один из множества факторов (например, только волатильности актива).

Для расчета значений технических индикаторов криптовалют будем использовать библиотеку Technical Analysis написанную для языка программирования Python [1], [5], [7].

Обозначение технических индикаторов и их ключевые параметры, используемые в расчетах показаны в таблице 1.

Таблица 1 – Технические индикаторы для прогнозирования криптовалют

«Класс индикатора	Обозначение индикатора	Параметры индикатора
Индикаторы текущего момента	RSI	Временное окно = 14
	TSI	Временное окно slow = 25, Временное окно fast = 13.
	ROC	Временное окно = 12
Индикаторы объема сделок	VPT	–
	EoM	Временное окно = 14
	OBV	–
	ADI	Временное окно = 14» [13]

Продолжение таблицы 1

«Класс индикатора»	Обозначение индикатора	Параметры индикатора
Индикаторы волатильности	ATR	Временное окно = 14
	Bollinger High Band	Временное окно = 20, Временное окно dev = 2.
	Bollinger Low Band	Временное окно = 20, Временное окно dev = 2.
Индикаторы линии тренда	EMA	Временное окно = 14
	MACD	Временное окно slow = 26, Временное окно fast = 12, Временное окно sign = 9.
	Parabolic SAR	Шаг = 0,02, Максимальный шаг = 0,2.
	WMA	Временное окно = 5, Временное окно = 10, Временное окно = 20, Временное окно = 30, Временное окно = 60.
	SMA	Временное окно = 5, Временное окно = 10, Временное окно = 20, Временное окно = 30, Временное окно = 60» [13]

Все технические индикаторы делятся на 4 группы:

- индикаторы текущего момента, которые направлены на определение направления движения котировок в текущий момент времени;
- индикаторы объема сделок, которые определяют движение котировок на основе учета заявок на покупку и продажу актива за выбранный промежуток времени;
- индикаторы волатильности, которые направлены на определения движения котировок за счет оценки силы колебания цены актива за выбранный промежуток времени;
- индикаторы тренда, которые направлены на определение движения котировок на основе предыдущих значений цены актива за выбранный промежуток времени.

При прогнозировании котировок криптовалюты мы будем использовать все технические индикаторы, представленные в таблице 1. Фундаментальные показатели – это метрики финансового актива, рассчитываемые на основе текущих параметров торгов. Рассчитать фундаментальный показатель означает найти его числовое значение (таблица 2).

Таблица 2 – Фундаментальные показатели криптовалют

Показатель	Описание
AdrActCnt	Суммарное количество уникальных адресов кошельков, которые были активны в блокчейн сети в этот день
AssetEODCompletionTime	Последнее время расчета показателей
BlkCnt	Количество блоков, добавленных в блокчейн за день.
BlkSizeByte	Размер блока в байтах
BlkSizeMeanByte	Средний размер всех блоков, созданных за день.
CapMVRVCur	Отношение суммарной стоимости USD текущего предложения к сумме реализованных USD текущего предложения.
CapMrktCurUSD	Сумма USD стоимости текущего предложения (т.н. сетевой стоимостью или рыночной капитализацией).
CapRealUSD	Суммарное значение USD, основанное на цене закрытия USD в день последнего перемещения нативного блока для всех нативных блоков.
DiffMean	Средняя сложность нахождения хэша за день
FeeMeanNtv	Средняя комиссия за транзакцию в монетах криптовалюты за день
FeeMeanUSD	Среднее значение комиссии в USD за одну транзакцию за день
FeeMedNtv	Медианная комиссия за транзакцию в монетах криптовалюты за день
FeeMedUSD	Медианная комиссия за транзакцию в USD за день
FeeTotNtv	Сумма всех вознаграждений, выплачиваемых майнерам, валидаторам транзакций, стейкерам и/или производителям блоков за день.
HashRate	Средняя скорость, с которой майнеры рассчитывают хэши за день
IssContNtv	Сумма USD, выпущенных в этот день.
IssContPctAnn	Процент новых монет криптовалюты выданных за день, экстраполированный на один год (т.е. умножается на 365) и разделённый на текущее предложение.
IssContUSD	Сумма в долларах США стоимости новых монет криптовалюты, выпущенных за день

Продолжение таблицы 2

IssTotNtv	Сумма всех новых монет криптовалюты, выпущенных в этот день.
IssTotUSD	Сумма в USD стоимости новых монет криптовалюты, выпущенных в этот день.
NVTAdj	Отношение стоимости блокчейн сети к скорректированной трансфертную стоимости.
NVTAdj90	Отношение стоимости сети к 90-дневному скользящему среднему значению скорректированной трансфертной стоимости.
PriceBTC	Фиксированная цена закрытия актива по состоянию на 00:00 UTC следующего дня, выраженная в BTC.
PriceUSD	Фиксированная цена закрытия актива по состоянию на 00:00 UTC следующего дня, выраженная в USD.
ROI1yr	Рентабельность инвестиций за 1 год
ROI30d	Рентабельность инвестиций за 30 дней
SplyCur	Сумма всех монет криптовалюты, когда-либо созданных на этот день.
SplyExpFut10yrCMBI	Данная метрика является улучшенной версией устаревшей метрики SplyExpFut10yr
SplyFF	Сумма всех монет криптовалюты, когда-либо созданных, за исключением монет, находящихся во владении инсайдерских компаний, контролирующих инвесторов и долгосрочных стратегических держателей на этот день
TxCnt	Количество транзакций за день
TxTfrCnt	Количество транзакций за выбранный интервал времени
TxTfrValAdjNtv	Сумма монет криптовалюты, переданных за день
TxTfrValAdjUSD	Сумма монет криптовалюты, переданных за день, выраженная в USD
TxTfrValMeanNtv	Суммарное значение переданных монет криптовалюты, деленное на количество транзакций между отдельными адресами в этом интервале
TxTfrValMeanUSD	Суммарное значение переданных монет криптовалюты, деленное на количество транзакций между отдельными адресами в этом интервале, выраженное в USD
TxTfrValMedNtv	Медианное число монет криптовалюты, переданное на один трансфер в этот день.
TxTfrValMedUSD	Медианное число монет криптовалюты, переданное на один трансфер в этот день, выраженное в USD
TxTfrValNtv	Сумма монет криптовалюты, переданных в этот день.
TxTfrValUSD	Сумма монет криптовалюты, переданных в этот день, выраженная в USD
VtyDayRet180d	Волатильность за 180 дней, измеряемая как отклонение логарифмической доходности

Помимо технических индикаторов и фундаментальных показателей при прогнозировании котировок финансовых активов применяются также макроэкономические показатели. Макроэкономические показатели – это те показатели, которые напрямую не связаны с рассматриваемым финансовым активом, но тем не менее оказывают значительное влияние на его котировки из-за их фундаментальной значимости для экономик всего мира.

Общепризнанными макроэкономическими показателями являются:

- индекс доллара DXY;
- средневзвешенная процентная ставка по кредитам LiborUSD;
- индекс фондового рынка США S&P500;
- индекс волатильности VIX;
- цена золота GLD.

Приведенные макроэкономические показатели будут также использованы при прогнозировании котировок криптовалют.

2.2 Нейронные сети при прогнозировании

При прогнозировании котировок криптовалют по техническим индикаторам, а также фундаментальным и макроэкономическим показателям будут применяться нейронные сети.

Самой распространённой архитектурой нейронной сети является многослойный персептрон [2], [9].

С одной стороны многослойной персептрон обладает простым математическим аппаратом, т. к. все связи в сети направлены в одном направлении от входного слоя к выходному, с другой стороны за счет увеличения скрытых слоев можно добиваться требуемой вычислительной мощности сети [15].

Многослойный персептрон состоит из слоев трех типов: входного слоя, выходного слоя и некоторого количества скрытых слоев (рисунок 7).

Входной слой сети необходим только для получения сетью набора

входных сигналов i_1, i_2, i_3 .

Выходной слой сети необходим только для формирования выходных сигналов O_1, O_2 .

Скрытые слои выполняют преобразование сигналов в соответствии с используемой функцией активации и значением весовых коэффициентов.

Существует множество алгоритмов обучения многослойного персептрона, но все они основаны на идее оптимизации параметров сети путем постепенной минимизации ошибки отклика сети за счет применения градиентного спуска [3]. Наиболее часто используемый алгоритм обучения сети - обратное распространение ошибки [4].

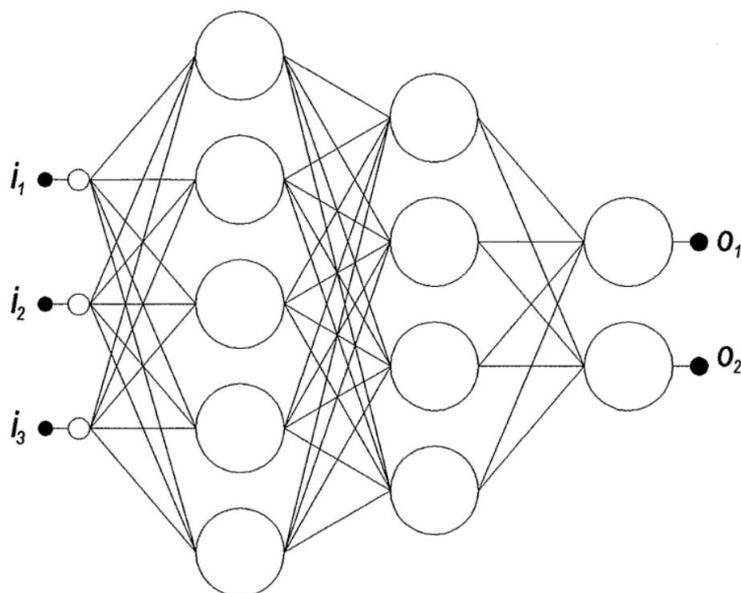


Рисунок 7 – Архитектура двухслойного персептрона с набором входных сигналов (i_1, i_2, i_3) и выходных сигналов (O_1, O_2)

Многослойный персептрон относится к сетям с прямым распространением сигналов. Однако существуют также рекуррентные сети, в которых сигналы могут многократно обрабатываться одним и тем же слоем или возвращаться на дообработку на предыдущий слой [6], [8]. Одной из рекуррентных нейронных сетей является LSTM. Архитектура сети показана на рисунке 8.

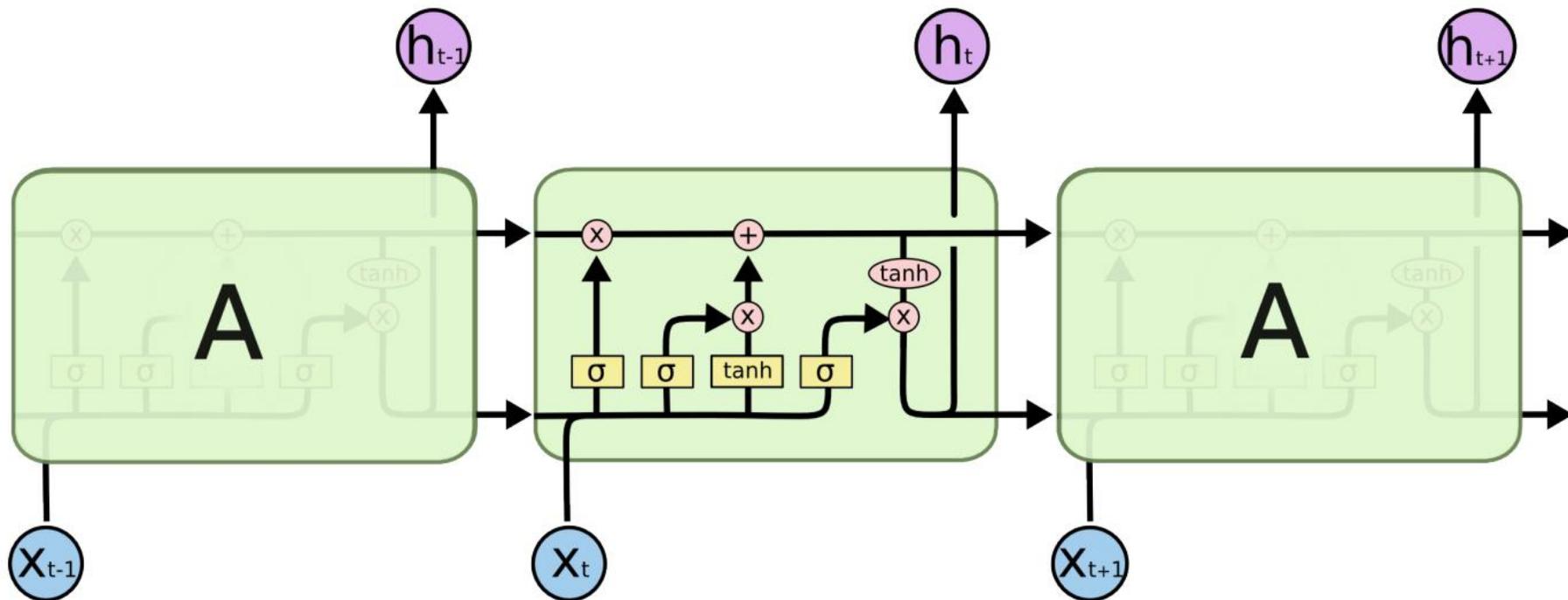


Рисунок 8 – Архитектура LSTM сети

LSTM сеть относится к глубоким нейронным сетям и часто используется для анализа и предсказания временных рядов. Глобально LSTM сеть состоит из последовательно связанных блоков. В каждом блоке содержится 4 слоя нейронов выполняющих обработку входящих в него сигналов [10], [12].

В LSTM сети состояние текущего модуля передается в следующий с помощью сигнала C_t (рисунок 9). Каждый последующий модуль при обработке сигналов выполняет их модификацию добавляя или удаляя часть информации [16].

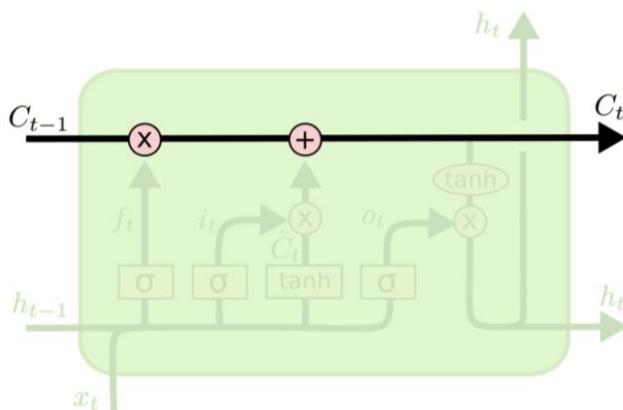


Рисунок 9 – Передача сигнала C_t в следующий блок сети

Рассмотрим основные этапы работы LSTM сети на примере одного модуля. На первом этапе сигнал обрабатывается с помощью узла forget gate (рисунок 10).

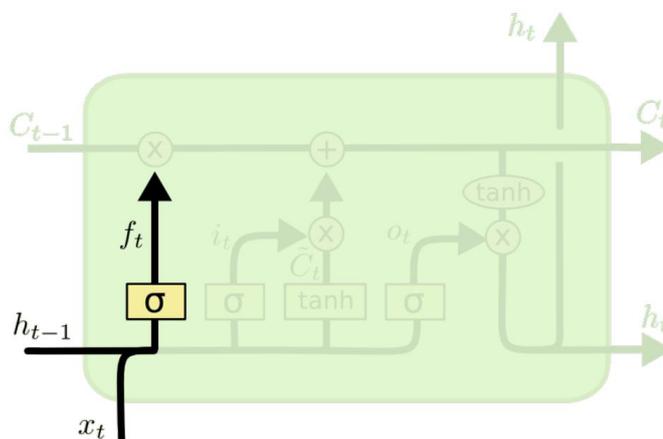


Рисунок 10 – Первый этап работы сети LSTM

Результат f_t обработки сигнала узлом forget gate рассчитывается по формуле (1).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

На следующем этапе осуществляется расчет коррекции сигнала состояние модуля по схеме, представленной на рисунке 11. Расчет осуществляется по формулам (2).

$$\begin{cases} i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \end{cases} \quad (2)$$

Рассчитанные значения будут использоваться для корректировки значения сигнала состояния модуля [11].

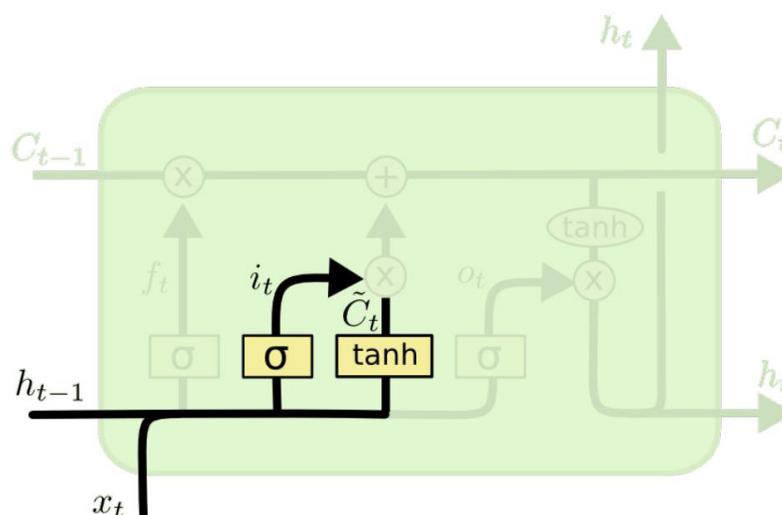


Рисунок 11 – Второй этап работы сети LSTM

Далее производится расчет C_t нового состояния модуля по схеме, представленной на рисунке 12.

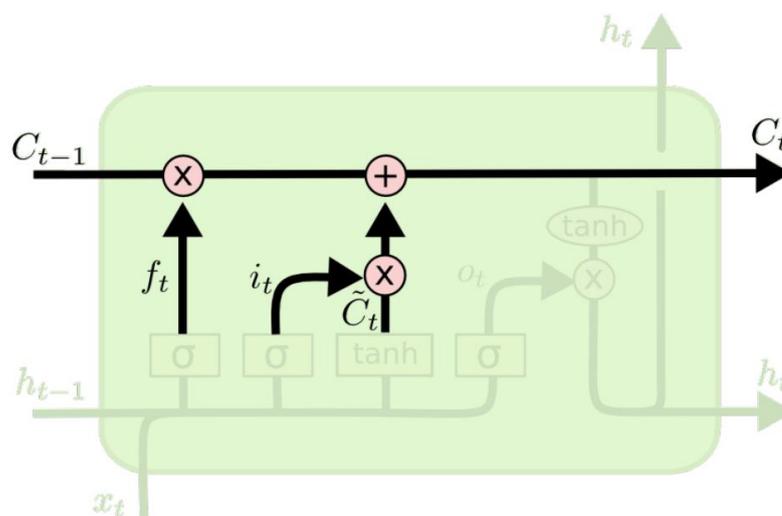


Рисунок 12 – Третий этап работы сети LSTM

Расчеты нового значения сигнала состояния осуществляются по формуле (3).

$$C_t = f_f * C_{t-1} + i_t * \tilde{C}_t \quad (3)$$

На последнем этапе осуществляется расчет сигналов узла output gate. Расчеты ведутся по схеме, показанной на рисунке 13.

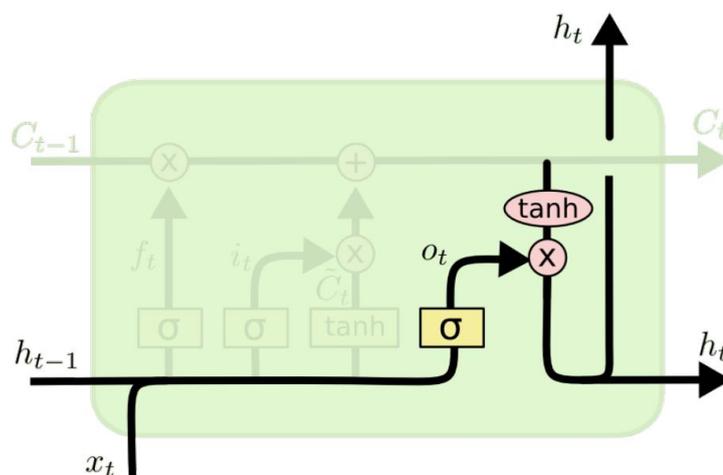


Рисунок 13 – Четвертый этап работы сети LSTM

Выполнение последнего этапа осуществляется по формулам (4).

$$\begin{cases} o_t = \sigma(W_0[h_{t-1}, x_t] + b_0) \\ h_t = o_t * \tanh(C_t) \end{cases} \quad (4)$$

Основное преимущество LSTM заключается в отсутствии проблем, связанных с остановкой обучения при малых значениях градиента. Это значительно экономит время, затрачиваемое на повторные запуски обучения сети [17], [19].

Переосмысление принципов работы рекуррентных нейронных LSTM в сторону упрощения математического аппарата привело к появлению сетей GRU. Главным изменением GRU по сравнению с LSTM является замена нескольких узлов в архитектуре на узел update gate (рисунок 14).

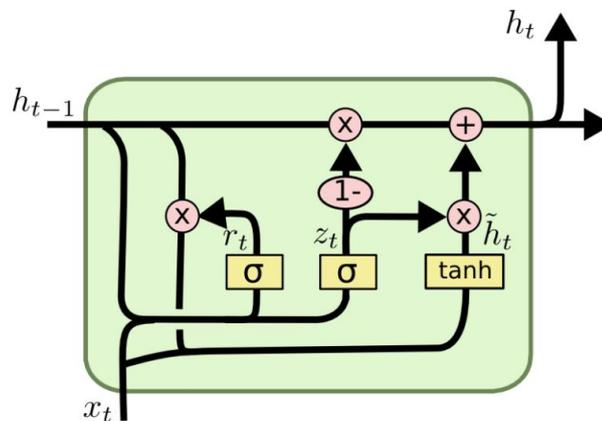


Рисунок 14 – Рекуррентная сеть GRU

Расчет сигналов GRU сети осуществляется по формулам (5).

$$\begin{cases} z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \\ r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \\ \tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t]) \\ h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \end{cases} \quad (5)$$

Выделим преимущества сетей GRU по сравнению с LSTM. Сюда можно отнести более простой математический аппарат и повышенную скорость обучения [18]. Практика использования GRU и LSTM сетей, показывает, что в некоторых задачах сети GRU могут оказаться точнее [21].

Примером пересмотра сетей LSTM в сторону усложнения архитектуры являются сети biLSTM. biLSTM – нейронная сеть, состоящая из двух LSTM сетей, работающих в противоположных направлениях в прямом и обратном. Усложнение архитектуры, хоть и замедляет процесс обучения, но в конечном итоге положительно сказывается на точности работы сети.

Дальнейшее усложнение архитектуры привело к появлению sLSTM сетей, архитектура которой показана на рисунке 15.

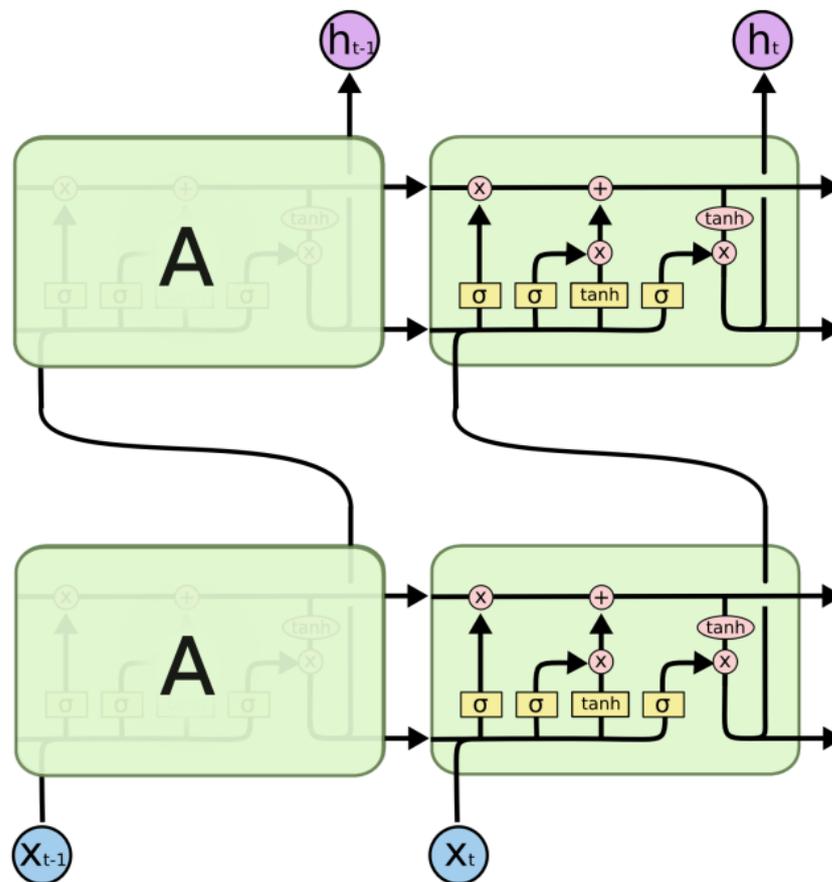


Рисунок 15 – Многослойная сеть sLSTM

sLSTM представляют из себя многослойную сеть состоящую из нескольких LSTM слоев. sLSTM относится к глубоким нейронным сетям. Благодаря возросшему количеству гиперпараметров sLSTM сети могут описывать более сложные зависимости, чем рассмотренные выше аналоги.

Таким образом, рассмотрены следующие виды нейронных сетей:

- MLP - Multi-layer perceptron;
- LSTM - Long short-term memory;
- GRU - Gated Recurrent Unit;
- biLSTM - Bidirectional Long short-term memory;
- sLSTM - LSTM-stacked.

Проверка эффективности этих нейронных сетей будет осуществляться при выполнении практической части работы.

Выводы по главе 2

В качестве выводов по главе можно отметить следующее:

- рассмотрены технические, фундаментальные и макроэкономические показатели, которые могут быть использованы при прогнозировании криптовалют, а также приведены их описание и характеристики.
- проведено сравнение нейронных сетей, подходящих для решения задачи прогнозирования криптовалют. Выбраны 4 типа нейронных сетей, которые будут использованы при создании системы прогнозирования котировок криптовалют. Среди них сети LSTM, GRU, biLSTM и sLSTM.

Теперь перейдем к рассмотрению анализа данных котировок криптовалют.

Глава 3 Анализ данных котировок криптовалют

3.1 Методика предобработка данных

В проведенном исследовании установлено, что набор данных, состоящих из признаков, причисленных в таблицах 1 и 2 обладают следующими особенностями, осложняющими анализ данных:

- диапазоны изменения значений признаков сильно различаются друг от друга, что может повлиять на результаты анализа данных;
- количество признаков слишком большое, для построения удовлетворительных по точности прогнозных моделей.

Для преодоления этих проблем предложена процедура предобработки данных, включающей в себя: нормировку данных MinMax функцией и снижение размерности признакового пространства с использованием метода SHapley Additive exPlanation.

Нормировка данных обычно используется для приведения значений признаков к одному заданному интервалу. Это позволяет обеспечить независимость значимости признаков от диапазона изменения их значений.

В нашем случае нормировка данных является необходимой, так как для анализа используются искусственные нейронные сети.

Нормировка данных осуществляется с помощью стандартной MinMax функции:

$$Xstd_i = \frac{X_i - \min(X)}{\max(X) - \min(X)}, \quad (6)$$

«где X_i – исходное i -ое значения ряда \mathbf{X} , $\min()$ – функция возвращающая минимальное значение ряда \mathbf{X} , $\max()$ – функция возвращающая максимальное значение ряда \mathbf{X} , $Xstd_i$ – нормированное i -ое значения ряда \mathbf{X} » [23].

Существует два основных метода для снижения признакового

пространства. Первый метод – это создание более компактного признакового пространства, производного от первого. Для это выбираются обоснованные подходы, объединяющие несколько признаков в один. Например, вместо того чтобы, хранить все значения временного ряда, оставляют только среднее значение. Таким образом количество признаков сокращается до приемлемого уровня. Но при этом информационная значимость данных снижается, так как используются производные признаки.

Второй метод основан на отсеивании малозначимых признаков, не оказывающих значимого влияния на результаты прогнозирования. Процедуру определения значимости признаков в настоящем исследовании предложено возложить на машинное обучение за счет использования метода SHapley Additive exPlanation.

Фильтрации подвергались технические и фундаментальные признаки, описанные в предыдущем разделе. За реализацию метода SHapley Additive explanation отвечала библиотека shar для языка программирования python.

Результаты определения значимости признаков с использованием метода SHapley Additive explanation для технических индикаторов криптовалют показаны на рисунке 16.

Результаты фундаментальных индикаторов криптовалют показаны на рисунке 17.

В соответствии с методом SHapley Additive explanation, считается чем сильнее отклоняется значение SHAP для признака от нулевого значения, тем больше его значимость.

Как видно из рисунков 16 и 17 у многих признаков среднее значение SHAP близко к нулю, поэтому эти признаки будут отброшены.

Таким образом из технических индукторов оставались:

- группа индикаторов SMA;
- группа индикаторов WMA;
- группа индикаторов MACD;
- группа индикаторов EMA.

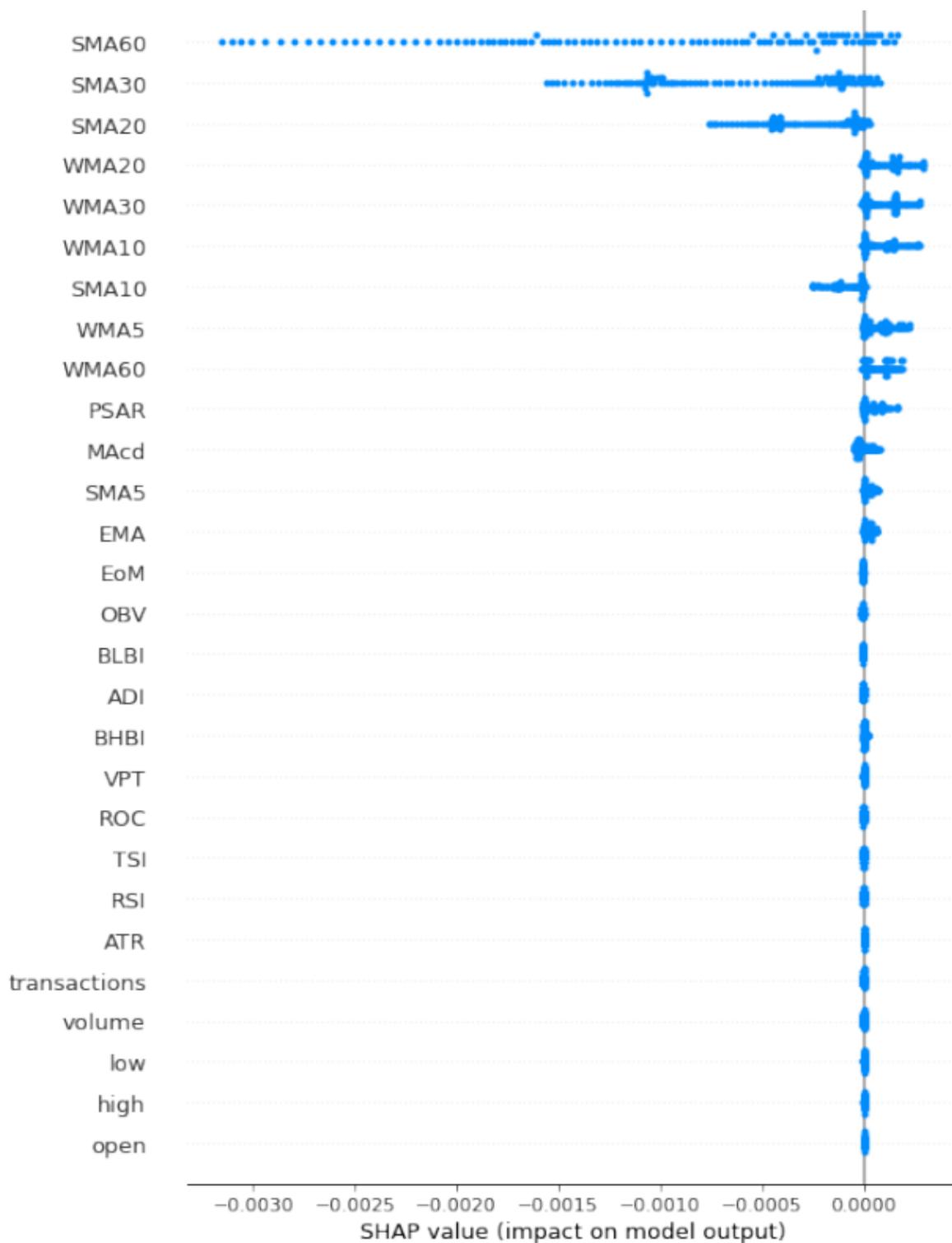


Рисунок 16 – Результаты определения значимости технических индикаторов криптовалют с использованием метода SHapley Additive explanation

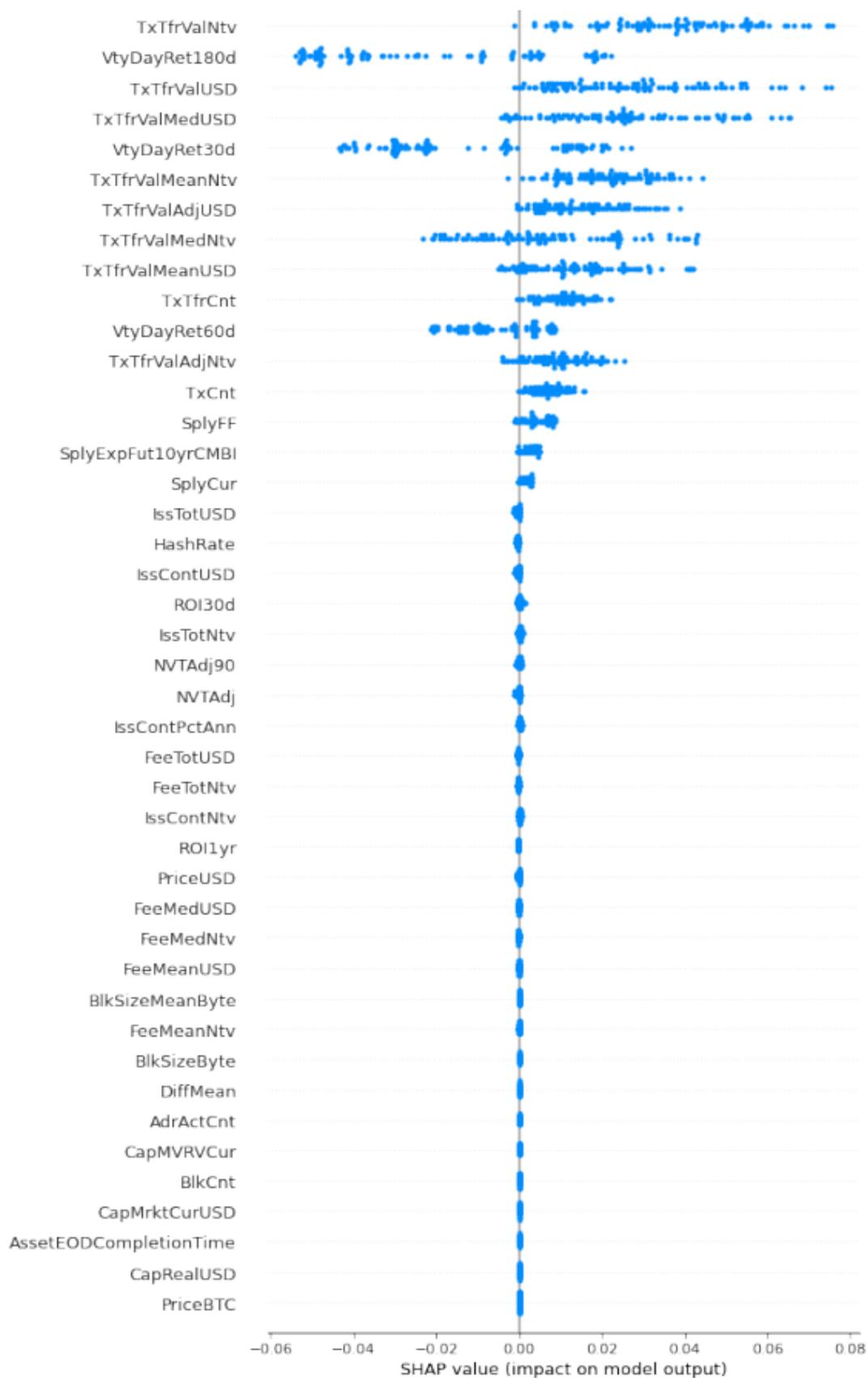


Рисунок 17 – Результаты определения значимости фундаментальных показателей криптовалют

Среди фундаментальных индикатор остаются:

- «TxTfrvalNtv;
- VtyDayRet180d;
- TxTfrValUSD;
- TxTfrValMedUSD;
- VtyDayRet30d;
- TxTfrValMeanNtv;
- TxTfrValAdjUSD;
- TxTfrValMedNtv;
- TxTfrValMeanUSD;
- TxTfrCnt;
- VtyDayRet60d;
- TxTfrValAdjNtv;
- TxCnt;
- SplyFF;
- SplyExpFut10yrCMBI;
- SplyCur» [13].

Теперь необходимо определить, как повлияло отсеиванием 56% признаков на точность прогнозирования.

3.2 Апробация предложенной методики предобработка данных

Для проведения вычислительных экспериментов были выбраны котировки двух криптовалют: Dash и Bitcoin. В ходе вычислительных экспериментов проверялось на сколько изменилась ошибка прогнозирования RMSE при снижении размерности признакового пространства. Вычислительные эксперименты проводились отдельно для технических индикаторов и фундаментальных показателей. Вычислительные

эксперименты повторялись при использовании разных типов нейронных сетей (LSTM, GRU, biLSTM и sLSTM).

Результаты вычислительных экспериментов для технических индикаторов с результатами определения ошибок RMSE показаны в таблицах 3 и 4.

Таблица 3 – Влияние отсеивания технических индикаторов на ошибку RMSE прогнозирования котировок Dash

Нейронные сети	Признаки участвующие в обучении	
	все 23 технических индикатора	13 индикаторов, выбранных по SHAP value
LSTM	0,059	0,059
GRU	0,058	0,059
biLSTM	0,057	0,059
sLSTM	0,059	0,066

Таблица 4 – Влияние отсеивания технических индикаторов на ошибку RMSE прогнозирования котировок Bitcoin

Нейронные сети	Признаки участвующие в обучении	
	все 23 технических индикатора	13 индикаторов, выбранных по SHAP value
LSTM	0,053	0,055
GRU	0,052	0,055
biLSTM	0,054	0,054
sLSTM	0,051	0,058

Результаты вычислительных экспериментов для фундаментальных показателей с результатами определения ошибок RMSE показаны в таблицах 5 и 6.

Как видно из таблиц 3 и 4 снижение размерности признакового пространства привело к увеличению ошибки RMSE прогнозирования котировок, но незначительную величину (значение ошибки увеличилось в

среднем на 4%)

Таблица 5 – Влияние отсеивания фундаментальных показателей на ошибку RMSE прогнозирования котировок Dash

Нейронные сети	Признаки участвующие в обучении	
	все 43 фундаментальных показателя	16 показателей, выбранных по SHAP value
LSTM	0,058	0,059
GRU	0,057	0,059
biLSTM	0,058	0,059
sLSTM	0,053	0,059

Таблица 6 – Влияние отсеивания фундаментальных показателей на ошибку RMSE прогнозирования котировок Bitcoin

Нейронные сети	Признаки участвующие в обучении	
	все 43 фундаментальных показателя	16 показателей, выбранных по SHAP value
LSTM	0,046	0,046
GRU	0,045	0,046
biLSTM	0,046	0,046
sLSTM	0,042	0,046

Как видно из таблиц 5 и 6 снижение размерности признакового пространства в этом случае также привело к увеличению ошибки RMSE прогнозирования котировок - значение ошибки увеличилось в среднем на 2,5%. Также можно сделать промежуточный вывод, что сокращение технических индикаторов повлияло более негативно, чем фильтрация фундаментальных показателей.

Для дополнительного подтверждения полученных результатов повторим вычислительные эксперименты на котировках криптовалют 5 криптовалют: btc, xrp, dash, xmr и ltc с использованием только LSTM нейронной сети. Полученные результаты сведем в таблицы 7 и 8.

Таблица 7 – Влияние отсеивания фундаментальных показателей на ошибку RMSE прогнозирования с использованием LSTM нейронной сети

Криптовалюта	Признаки участвующие в обучении	
	все 23 технических индикатора	13 индикаторов, выбранных по SHAP value
btc	0,053	0,054
xrp	0,043	0,043
dash	0,059	0,059
xmr	0,051	0,052
ltc	0,018	0,019

Таблица 8 – Влияние отсеивания фундаментальных показателей на ошибку RMSE прогнозирования с использованием LSTM нейронной сети

Криптовалюта	Признаки участвующие в обучении	
	все 43 фундаментальных показателя	16 показателей, выбранных по SHAP value
btc	0,046	0,046
xrp	0,041	0,044
dash	0,058	0,059
xmr	0,054	0,055
ltc	0,019	0,21

Полученные результаты позволяют сделать следующий вывод. Разработанная методика предобработки данных котировок криптовалют в сочетании с нейронной сетью LSTM позволила снизить количество используемых для прогнозирования фундаментальных показателей и технических индикаторов суммарно на 56% без существенного увеличения ошибки RMSE

3.3 Сравнение эффективности различных типов нейронных сетей

Теперь, когда, была экспериментально доказана эффективность использования предложенного подхода по снижению размерности

признакового пространства, необходимо определить наиболее эффективную нейронную сеть для анализируемых в магистерской диссертации данных.

В данном разделе проводится сравнение эффективности различных архитектур нейронных сетей, используемых для прогнозирования котировок криптовалют. Сравнению подвергались следующие типы нейронных сетей: многослойный персептрон (MLP), LSTM, GRU, biLSTM и sLSTM. Эти архитектуры были выбраны в силу их распространенности и высокой эффективности в задачах обработки временных рядов.

Опишем методику сравнения эффективности различных типов нейронных сетей. Для оценки эффективности нейронных сетей использовались данные о котировках двух криптовалют: Bitcoin и Dash.

Технические индикаторы и фундаментальные показатели, описанные ранее, служили входными признаками. Основные этапы эксперимента включали:

- предобработку данных, включающую нормализацию с использованием MinMaxScaler и снижение размерности пространства признаков с помощью SHAP;
- обучение каждой модели на обучающем наборе данных;
- оценку моделей на тестовом наборе данных с использованием метрики RMSE (Root Mean Squared Error).

Каждая модель обучалась с одинаковыми гиперпараметрами (количество нейронов, функция активации, количество эпох, размер мини-батча), что обеспечивало справедливость сравнения.

Для каждой модели проводилось по пять независимых запусков, чтобы минимизировать влияние случайных факторов.

Результаты вычислительных экспериментов для моделей на основе технических индикаторов представлены в таблице 9, а для моделей на основе фундаментальных показателей – в таблице 10.

Таблица 9 – Результаты сравнения ошибок RMSE при использовании различных типов нейронных сетей при прогнозировании на основе 13 технических индикаторов (выбранных по SHAP value)

Модель	Bitcoin (RMSE)	Dash (RMSE)
MLP	0,062	0,068
LSTM	0,053	0,059
GRU	0,052	0,058
biLSTM	0,054	0,057
sLSTM	0,051	0,056

Таблица 10 – Результаты сравнения ошибок RMSE при использовании различных типов нейронных сетей при прогнозировании на основе 16 фундаментальных показателей (выбранных по SHAP value)

Модель	Bitcoin (RMSE)	Dash (RMSE)
MLP	0,057	0,063
LSTM	0,046	0,058
GRU	0,045	0,057
biLSTM	0,046	0,058
sLSTM	0,042	0,053

Анализ полученных результатов показывает, что:

- MLP продемонстрировал худшую точность среди всех моделей. Это объясняется ограниченной способностью многослойного персептрона захватывать временные зависимости в данных;
- LSTM и GRU показали близкие результаты, однако GRU имеет незначительное преимущество в скорости обучения и точности, что связано с более простой архитектурой;
- biLSTM увеличивает точность за счет двунаправленного анализа временных рядов, однако выигрыш по сравнению с обычным LSTM незначителен;
- sLSTM показал наилучшую точность за счет использования

глубокой архитектуры с несколькими слоями LSTM, что позволяет модели захватывать более сложные зависимости.

При использовании технических индикаторов модели GRU и sLSTM показали минимальную ошибку RMSE.

При использовании фундаментальных показателей sLSTM вновь продемонстрировал наилучшие результаты, что подтверждает его способность извлекать сложные зависимости между признаками.

Выводы по главе 3

На основании проведенного анализа можно сделать следующие выводы:

- разработана методика предобработки данных котировок криптовалют, включающая в себя нормировку данных и фильтрацию признаков на основе определения их значимости по значению SHAP;
- разработанная методика предобработки данных котировок криптовалют в сочетании с нейронной сетью LSTM позволила снизить количество используемых для прогнозирования фундаментальных показателей и технических индикаторов суммарно на 56% без существенного увеличения ошибки RMSE;
- для задач прогнозирования котировок криптовалют архитектуры рекуррентных нейронных сетей, такие как GRU и LSTM, превосходят многослойный персептрон;
- использование более сложных архитектур, таких как biLSTM и sLSTM, позволяет повысить точность прогнозирования, особенно при наличии большого количества данных;
- каждая модель обучалась с одинаковыми гиперпараметрами (количество нейронов, функция активации, количество эпох, размер мини-батча), что обеспечивало справедливость сравнения;
- для каждой модели проводилось по пять независимых запусков, чтобы минимизировать влияние случайных факторов;

- наиболее оптимальной моделью с точки зрения точности и сложности реализации является sLSTM, которая обеспечивает минимальную ошибку RMSE при прогнозировании на основе как технических индикаторов, так и фундаментальных показателей;
- при использовании фундаментальных показателей sLSTM вновь продемонстрировал наилучшие результаты, что подтверждает его способность извлекать сложные зависимости между признаками;
- рекомендуется использовать sLSTM в сочетании с методами предобработки данных, такими как нормализация и снижение размерности, для построения высокоточных систем прогнозирования криптовалютных рынков.

Теперь можно переходить к рассмотрению программной реализации предложенных в работе исследований и описанию программного модуля сбора данных, программного модуля анализа данных и программного модуля визуализации данных.

Глава 4 Программная реализация анализа данных

4.1 Программный модуль сбора данных

Для реализации и проверки предложенных в магистерской диссертации исследовательских решений на языке программирования Python разработано программное решение, которое состоит из следующих модулей:

- модуль сбора финансовых данных;
- модуль анализа данных;
- модуль визуализации.

Рассмотрим работу модуля для сбора финансовых данных.

Данный модуль предназначен для сбора котировок криптовалют, включая исторические данные, и сохранение их в файлы csv и xlsx для последующего анализа.

Выбор источника для получения котировок криптовалют продиктован следующими требованиями – бесплатностью доступа, поддержкой наиболее высоко капитализированных токенов, наличие API. Единственный источник финансовых данных, который удовлетворяет описанным требованиям – yahoo!finance (рисунок 18). Стоит отдельно отметить, что данный сервис постепенно закрывает свой API и через некоторое время планируется отключение доступа к API для внешних пользователей.

Взаимодействие с API сервиса осуществляется с использованием библиотеки с открытым исходным кодом – ufinance. Подключение библиотеки к проекту осуществляется с использованием инструкции import. Для доступа к требуемым котировкам необходимо указать шифр токена в сервисе yahoo!finance (например, шифр Bitcoin – «BTC-USD»). Параметр period со значением max указывает на то, что будут загружены все доступные исторические данные для указанного токена.

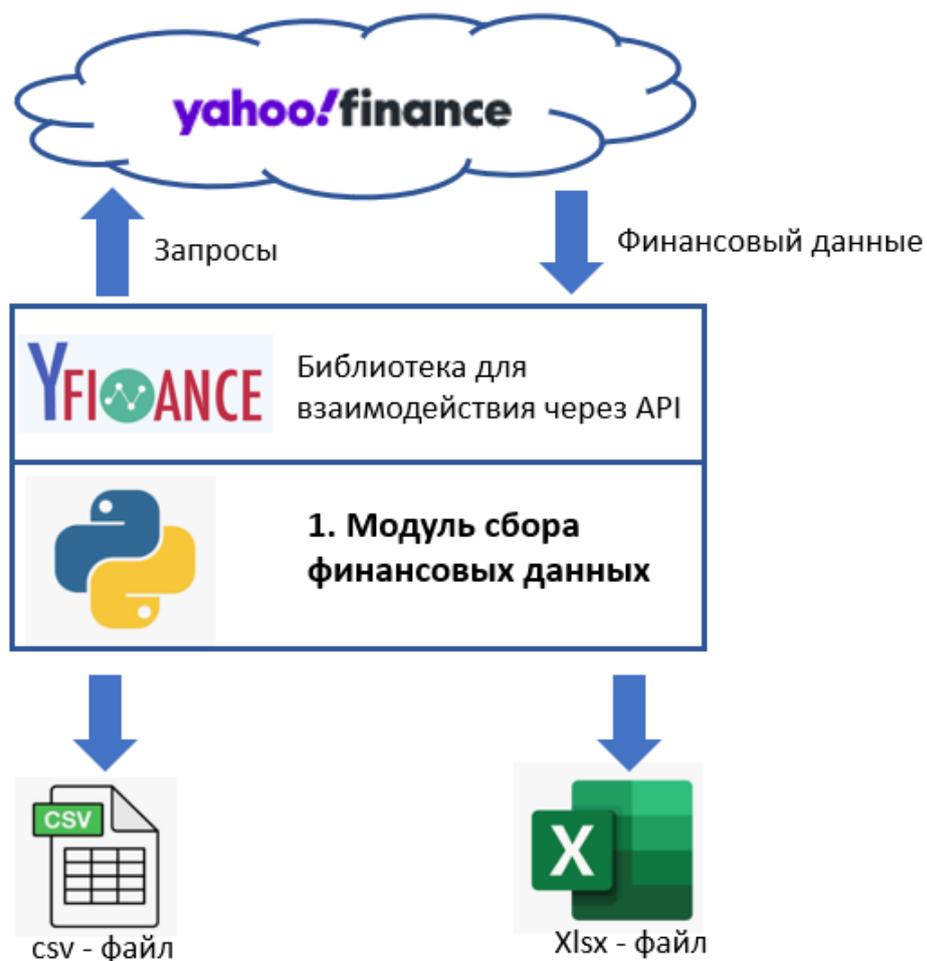


Рисунок 18 – Схема работы модуля для сбора финансовых данных

Все загруженные данные с помощью функции `to_csv` сохраняются в текстовом виде в файл с указанным именем. Пример программного кода для загрузки и сохранения котировок Bitcoin показан на рисунке 19.

```
[ ] import yfinance as yf
w_data = yf.download('BTC-USD', period='max')
w_data.to_csv("w-data-max.csv")
```

⇒ [*****100%*****] 1 of 1 completed

Рисунок 19 – Пример программного кода для загрузки данных с сервиса yahoo!finance

Аналогичный программный код выполняется и для других токенов, анализируемых в исследовании, а именно: xrp, dash, xmr и ltc. Также в коде предусмотрен механизм обработки исключений для случаев, когда котировки криптовалют не удастся загрузить.

4.2 Программный модуль для анализа данных

Второй программный модуль предназначен для реализации предложенных в диссертационной работе подходов по анализу криптовалютных котировок (рисунок 20), и он включает в себя:

- моделирование работы нейронных сетей (типов LSTM и GRU),
- «предобработки данных (нормализация с помощью функции MinMaxScaler);
- снижение размерности признакового пространства с использованием метода SHapley Additive explanation;
- обучения нейронных сетей;
- выполнения прогнозирования котировок.

Работа программного модуля основана использовании следующих библиотек» [27]:

- «библиотека NumPy, которая предназначена для работы с массивами и матрицами данных. Использование: Быстрые математические операции с данными, включая вычисления для обработки временных рядов, которые характерны для анализа котировок криптовалют» [25];
- библиотека Pandas, которая предназначена для обработки и анализ данных в табличном формате. Использование: чтение и манипуляция данными (например, загрузка исторических цен криптовалют из файлов CSV); подготовка данных, преобразование временных рядов и извлечение определенных признаков [28];

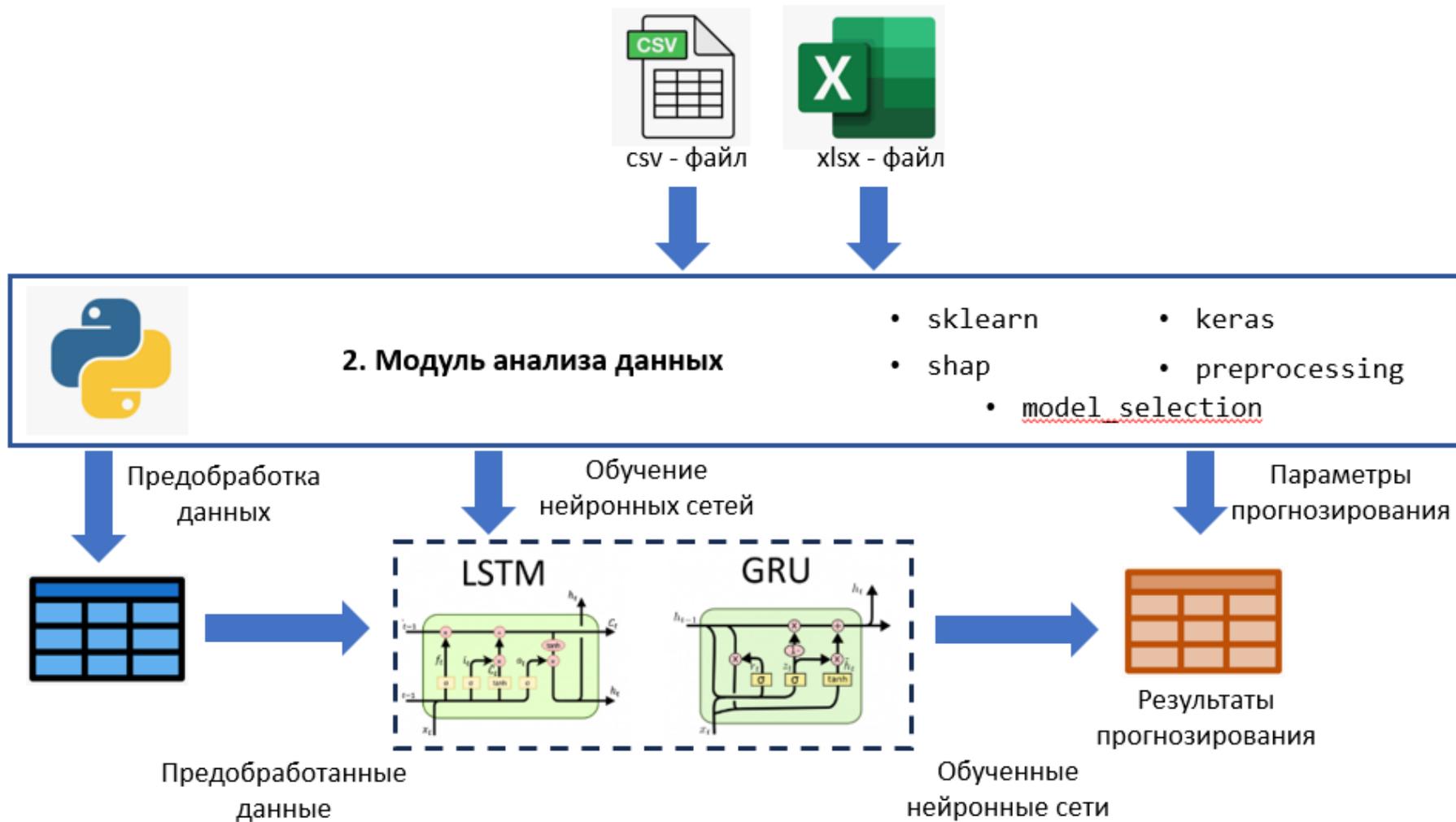


Рисунок 20 – Схема работы модуля для анализа данных

- «библиотека `sklearn`, раздел `preprocessing.MinMaxScaler`, который используется для нормализации данных к определенному диапазону (обычно $[0, 1]$). Использование: масштабирование данных для улучшения сходимости моделей машинного обучения, нормализация цен криптовалют для моделей нейронных сетей;
- библиотека `sklearn`, раздел `model_selection.train_test_split`, который используется для разделения данных на обучающую и тестовую выборки» [25]. Использование: позволяет проводить обучение и тестирование моделей на различных частях данных, чтобы проверить их точность и избежать переобучения;
- библиотека `shap`, которая предназначена для интерпретации и объяснения моделей машинного обучения. Использование: визуализация важности признаков для понимания, как каждый параметр влияет на предсказания модели (выявление факторов, влияющих на цену криптовалют);
- библиотека `tensorflow.keras`, которая предназначена для библиотеки `pandas`, которая предназначена для обработки и анализа данных в табличном формате и используется при построения и обучение нейронных сетей.

Из библиотеки в программном модуле используются следующие компоненты:

- `Sequential` для упрощенного способа создания последовательной модели нейронной сети;
- `LSTM (Long Short-Term Memory)` для реализации рекуррентных сетей, предназначенный для работы с временными рядами;
- `GRU (Gated Recurrent Unit)` для реализации альтернативного типа рекуррентных сетей с более простой архитектурой, чем `LSTM`;
- `Dense` для задания полносвязных слоев в нейронных сетях;
- `Dropout` для управления процессом регуляризации предотвращения переобучения, исключая случайные нейроны из обучения.

Подключение библиотек, необходимых для выполнения предобработки данных и моделирования нейронных сетей показано на рисунке 21.

```
[ ] import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
import shap
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, GRU, Dense, Dropout
```

Рисунок 21 – Подключение библиотек для моделирования нейронных сетей

После инициализации библиотек разработанный программный модуль последовательно выполняет следующие этапы анализа: загрузка ранее собранных данных с котировками криптовалют из файла, предобработка данных, снижение размерности признакового пространства методом SHAP, обучение на полученных данных нейронных сетей, проверка их точности работы.

Для загрузки данных разработана функция `load_data`, которая с использованием метода `read_csv` считывает данные из текстового файла формата csv и конвертирует их в удобный для анализа тип данных – `DataFrame` (рисунок 21).

```
[ ] def load_data(file_path):
    # Загружаем данные из CSV (или другого формата)
    data = pd.read_csv(file_path)
    return data
```

Рисунок 21 – Импорт данных из файла

Для предобработки данных разработана функция `preprocess_data`, которая на вход получает анализируемые данные – `data` и `target_column`. После загрузки данных, они нормируются к диапазону `[0; 1]`. Для этого используется экземпляр класса `MinMaxScaler` из компонента `preprocessing`. Нормировка выполняется с помощью вызова функции `scaler.fit_transform`, а нормированные данные сохраняются в переменной `data_scaled` (рисунок 22). Нормировка данных – обязательный шаг при работе с рекуррентными нейронными сетями типа GRU и LSTM.

```
[ ] def preprocess_data(data, target_column):  
    # Нормализация данных  
    scaler = MinMaxScaler(feature_range=(0, 1))  
    data_scaled = scaler.fit_transform(data)
```

Рисунок 22 – Нормировка данных в функции `preprocess_data`

Для выделения целевого значения используется операция среза (рисунок 23), а разделение на обучающую и тестовую выборки осуществляется с помощью функции `train_test_split` (рисунок 24).

```
[ ] X = data_scaled[:, :-1]  
    y = data_scaled[:, target_column]
```

Рисунок 23 – Выделение целевого значения в функции `preprocess_data`

```
[ ] X_train, X_test,  
    y_train, y_test = train_test_split(X, y, test_size=0.2,  
                                       random_state=42)  
  
    return X_train, X_test, y_train, y_test, scaler
```

Рисунок 24 – Разделение на обучающую и тестовую выборки в функции preprocess_data

Следующим шагом выполняется снижение размерности данных для обучения нейронной сети, основываясь на значениях важности признаков, вычисленных с помощью библиотеки SHAP (SHapley Additive exPlanations).

Снижение размерности признакового пространства помогает сократить размерность данных, используемых при обучении нейронных сетей. Это приводит к снижению вычислительных затрат, ускорению процесса обучения и может улучшить качество прогнозных моделей за счет удаления малозначительных признаков.

Входными параметрами функции reduce_dimensionality, отвечающей за выполнение данного шага являются: матрицы признаков обучающей и тестовой выборок (X_train, X_test), целевые метки обучающей выборки (y_train) и модель машинного обучения (model).

Следующим шагом создаётся «объяснитель» (Explainer) на основе переданной модели и обучающих данных. SHAP вычисляет вклад каждого признака в прогноз модели. Значения SHAP вычисляются для всех признаков исходной выборки данных.

Важность каждого признака рассчитывается как среднее от абсолютных значений SHAP по всем записям выборки данных. Также в коде устанавливается порог важности как значение верхнего квартиля, что позволяет отобрать только те признаки, которые входят в 25% наиболее значимых.

Обучающая и тестовая выборки уменьшаются, оставляя только отобранные важные признаки. Результат такого отбора помещается в переменные `X_train_reduced` и `X_test_reduced`.

Функция `reduce_dimensionality` возвращает уменьшенные по размерности версии обучающей и тестовой выборок.

Программный код функции `reduce_dimensionality` для снижения размерности данных показан на рисунке 25.

```
def reduce_dimensionality(X_train, X_test, y_train, model):
    explainer = shap.Explainer(model, X_train)
    shap_values = explainer(X_train)

    # Определение важности признаков
    feature_importance = np.abs(shap_values.values).mean(axis=0)

    # Оставляем только важные признаки
    threshold = np.percentile(feature_importance, 75) # Верхний квартиль
    important_features = np.where(feature_importance >= threshold)[0]

    X_train_reduced = X_train[:, important_features]
    X_test_reduced = X_test[:, important_features]

    return X_train_reduced, X_test_reduced
```

Рисунок 25 – Снижение размерности данных с помощью SHAP

Следующим шагом осуществляется конфигурирование предсказательных моделей. Для этого используется программный код, который включает в себя две функции: `create_lstm_model` и `create_gru_model`, каждая из которых создаёт и возвращает нейронную сеть для обработки временных рядов с использованием соответствующей архитектуры рекуррентных нейронных сетей.

Функция `create_lstm_model` предназначена для построения модели на основе Long Short-Term Memory (LSTM) сети. Модель состоит из двух слоёв

LSTM с числом нейронов, равным 50. В каждом слое используется функция активации \tanh , которая обеспечивает нелинейность и позволяет сети обрабатывать сложные зависимости во временных данных. Первый слой LSTM настроен на возвращение последовательностей (`return_sequences=True`), что позволяет передавать данные следующему слою LSTM. Между слоями применяются слои Dropout с коэффициентом 0.2, которые предотвращают переобучение за счёт случайного отключения части нейронов. После рекуррентных слоёв добавлен плотный слой Dense с одним нейроном, предназначенный для генерации итогового предсказания. Модель компилируется с использованием оптимизатора Adam, который адаптивно настраивает скорость обучения, и функции потерь MSE (среднеквадратичная ошибка), подходящей для регрессионных задач (рисунок 26).

```
def create_lstm_model(input_shape):
    model = Sequential([
        LSTM(50, activation='tanh', return_sequences=True, input_shape=input_shape),
        Dropout(0.2),
        LSTM(50, activation='tanh'),
        Dropout(0.2),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mse')
    return model

def create_gru_model(input_shape):
    model = Sequential([
        GRU(50, activation='tanh', return_sequences=True, input_shape=input_shape),
        Dropout(0.2),
        GRU(50, activation='tanh'),
        Dropout(0.2),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mse')
    return model
```

Рисунок 26 – Конфигурирование сетей LSTM и GRU

Функция `create_gru_model` аналогична предыдущей, но вместо LSTM-слоёв используются слои Gated Recurrent Unit (GRU). GRU является более простой и быстрой альтернативой LSTM, сохраняя при этом возможность учитывать временные зависимости.

Архитектура модели также включает два слоя GRU с функцией активации `tanh` и настройкой `return_sequences=True` для первого слоя. Между слоями GRU добавлены слои Dropout с теми же параметрами. Заключительный плотный слой и настройки компиляции идентичны модели LSTM.

Обе функции возвращают последовательные модели, готовые к обучению на данных временных рядов. Различие между ними заключается в типе рекуррентной сети, используемой для обработки входных последовательностей.

Для обучения нейронных сетей используется функция `train_and_evaluate`. «Функция принимает параметры, включающие модель, обучающие и тестовые выборки, а также дополнительные настройки количества эпох и размера порций данных.

В процессе работы сначала происходит обучение модели с использованием метода `fit`, где данные `X_train` и `y_train` используются для подбора параметров модели» [23].

Процесс обучения осуществляется в течение заданного числа эпох, каждая из которых разбивается на порции данных (`batch_size`).

Также для контроля качества обучения предоставляются тестовые данные `X_test` и `y_test` в качестве валидационной выборки, что позволяет модели оценивать свои прогнозы на независимых данных.

После завершения обучения функция выполняет оценку модели на тестовых данных с помощью метода `evaluate`, вычисляя итоговое значение функции потерь.

Для удобства пользователя результат этой оценки выводится в консоль с указанием значения ошибки модели.

Функция возвращает обученную модель, что позволяет использовать её для дальнейшего прогнозирования котировок криптовалют.

```
def train_and_evaluate(model, X_train, y_train, X_test, y_test,
                       epochs=50, batch_size=32):
    model.fit(X_train, y_train, epochs=epochs, batch_size=batch_size,
              validation_data=(X_test, y_test), verbose=2)
    loss = model.evaluate(X_test, y_test, verbose=0)
    print(f"Model Loss: {loss}")
    return model
```

Рисунок 27 – Функция для обучение нейронных сетей

Основной блок выполнения, который связывает все разработанные функции в единый рабочий процесс.

Сначала задаётся путь к файлу, содержащему данные о котировках криптовалют – `crypto_prices.csv`.

Эти данные загружаются с помощью функции `load_data`, после чего определяется целевой столбец, содержащий значения для прогнозирования – `PriceUSD` (рисунок 28).

```

if __name__ == "__main__":
    # Путь к данным
    file_path = "crypto_prices.csv"

    # Загрузка данных
    data = load_data(file_path)

    # Название целевого столбца ('PriceUSD')
    target_column = data.columns.get_loc('PriceUSD')

    # Предобработка данных
    X_train, X_test, y_train, y_test, scaler = preprocess_data(data,
                                                                target_column)

    # Снижение размерности
    dummy_model = Sequential([Dense(1, input_dim=X_train.shape[1])])
    X_train_reduced, X_test_reduced = reduce_dimensionality(X_train, X_test,
                                                            y_train, dummy_model)

    # Создание и обучение модели LSTM
    lstm_model = create_lstm_model((X_train_reduced.shape[1], 1))
    train_and_evaluate(lstm_model, X_train_reduced, y_train, X_test_reduced,
                       y_test)

    # Создание и обучение модели GRU
    gru_model = create_gru_model((X_train_reduced.shape[1], 1))
    train_and_evaluate(gru_model, X_train_reduced, y_train, X_test_reduced,
                       y_test)

```

Рисунок 28 – Основной блок программного кода

Далее осуществляется предобработка данных, включающая разделение на обучающую и тестовую выборки, а также их нормализацию, что позволяет привести данные к масштабам, подходящим для обучения нейронных сетей. Затем выполняется снижение размерности данных. Для этого создаётся временная модель, которая используется в функции `reduce_dimensionality` для вычисления важности признаков с помощью SHAP и отбора наиболее значимых из них. Результатом являются уменьшенные по размерности выборки для обучения и тестирования.

После этого создаются и обучаются две разные модели. Первая из них

основана на архитектуре LSTM, подходящей для анализа временных рядов. Она строится с использованием функции `create_lstm_model`, а затем обучается на подготовленных данных с помощью функции `train_and_evaluate`. Аналогичным образом создаётся и обучается модель на основе архитектуры GRU, которая также предназначена для обработки последовательностей данных, но отличается более простой структурой. Обе модели проходят оценку на тестовой выборке, что позволяет проверить их производительность и точность.

Так как модуль анализа данных является наиболее сложным и важным в рамках данного диссертационного исследования опишем его дополнительно с помощью UML диаграмм.

Сначала приведем UML диаграмму классов модуля анализа данных (рисунок 29).

Описание классов представлено ниже:

- `DataLoader` – класс для загрузки данных;
- `DataPreprocessor` – класс для предобработки данных (включая нормализацию и SHAP);
- `ModelBuilder` – класс для создания моделей LSTM и GRU;
- `ModelTrainer` – класс для обучения и оценки моделей;
- `Main` – главный процесс, связывающий остальные классы.

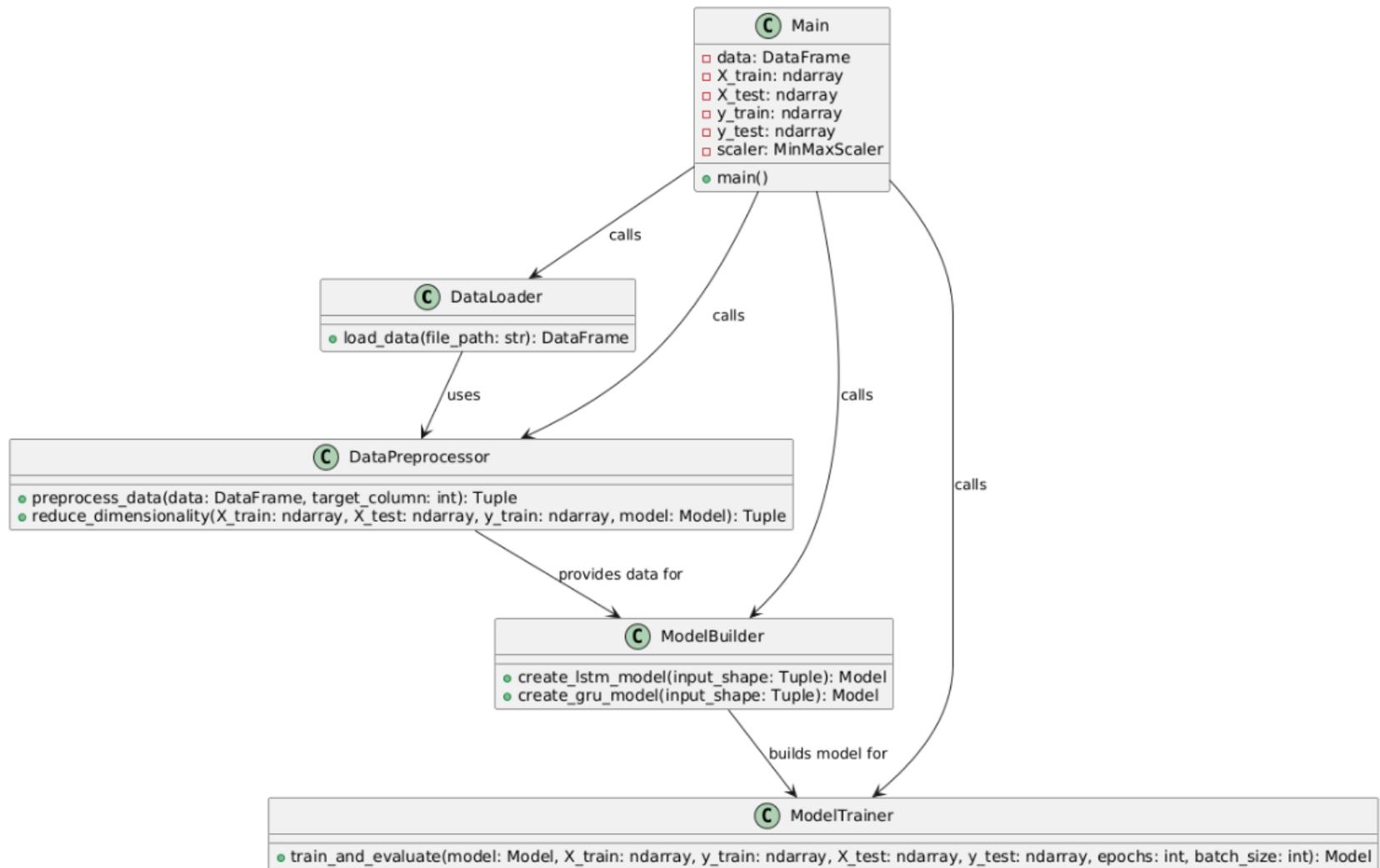


Рисунок 29 – UML диаграмма классов

Код для визуализации UML диаграммы классов модуля анализа данных показан ниже.

```
@startuml
class DataLoader {
    + load_data(file_path: str): DataFrame
}
class DataPreprocessor {
    + preprocess_data(data: DataFrame, target_column: int): Tuple
    + reduce_dimensionality(X_train: ndarray, X_test: ndarray, y_train: ndarray,
model: Model): Tuple
}
class ModelBuilder {
    + create_lstm_model(input_shape: Tuple): Model
    + create_gru_model(input_shape: Tuple): Model
}
class ModelTrainer {
    + train_and_evaluate(model: Model, X_train: ndarray, y_train: ndarray, X_test:
ndarray, y_test: ndarray, epochs: int, batch_size: int): Model
}
class Main {
    - data: DataFrame
    - X_train: ndarray
    - X_test: ndarray
    - y_train: ndarray
    - y_test: ndarray
    - scaler: MinMaxScaler
    + main()
}
```

DataLoader --> DataPreprocessor : uses

DataPreprocessor --> ModelBuilder : provides data for

ModelBuilder --> ModelTrainer : builds model for

Main --> DataLoader : calls

Main --> DataPreprocessor : calls

Main --> ModelBuilder : calls

Main --> ModelTrainer : calls

@enduml

UML диаграмма вариантов использования (use case) модуля анализа данных показана на рисунке 30.

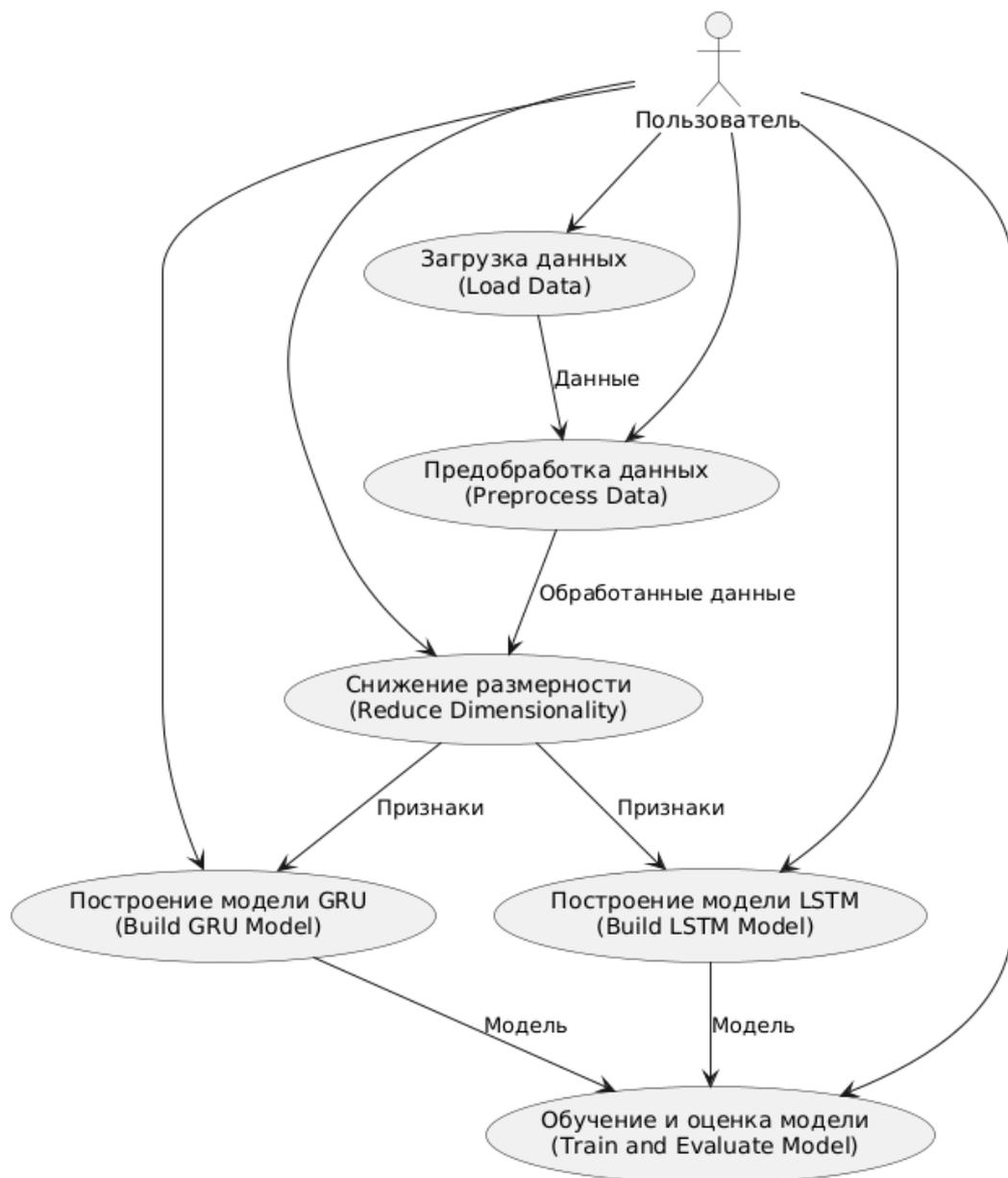


Рисунок 30 – UML диаграмма вариантов использования (use case)

Код для визуализации UML диаграммы вариантов использования модуля анализа данных.

```
@startuml
```

```
actor Пользователь
```

```
usecase "Загрузка данных\n(Load Data)" as UC1
```

```
usecase "Предобработка данных\n(Preprocess Data)" as UC2
```

```
usecase "Снижение размерности\n(Reduce Dimensionality)" as UC3
```

```
usecase "Построение модели LSTM\n(Build LSTM Model)" as UC4
```

```
usecase "Построение модели GRU\n(Build GRU Model)" as UC5
```

```
usecase "Обучение и оценка модели\n(Train and Evaluate Model)" as UC6
```

```
Пользователь --> UC1
```

```
Пользователь --> UC2
```

```
Пользователь --> UC3
```

```
Пользователь --> UC4
```

```
Пользователь --> UC5
```

```
Пользователь --> UC6
```

```
UC1 --> UC2 : Данные
```

```
UC2 --> UC3 : Обработанные данные
```

```
UC3 --> UC4 : Признаки
```

```
UC3 --> UC5 : Признаки
```

```
UC4 --> UC6 : Модель
```

```
UC5 --> UC6 : Модель
```

```
@enduml
```

Теперь перейдем к описанию модуля визуализации данных.

4.3 Программный модуль для визуализации данных

Для того, чтобы продемонстрировать работу прогнозных моделей, обученных на основе подходов, предложенных в диссертационном исследовании, был разработан модуль визуализации данных (рисунок 31).

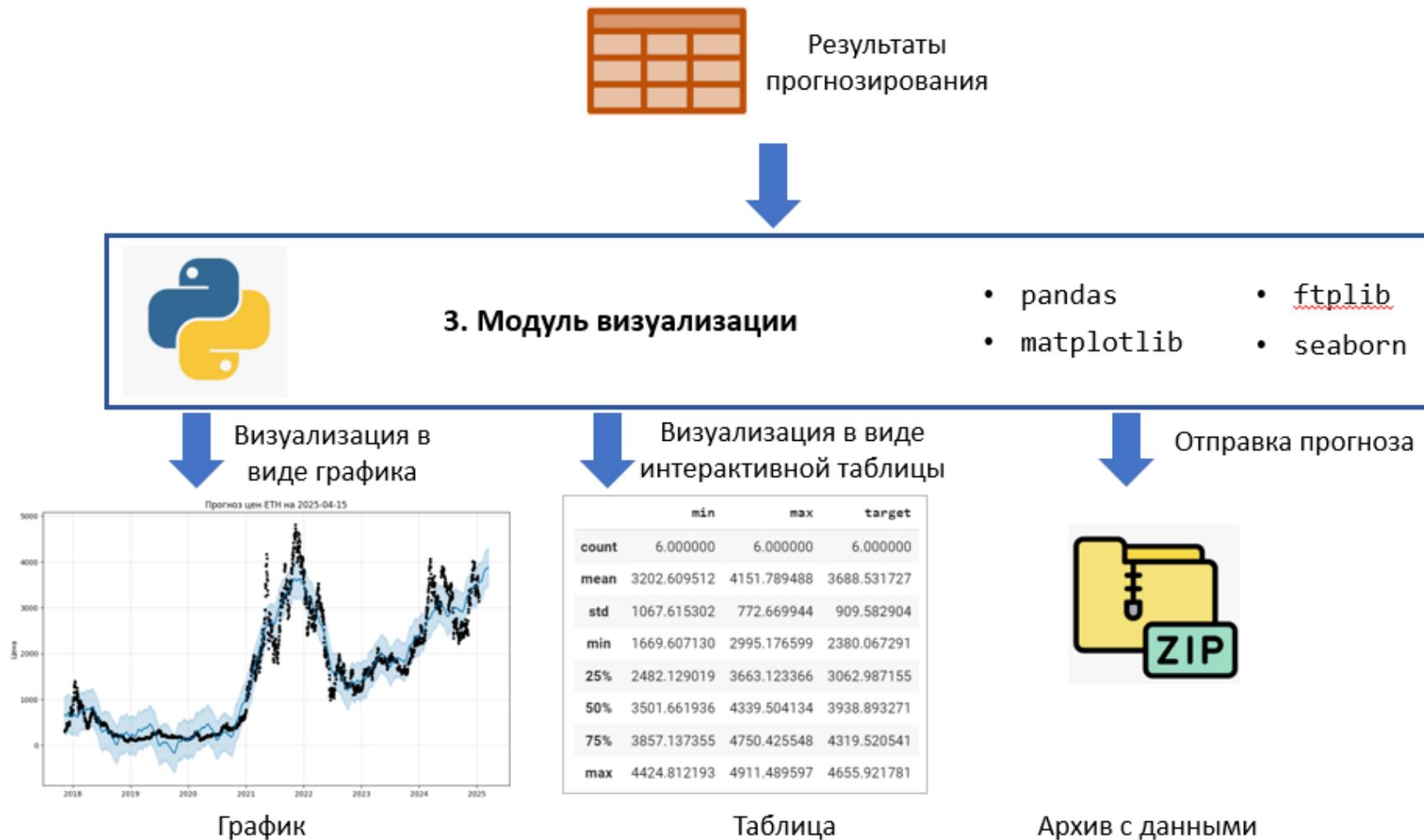


Рисунок 31 – Схема работы модуля для визуализации результатов прогнозирования

Данный модуль реализует процесс прогнозирования цен криптовалюты на заданное количество дней вперёд с использованием обученных ранее нейронных сетей, а также визуализацию и сохранение полученных результатов.

Сначала создаётся набор данных для прогноза на определённое число дней вперёд. Это достигается с помощью метода `make_future_dataframe`, который добавляет дополнительные временные точки к существующим данным для прогнозирования.

Затем модель прогнозирует значения на основе этого расширенного набора данных, результаты сохраняются в переменной `forecast`.

Для анализа полученных прогнозов используется визуализация. Метод `plot` строит график с предсказанными значениями, а также верхними и нижними границами доверительного интервала. Заголовок графика включает название криптовалюты (переменная `ticker_name`) и конечную дату прогноза (переменная `a`).

Дополнительно визуализируются компоненты прогноза, такие как тренд и сезонные колебания, с помощью метода `plot_components`.

После визуализации результаты прогноза сохраняются в CSV-файл. Выходной файл содержит столбцы с временными метками (`ds`), предсказанными значениями (`yhat`), а также границами доверительного интервала (`yhat_lower` и `yhat_upper`).

Это позволяет использовать прогнозы в дальнейшем для анализа или интеграции в другие системы.

Программный код для визуализации прогноза по BTC в виде графика показан на рисунке 32.

```

# Создание данных для прогноза на target_day дней вперед
future = model.make_future_dataframe( periods=target_day)
forecast = model.predict(future)

# Визуализация прогноза
fig = model.plot(forecast)
plt.title('Прогноз цен '+ ticker_name +' на '+a)
plt.xlabel('Дата')
plt.ylabel('Цена')
plt.show()

# Визуализация компонент прогноза
fig2 = model.plot_components(forecast)
plt.show()

# Сохранение прогноза в CSV файл
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].to_csv('data_forecast-max.csv', index=False)

```

Рисунок 32 – Программный код для визуализации прогноза по BTC в виде графика

Для того чтобы объединить данные прогнозов нейронной сети на различных диапазонах в одну матрицу применяет метод `np.array` с перечислением объединяемых данных. Затем полученный массив конвертируется в `DataFrame`.

Базовая статистика по полученному массиву рассчитывается и выводится на экран в виде с помощью метода `describe`. Данная таблица содержит такие значения, как прогнозируемое значение токена на указанную дату (столбец `target`, строка `mean`), так и минимально и максимально возможные цены на токен (столбцы `min` и `max`, строка `mean`). Программный код для визуализации прогноза по BTC в виде таблицы и пример выводимой таблицы показан на рисунке 33.

```
[ ] import numpy as np

all_data = pd.DataFrame(np.array([[t1mo_b, t1mo_t, t1mo_p],
    [t3mo_b, t3mo_t, t3mo_p],
    [t6mo_b, t6mo_t, t6mo_p],
    [t1y_b, t1y_t, t1y_p],
    [t2y_b, t2y_t, t2y_p],
    [t5y_b, t5y_t, t5y_p],
    [t10y_b, t10y_t, t10y_p]]), columns = ["min", "max", "target"])
all_data.describe()
```



	min	max	target
count	7.000000	7.000000	7.000000
mean	98500.968411	108526.186147	103508.512335
std	9497.887575	8038.444595	8698.168458
min	81925.567080	94531.901837	88364.775774
25%	94425.458155	105684.564992	99981.164898
50%	97822.804296	107225.952312	102276.541376
75%	105389.726131	114097.149129	110305.181861
max	110128.038930	118362.020641	113345.575675

Рисунок 33 – Программный код для визуализации прогноза по BTC в виде таблицы

Теперь протестируем работу программного обеспечения при решении задачи прогнозирования котировок криптовалют на временном диапазоне 90 дней.

4.4 Тестирование предложенных подходов

Протестируем предложенные подходы на практике спрогнозировав цену криптовалют через 90 дней. Прогнозируемая цена BTC на 23 апреля составляет – 97128\$. Диапазон, внутри которого будет находиться цена на эту дату – 90388\$...103905\$ (рисунок 34).

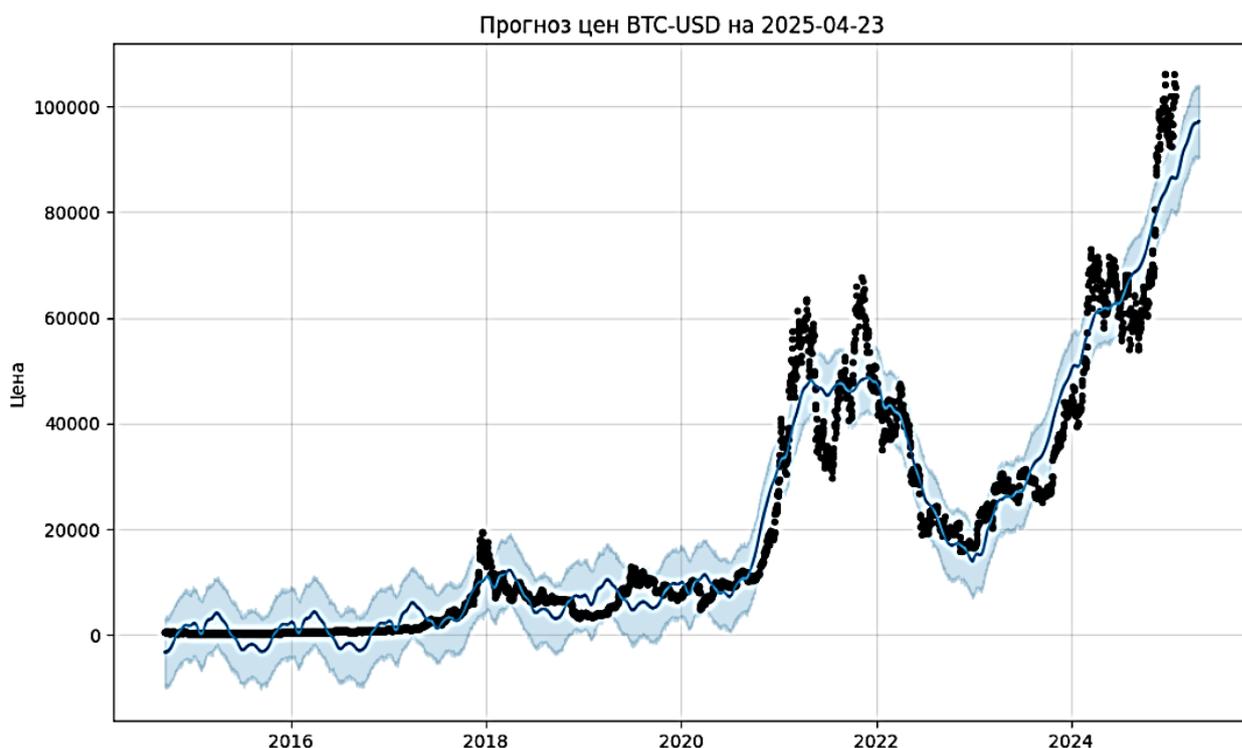


Рисунок 34 – Визуализация прогноза по BTC

Прогнозируемая цена BTC на 23 апреля составляет – 1,46\$. Диапазон, внутри которого будет находиться цена на эту дату – 1,07\$...1,80\$ (рисунок 35).

Прогнозируемая цена DASH на 23 апреля составляет – 46,14\$. Диапазон, внутри которого будет находиться цена на эту дату – 3,15\$...135,78\$ (рисунок 36).

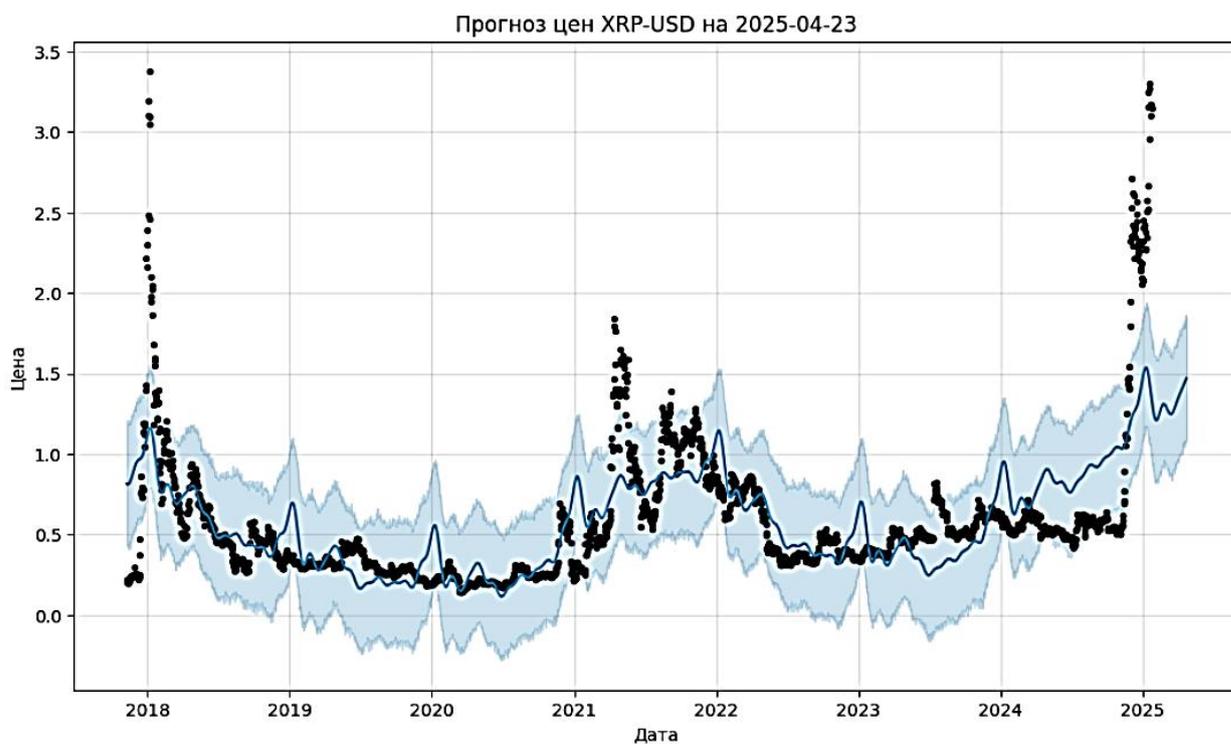


Рисунок 35 – Визуализация прогноза по XRP

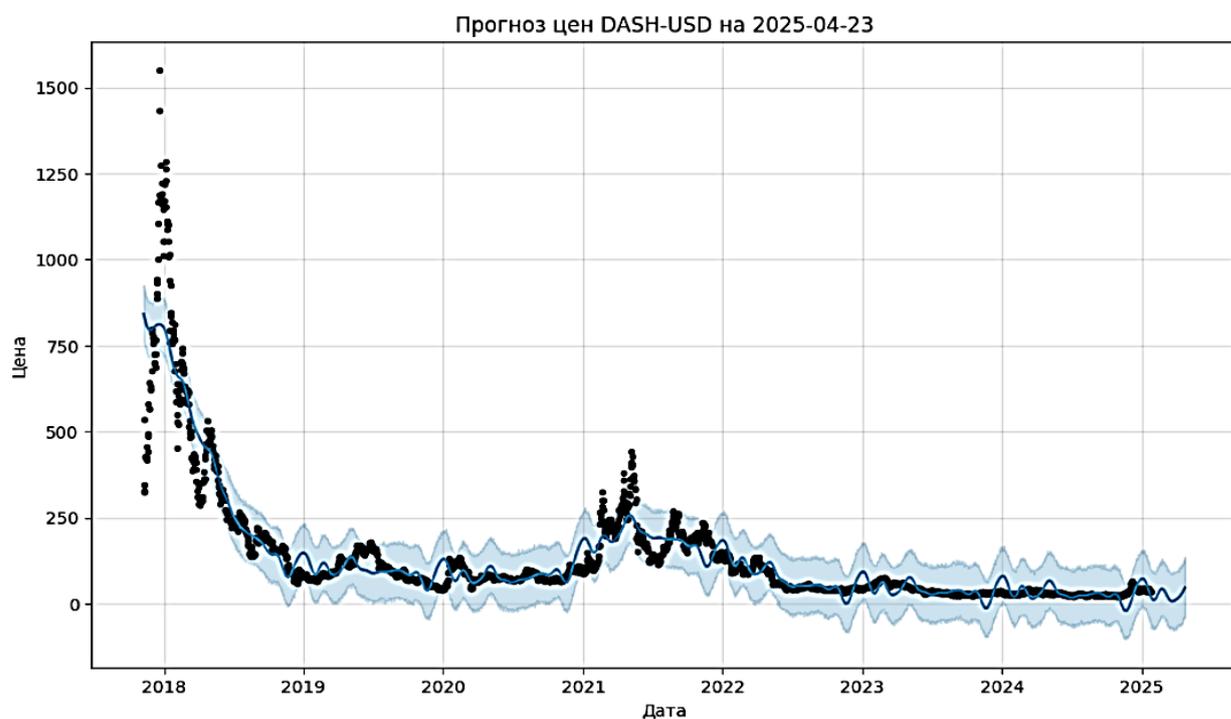


Рисунок 36 – Визуализация прогноза по DASH

Прогнозируемая цена DASH на 23 апреля составляет – 197,8\$. Диапазон, внутри которого будет находиться цена на эту дату – 154,9\$...238,7\$

(рисунок 37).

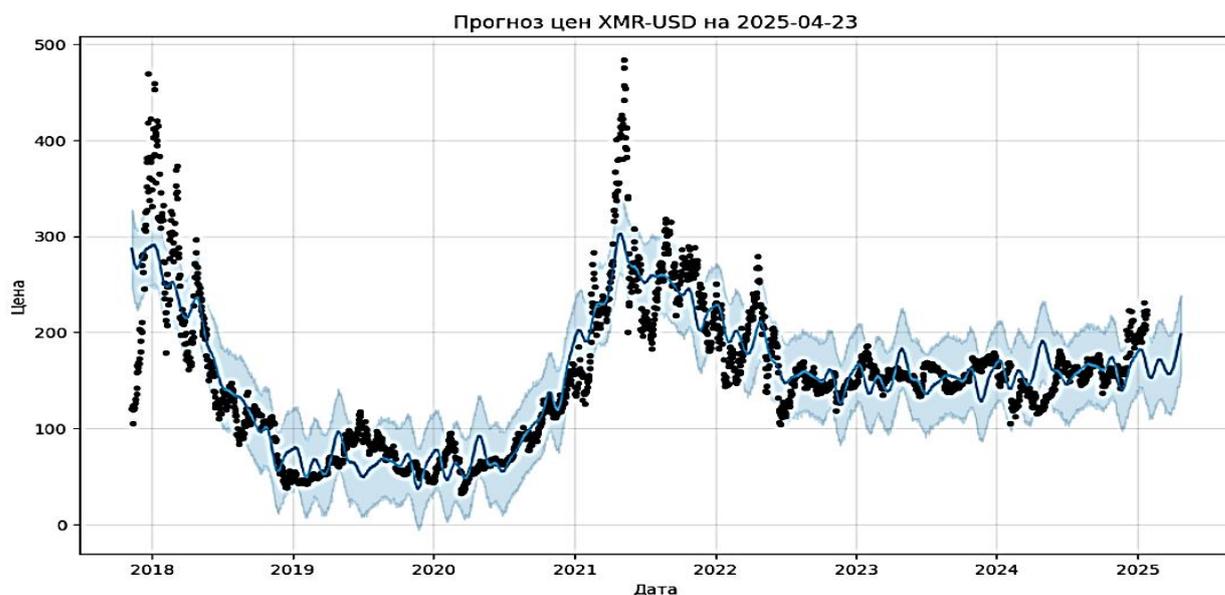


Рисунок 37 – Визуализация прогноза по XMR

Прогнозируемая цена LTC на 23 апреля составляет – 102,46\$. Диапазон, внутри которого будет находиться цена на эту дату – 72,16\$...135,29\$ (рисунок 38).

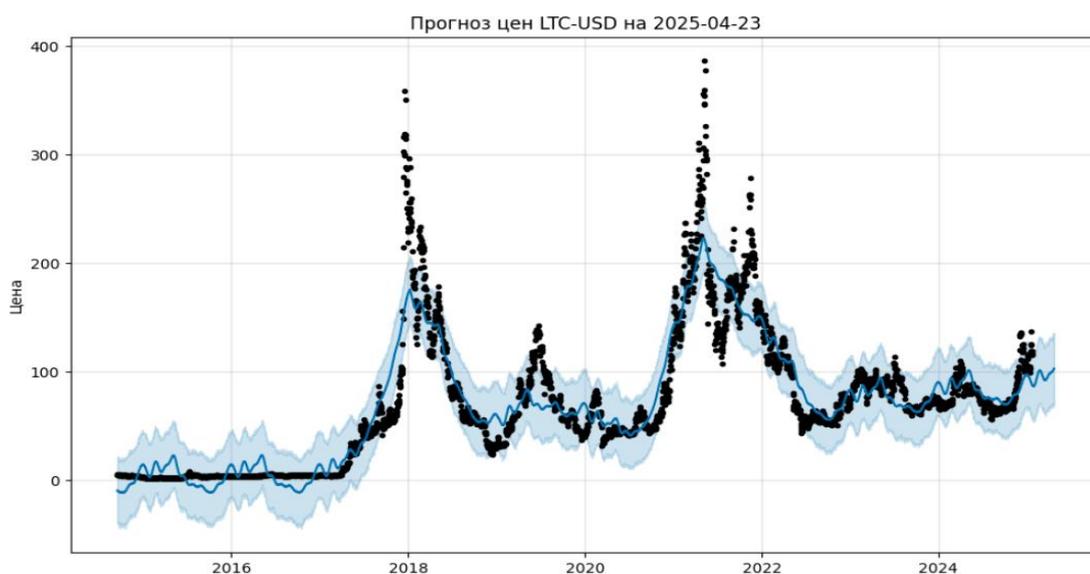


Рисунок 38 – Визуализация прогноза по LTC

Как видно из рисунков 34-38, предложенные в магистерском исследовании подходы позволяют получать адекватные прогнозы по изменению котировок криптовалют.

Выводы по главе 4

Продемонстрирована реализация предложенных в диссертационной работе подходов виде программных модулей, разработанных на языке программирования Python, которые выполняют этапы загрузки, предобработки данных, снижения размерности признаков, обучения нейронных сетей и проверки их точности при решении задачи прогнозирования котировок криптовалют.

Программная реализация на практике подтвердила эффективность применения метода SHAP для отбора значимых признаков и эффективность использования архитектур LSTM и GRU для прогнозирования криптовалютного рынка.

Заключение

Приведем основные результаты исследования:

- проведен анализ литературных источников по теме прогнозирования котировок криптовалют. Существующие исследования имеют следующие недостатки: рассматривается вопрос прогнозирования одной криптовалюты без обобщения полученных результатов на другие, не используется весь потенциал технологий машинного обучения (например, снижение размерности датасета и кросс-валидация), при прогнозировании осуществляется анализ признаков только из одного типа индикаторов (или фундаментальных или технических);
- рассмотрены технические, фундаментальные и макроэкономические показатели, которые могут быть использованы при прогнозировании криптовалют, а также приведены их описание и характеристики;
- проведено сравнение нейронных сетей, подходящих для решения задачи прогнозирования криптовалют. Выбраны 4 типа нейронных сетей, которые будут использованы при создании системы прогнозирования котировок криптовалют. Среди них сети LSTM, GRU, biLSTM и sLSTM;
- разработана методика предобработки данных котировок криптовалют, включающая в себя нормировку данных и фильтрацию признаков на основе определения их значимости по значению SHAP;
- разработанная методика предобработки данных котировок криптовалют в сочетании с нейронной сетью LSTM позволила снизить количество используемых для прогнозирования фундаментальных показателей и технических индикаторов суммарно на 56% без существенного увеличения ошибки RMSE;

- установлено, что для задач прогнозирования котировок криптовалют архитектуры рекуррентных нейронных сетей, такие как GRU и LSTM, превосходят многослойный перцептрон;
- установлено, что использование более сложных архитектур, таких как biLSTM и sLSTM, позволяет повысить точность прогнозирования в задаче анализа криптовалютных котировок, особенно при наличии большого количества данных;
- наиболее оптимальной моделью с точки зрения точности и сложности реализации при анализе криптовалютных котировок является sLSTM, которая обеспечивает минимальную ошибку RMSE при прогнозировании на основе как технических индикаторов, так и фундаментальных показателей;
- рекомендуется использовать sLSTM в сочетании с предложенным в работе подходом предобработки данных, включающим в себя нормализацию данных и снижение размерности, для построения высокоточных систем прогнозирования криптовалютных рынков;
- продемонстрирована возможность реализации предложенных в диссертационной работе подходов виде программных модулей, разработанных на языке программирования Python, которые выполняют этапы загрузки, предобработки данных, снижения размерности признаков, обучения нейронных сетей и проверки их точности при решении задачи прогнозирования котировок криптовалют;
- осуществляется предобработка данных, включающая разделение на обучающую и тестовую выборки, а также их нормализацию, что позволяет привести данные к масштабам, подходящим для обучения нейронных сетей. Затем выполняется снижение размерности данных. Для этого создаётся временная модель, которая используется в функции `reduce_dimensionality` для вычисления важности признаков с помощью SHAP и отбора

наиболее значимых из них;

- результатом являются уменьшенные по размерности выборки для обучения и тестирования. После этого создаются и обучаются две разные модели. Первая из них основана на архитектуре LSTM, подходящей для анализа временных рядов. Она строится с использованием функции `create_lstm_model`, а затем обучается на подготовленных данных с помощью функции `train_and_evaluate`. Аналогичным образом создаётся и обучается модель на основе архитектуры GRU, которая также предназначена для обработки последовательностей данных, но отличается более простой структурой.

Обе модели прошли оценку на тестовой выборке, что позволило проверить их производительность и точность.

Гипотеза исследования подтверждена.

Список используемой литературы и используемых источников

1. Акимов, С.С. ПРОГНОЗИРОВАНИЕ ЦЕН АКЦИЙ ПРИ ПОМОЩИ ИДЕНТИФИКАЦИИ ЗАКОНА РАСПРЕДЕЛЕНИЯ / С.С. АКИМОВ // Интеллект. Инновации. Инвестиции. 2018. №9. URL: <https://cyberleninka.ru/article/n/prognozirovanie-tsen-aktsiy-pri-pomoschi-identifikatsii-zakona-raspredeleniya> (дата обращения: 16.01.2025).
2. Бугорский, В.Н. Использование нейронных сетей для моделирования прогноза котировок ценных бумаг / Бугорский Владимир Николаевич, Сергиенко Анатолий Геннадьевич // Прикладная информатика. 2008. №3. URL: <https://cyberleninka.ru/article/n/ispolzovanie-neyronnyh-setey-dlya-modelirovaniya-prognoza-kotirovok-tsennyh-bumag> (дата обращения: 16.01.2025).
3. Жданов А. И., Муравьев Д. Г. Об одном регрессионном методе прогноза котировок валют // Вестник СГАУ. 2005. №1. URL: <https://cyberleninka.ru/article/n/ob-odnom-regressionnom-metode-prognoza-kotirovok-valyut> (дата обращения: 16.01.2025).
4. Казаковцева М.В. Прогнозирование котировок финансовых инструментов с помощью нейронных сетей / Казаковцева Марина Вадимовна, Конакова Екатерина Витальевна // Вестник Марийского государственного университета. Серия «Сельскохозяйственные науки. Экономические науки». 2023. №4. URL: <https://cyberleninka.ru/article/n/prognozirovanie-kotirovok-finansovyh-instrumentov-s-pomoschyu-neyronnyh-setey> (дата обращения: 16.01.2025).
5. Кратович, П. В. Нейронные сети и модели Arima для прогнозирования котировок / П.В. Кратович // Программные продукты и системы. 2011. №1. URL: <https://cyberleninka.ru/article/n/neyronnye-seti-i-modeli-arima-dlya-prognozirovaniya-kotirovok> (дата обращения: 16.01.2025).
6. Кузнецов, Р. С. Прогнозирование котировок акций пао газпром с использованием нейронных сетей LSTM / Кузнецов Р. С., Тумарова Т. Г. //

Вестник Института экономики Российской академии наук. 2023. №3. URL: <https://cyberleninka.ru/article/n/prognozirovanie-kotirovok-aktsiy-pao-gazprom-s-ispolzovaniem-neyronnyh-setey-lstm> (дата обращения: 16.01.2025).

7. Кузнецов, Р.С. Прогнозирование котировок акций китайской компании NIO с использованием двунаправленных LSTM нейронных сетей / Кузнецов Роман Сергеевич, Тумарова Татьяна Гельцевна, Пилипенко Сергей Викторович // ЭПИ. 2023. №4. URL: <https://cyberleninka.ru/article/n/prognozirovanie-kotirovok-aktsiy-kitayskoy-kompanii-nio-s-ispolzovaniem-dvunapravlennyh-lstm-neyronnyh-setey> (дата обращения: 16.01.2025).

8. Петров С. С., Трушанина О. Ю. Краткосрочное прогнозирование цен акций на основе анализа тенденций спроса и предложения на фондовой бирже // Финансовая аналитика: проблемы и решения. 2012. №12. URL: <https://cyberleninka.ru/article/n/kratkosrochnoe-prognozirovanie-tsen-aktsiy-na-osnove-analiza-tendentsiy-sprosa-i-predlozheniya-na-fondovoy-birzhe> (дата обращения: 16.01.2025).

9. Погорелов, Н.Е. Интеллектуальная информационная система для анализа и прогнозирования биржевых котировок акций / Погорелов Никита Евгеньевич, Рейзенбук Кристина Эдуардовна, Пимонов Александр Григорьевич // Вестник КузГТУ. 2012. №6(94). URL: <https://cyberleninka.ru/article/n/intellektualnaya-informatsionnaya-sistema-dlya-analiza-i-prognozirovaniya-birzhevyyh-kotirovok-aktsiy> (дата обращения: 16.01.2025).

10. Рыжкина, Д. А. ПРОГНОЗИРОВАНИЕ КОТИРОВОК АКЦИЙ КОМПАНИИ ЯНДЕКС В СРЕДЕ FUZZYTECH / Д.А. Рыжкина // Форум молодых ученых. 2017. №5 (9). URL: <https://cyberleninka.ru/article/n/prognozirovanie-kotirovok-aktsiy-kompanii-yandeks-v-srede-fuzzytech> (дата обращения: 16.01.2025).

11. Смирнов, Д. С. Методы прогнозирования динамики фондового рынка и их ограничения / Смирнов Д. С. // Пространство экономики. 2009. №2-

3. URL: <https://cyberleninka.ru/article/n/metody-prognozirovaniya-dinamiki-fondovogo-rynka-i-ih-ogranicheniya> (дата обращения: 16.01.2025).

12. Солдатова, О.П. Исследование погрешности прогнозирования котировок акций при помощи модели нечеткой нейронной сети Ванга - Менделя / Солдатова Ольга Петровна // Известия вузов. Поволжский регион. Технические науки. 2015. №4 (36). URL: <https://cyberleninka.ru/article/n/issledovanie-pogreshnosti-prognozirovaniya-kotirovok-aktsiy-pri-pomoschi-modeli-nechetkoj-neyronnoy-seti-vanga-mendelya> (дата обращения: 16.01.2025).

13. Щекин, А.А. Методы и модели прогнозирования курса криптовалютных активов на основе анализа временных рядов и факторных данных / Щекин А.А // Вестник науки: №4 (85) том 4. С. 750 - 756. 2025 г.

14. Alonso-Monsalve, S. Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators / S. Alonso-Monsalve, A. L. Suárez-Cetrulo, A. Cervantes, and D. Quintana // Expert Systems with Applications, 2020, vol. 149, p. 113250.

15. Bitcoin: A peer-to-peer electronic cash system: [Электронный ресурс] // Nakamoto. URL: <http://bitcoin.org/bitcoin.pdf>. (Дата обращения: 10.11.2023)

16. Chen, Y. Integrating principle component analysis and weighted support vector machine for stock trading signals prediction / Y. Chen, Y. Hao // Neurocomputing, 2018, vol. 321, pp. 381–402.

17. Chung, J. Empirical evaluation of gated recurrent neural networks on sequence modeling / J. Chung, C. Gulcehre, K. Cho, Y. Bengio // NIPS 2014 Workshop on Deep Learning, 2014.

18. Corbet, S. The impact of macroeconomic news on bitcoin returns / S. Corbet, C. Larkin, B. M. Lucey, A. Meegan, L. Yarovaya // The European Journal of Finance, 2020, vol. 26, no. 14, pp. 1396–1416

19. Cryptocurrency Prices, Charts And Market Capitalizations: [Электронный ресурс] // URL: <https://coinmarketcap.com/>. (Дата обращения: 10.11.2023)

20. Dutta, A. A gated recurrent unit approach to bitcoin price prediction / A. Dutta, S. Kumar, M. Basu // *Journal of Risk and Financial Management*, 2020, vol. 13, p. 23.
21. Graves, A. Framewise phoneme classification with bidirectional lstm and other neural network architectures / A. Graves, J. Schmidhuber // *IJCNN 2005 Neural Networks*, 2005, vol. 18, no. 5, pp. 602–610.
22. Hansson, M. On stock return prediction with lstm networks / Magnus Hansson // 2017. – pp. 36.
23. Hochreiter, S. Long Short-Term Memory / S. Hochreiter, J. Schmidhuber // *Neural Computation*, vol. 9, 1997, pp. 1735–1780.
24. Huang, W. Neural networks in finance and economics forecasting / W. Huang, K. K. Lai, Y. Nakamori, S. Wang, L. Yu // *International Journal of Information Technology & Decision Making*, 2007, vol. 06, no. 01, pp. 113– 140.
25. Karakoyun, E. Comparison of arima time series model and lstm deep learning algorithm for bitcoin price forecasting / E. Karakoyun, A. Cibikdiken // *The 13th Multidisciplinary Academic Conference in Prague*, 2018, vol. 2018, pp. 171–180,
26. Neely, C. J. Forecasting the Equity Risk Premium: The Role of Technical Indicators / C. J. Neely, D. E. Rapach, J. Tu, G. Zhou // *Management Science*, July 2014, vol. 60, pp. 1772–1791
27. Qiang, Z. Bitcoin High-Frequency Trend Prediction with Convolutional and Recurrent Neural Networks / Z. Qiang // *Stanford winter report*, Dtanford, Mar. 2021, pp. 34
28. Wang, W. Portfolio formation with preselection using deep learning from long-term financial data / W. Wang, W. Li, N. Zhang, K. Liu // *Expert Systems with Applications*, 2020, vol. 143, p. 113042.
29. Yildirim, D.C. Forecasting directional movement of forex data using lstm with technical and macroeconomic indicators / D. C. Yildirim, I. H. Toroslu, and U. Fiore // *Financial Innovation*, Jan 2021, vol. 7, pp. 10.
30. Zoumpikas, T. ETH analysis and predictions utilizing deep learning /

T. Zoumpikas, E. Houstis, M. Vavalis // Expert Systems with Applications, 2020,
vol. 162, p. 113866.