МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»
Кафедра дискретной математики и информационных технологий

РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ ПРЕДСКАЗАНИЯ ВОЛАТИЛЬНОСТИ ФИНАНСОВЫХ ИНСТРУМЕНТОВ С ПОМОЩЬЮ СТАТИСТИЧЕСКИХ МОДЕЛЕЙ И МАШИННОГО ОБУЧЕНИЯ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 421 группы
направления 09.03.01 — Информатика и вычислительная техника
факультета КНиИТ
Варыпаева Артема Андреевича

Научный руководитель
к.ф.м.н., д.экон.н, профессор _____ Л. В. Кальянов

Заведующий кафедрой
доцент, к.ф.-м.н. _____ Л. Б. Тяпаев

Саратов 2024

ВВЕДЕНИЕ

Волатильность, или изменчивость, представляет собой меру колебаний цен на финансовые активы, такие как акции, валюты и сырьевые товары. В базовом смысле волатильность отражает степень неопределенности или риска, связанного с изменением стоимости актива. Когда говорят, что актив имеет высокую волатильность, это означает, что его цена может резко изменяться в короткие промежутки времени. Напротив, низкая волатильность свидетельствует о стабильных и предсказуемых изменениях цены.

С научной точки зрения, волатильность определяется как статистический показатель дисперсии доходности финансового инструмента. Данная мера изменчивости показывает разброс данных. Это ключевая метрика в финансах, которая широко используется для оценки риска и формирования инвестиционных стратегий. В финансовой теории и практике различают историческую волатильность, измеряемую на основе прошлых данных, и подразумеваемую волатильность, основанную на текущих рыночных ожиданиях.

Актуальность исследования волатильности трудно переоценить. В современных условиях прогнозирование волатильности становится критически важным для участников финансовых рынков, включая инвесторов, трейдеров, и регуляторов. Умение точно предсказывать волатильность позволяет не только минимизировать риски, но и оптимизировать доходность инвестиционных портфелей, что является значимым преимуществом в условиях высокой неопределенности и динамичности финансовых рынков.

В последние десятилетия наблюдается значительное развитие методов прогнозирования волатильности. Классические методы описанные ещё в двадцатом веке, успешно используются для анализа временных рядов и предсказания будущих изменений волатильности. Однако, с развитием технологий и доступом к большим объемам данных, становятся популярными более сложные и точные подходы, такие как методы машинного обучения и глубокие нейронные сети. Эти современные методы позволяют учитывать нелинейные зависимости и сложные взаимодействия в данных, что способствует более точным прогнозам.

Целью данной работы является создание веб-приложения для сравнения и использования различных статистических моделей и моделей машинного обучения для предсказания волатильности. В рамках исследования будут

рассмотрены как традиционные эконометрические методы, так и современные подходы машинного обучения. Основными задачами работы являются:

1. Анализ существующих методов прогнозирования временных рядов и их теоретических основ.
2. Подключение и настройка предсказывающих моделей.
3. Разработка веб-приложения и включение моделей в программу.
4. Реализация сравнения эффективности моделей.

Под волатильностью в данной работе подразумевается скользящее среднеквадратичное отклонение. Данные о котировках активов предоставляются в реальном времени сервисом Yahoo! Finance.

Данная работа имеет следующую структуру: введение, в котором обосновывается актуальность темы, а также формулируются цель и задачи; первая глава под названием «Теоретические основы использованных технологий», состоящая из 13 разделов; вторая глава «Разработка веб-приложения для прогнозирования волатильности», состоящая из двух разделов; заключение; список использованных источников и приложения с кодом программы.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

В первой главе под названием «Теоретические основы использованных технологий» описываются инструменты, которые были использованы в работе. Описываются такие статистические модели, как AR, ARMA и ARIMA.

Авторегрессионная модель (AR) используется для анализа временных рядов и предсказания будущих значений на основе прошлых данных. Основной идеей модели является предположение, что текущее значение временного ряда зависит от его предыдущих значений.

Автогрессионная модель со скользящим средним (ARMA) комбинирует в себе авторегрессионную модель (AR) и модель скользящего среднего (MA). Эта модель может моделировать как зависимости от предыдущих значений временного ряда, так и от предыдущих ошибок прогноза.

Автогрессионная модель со скользящим средним и интегрированием (ARIMA) расширяет ARMA модель за счет включения интегрирования, что позволяет работать с нестационарными временными рядами. Другими словами, эта модель делает временной ряд стационарным.

Для реализации моделей AR, ARMA и ARIMA была использована библиотека statsmodels на языке программирования Python. Statsmodels — это модуль, предоставляющий классы и функции для оценки множества различных моделей, а также он удобен для проведения статистических тестов и анализа данных. Выбор пал именно на эту библиотеку, потому что она проста в использовании и не требует большого количества кода для реализации.

Помимо этого были использованы модели машинного обучения.

Линейная регрессия — это классическая модель, используемая для анализа зависимости между одной или несколькими независимыми переменными и одной зависимой переменной. Она основана на предположении о линейной связи между переменными.

Деревья решений — модель, используемая для принятия решений на основе последовательности условий. Деревья разделяют набор данных на более мелкие группы, решая задачу классификации или регрессии (для данной работы — только регрессии). Каждый узел дерева представляет собой вопрос о значении одной из переменных, а каждый лист дерева соответствует прогнозу.

Случайный лес — это ансамблевая модель, состоящая из множества

деревьев решений, обученных на случайных подвыборках данных. Он комбинирует прогнозы всех деревьев для получения более точного и устойчивого результата.

Градиентный бустинг — это также ансамблевый метод, в котором модели строятся последовательно, при этом каждая новая модель исправляет ошибки предыдущей.

Для реализации перечисленных моделей была использована библиотека scikit-learn (sklearn) на языке программирования Python. Scikit-learn предоставляет простой и эффективный инструментарий для машинного обучения и анализа данных. Её гибкость и многофункциональность позволяют легко реализовывать различные методы машинного обучения, включая модели линейной регрессии, деревьев решений, случайного леса и градиентного бустинга.

Отдельно стоит сказать про модели построенные на основе нейронных сетей.

Полносвязные слои — это основной компонент многих моделей глубокого обучения. Каждый нейрон в такого слоя связан с каждым нейроном предыдущего и следующего слоя, это позволяет алгоритму обучаться извлекать сложные зависимости из данных.

Рекуррентные нейронные сети (RNN) — это класс нейронных сетей, способных обрабатывать последовательные данные, сохраняя информацию о предыдущих входах, потому что у этого слоя есть ячейка памяти.

Долгая краткосрочная память (LSTM) — это тип рекуррентных нейронных сетей, способных эффективно обрабатывать длинные зависимости в последовательных данных.

Управляемый рекуррентный блок (GRU) — это еще один тип рекуррентных нейронных сетей, похожий на модель LSTM. GRU можно считать модернизацией предыдущей модели, потому что GRU имеет меньше параметров, чем LSTM и при прочих равных, быстрее учится, но, несмотря на это, GRU и LSTM показывают сопоставимое качество на многих задачах.

Сверточные нейронные сети (CNN) — это класс нейронных сетей, особенно хорошо зарекомендовавший себя в обработке изображений, однако они хорошо справлются и с обработкой временных рядов.

Для реализации этих моделей была использована библиотека Keras.

Keras является высокоуровневым интерфейсом для создания и обучения нейронных сетей, который позволяет легко и быстро строить сложные модели глубокого обучения с минимальным количеством кода.

Для разработки веб-приложения, предназначенного для предсказания волатильности, был использован фреймворк Flask. Flask — это легковесный и гибкий фреймворк для создания веб-приложений на языке программирования Python. Особенность этого фреймворка в том, что он позволяет с лёгкостью перейти на более сложные библиотеки для веб-приложений, если у разработчиков появляется такая необходимость.

Вторая глава под названием «Разработка веб-приложения для прогнозирования волатильности» содержит два раздела: «Описание интерфейса программы» и «Техническое описание веб-приложения».

В первом разделе описывается структура и функциональность интерфейса веб-приложения. Основные элементы управления доступны пользователю на главной странице разработанной программы, которая изображена на рисунке 1.

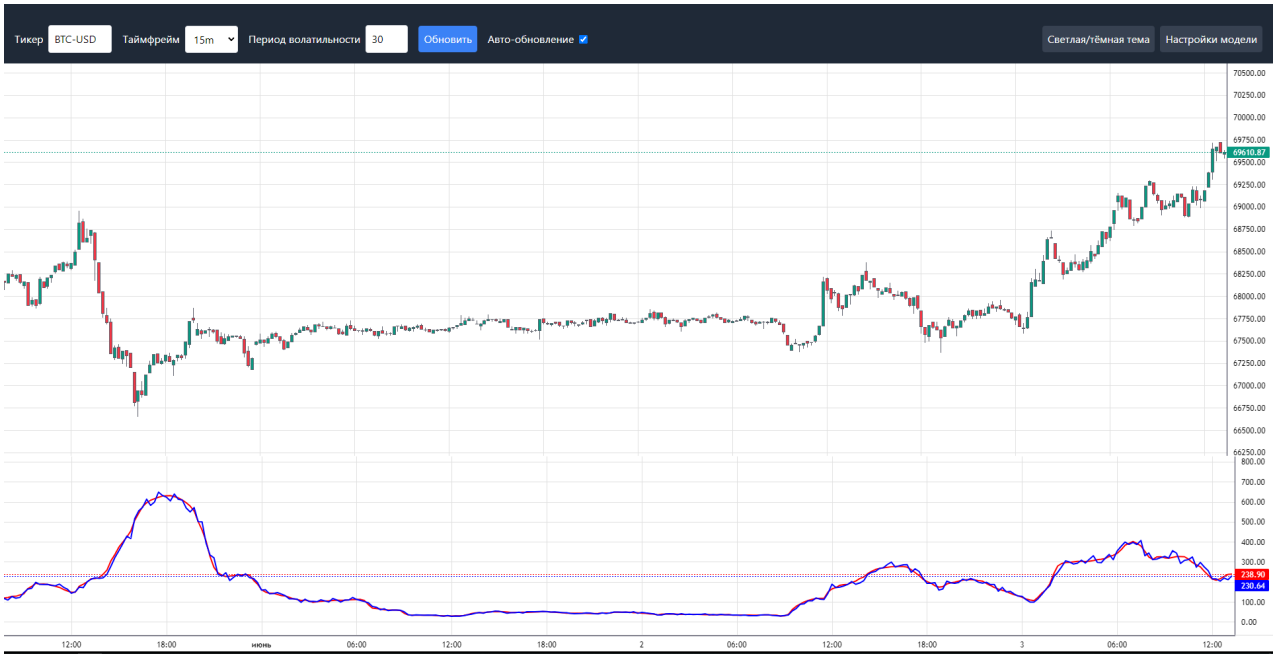

Рисунок 1 — Главная страница программы

В верхней части экрана находится панель управления с несколькими основными настройками, показанным на рисунке 2. Пользователь может ввести сокращённое название финансового инструмента в соответствующее поле, выбрать временной период (таймфрейм) из раскрывающегося списка и

установить период для расчёта волатильности. Кнопка **«Обновить»** позволяет получить обновленные данные на основе введенных параметров. Опция **«Автоматическое обновление»** автоматизирует этот процесс.

Рисунок 2 — Верхняя панель приложения

Панель управления также включает кнопки изменения темы интерфейса и открытия настроек моделей. На рисунке 3 можно увидеть внешний вид программы в тёмной теме. Все эти элементы управления обеспечивают быстрый доступ к основным функциям программы и делают ее простой и интуитивно понятной в использовании.

Рисунок 3 — Тёмная тема приложения

Основная часть экрана разделена на две части: свечной график котировок актива и график стандартного отклонения. График стандартного отклонения показывает фактическую волатильность красным цветом и прогнозируемую моделью волатильность синим цветом. Это позволяет пользователю легко сравнивать фактические данные с ожидаемыми, оценивать точность моделей и делать выводы на основе визуального анализа, а также видеть прогноз на будущее.

В правой части экрана находится панель выбора модели прогнозирования, которая открывается по нажатию на кнопку **«Настройки модели»**.

Панель, которую можно увидеть на рисунке 4, содержит таблицу со списком доступных шаблонов. Пользователь может нажать на каждую строку таблицы, чтобы выбрать интересующую модель. Выбранная модель выделяется зеленым цветом. В таблице также показана метрика MAPE (средняя абсолютная процентная ошибка) для каждой модели, которая позволяет пользователю быстро оценить производительность различных моделей.

Рисунок 4 — Панель настройки модели

Под таблицей выбора модели расположен ползунок, позволяющий регулировать объем набора данных для обучающих и тестовых выборок. Этот ползунок особенно полезен, если пользователь не хочет использовать весь доступный набор данных, чтобы значительно сократить время обучения модели. Рядом с ползунком отображается количество значений, используемых в тренировочной и тестовой выборке.

Кнопка **«Автоматически найти лучшую модель»** запускает процесс автоматического выбора лучшей модели из списка. После завершения процесса лучшая модель автоматически выбирается и выделяется зеленым цветом.

Во втором разделе описывается разделение веб-приложения на две части. Первая часть — клиентская сторона, а второрая — серверная часть.

Клиентская часть, также известная как фронтенд, — это та часть веб-приложения, с которой напрямую взаимодействует пользователь. Она включает в себя всё, что можно увидеть в браузере: тексты, изображения, кнопки

интерфейса. Фронтенд использует HTML для структурирования контента, CSS для стилизации и JavaScript для добавления интерактивности. Эти технологии позволили создать динамичный и адаптивный интерфейс, который мгновенно реагирует на действия пользователя, не требуя перезагрузки страницы.

Серверная часть, или бэкенд, — это сервер, который обрабатывает запросы, поступающие от клиентской части. Он выполняет бизнес-логику приложения, управляет базами данных, обеспечивает безопасность данных и возвращает информацию обратно клиенту.

Клиентская часть приложения, реализованная в файлах index.html и main.js, отвечает за отображение и анализ финансовых данных через интерактивный интерфейс.

Файл index.html описывает каркас, структурирует и оформляет веб-страницу с использованием HTML и Tailwind CSS, создавая заголовок, основной блок с графиками и боковую панель для выбора параметров. Подключенная библиотека Lightweight Charts от Trading View используется для отображения свечных графиков, обеспечивая их оптимальное отображение при изменении масштаба.

Файл main.js отвечает за загрузку и отображение данных, обновление графиков и обработку пользовательских взаимодействий. Именно он делает веб-приложение интерактивным. Этот файл содержит множество различных обработчиков событий, с помощью которых происходит обмен информацией между клиентской частью и серверной.

Серверная часть приложения состоит из нескольких файлов, главным из которых является app.py. Остальные файлы содержат реализации моделей машинного обучения и статистических методов.

Файл app.py является центральным элементом серверной части. Он реализует веб-сервер на базе Flask, обрабатывает запросы, получает данные, выполняет расчеты и предоставляет данные клиентской части. В этом файле используются библиотеки Flask для создания веб-сервера, yfinance для загрузки финансовых данных, pandas и pandas_ta для обработки данных и стандартные модули Python для работы с датами и временем. Основная задача файла — обеспечить клиентскую часть данными для визуализации и анализа финансовых графиков.

Flask используется для создания веб-сервера и маршрутизации запросов, yfinance — для загрузки финансовых данных с Yahoo Finance, pandas и pandas_ta — для обработки данных и расчета стандартного отклонения, а стандартные модули Python — для работы с датами и временем.

Ключевая функция `fetch_yahoo_data` загружает и обрабатывает данные. Она принимает параметры тикера, интервала, периода стандартного отклонения, длины датасета и модели предсказания. Функция определяет даты начала и конца периода, загружает данные с Yahoo Finance, обрабатывает их, рассчитывая стандартное отклонение, и выполняет различные модели предсказания.

Результаты обработки данных преобразуются в формат, удобный для отправки клиенту. Свечные данные и данные стандартного отклонения преобразуются в списки словарей с временной меткой и значениями.

Основные маршруты в Flask включают `index`, который загружает начальную HTML-страницу, и `get_data`, который возвращает финансовые данные в формате JSON. Маршрут `get_data` вызывает функцию `fetch_yahoo_data`, передает ей параметры из URL и возвращает JSON-ответ с данными. Маршрут `index` возвращает HTML-шаблон index.html, предоставляя интерфейс для взаимодействия пользователя с приложением.

Файл app.py подключается к отдельным файлам, в которых хранятся модели. Модели предсказания вынесены специально в отдельный файл, чтобы обеспечить модульность, это означает, что разработка может вестись параллельно, могут добавляться новые модели, они могут быть модернизированы без вмешательства в основной файл app.py.

ЗАКЛЮЧЕНИЕ

В ходе данной дипломной работы было разработано веб-приложение для предсказания волатильности финансовых инструментов. Основной целью проекта было создание инструмента для анализа и прогнозирования волатильности различных финансовых активов с использованием современных моделей временных рядов и веб-технологий, который может быть использован как частными трейдерами, так и институциональными игроками.

Клиентская часть приложения обеспечивает высокую производительность и интерактивность графиков, а постоянное взаимодействие с серверной частью позволяет динамически обновлять данные без перезагрузки страницы. Серверная часть обрабатывает запросы и предоставляет данные для визуализации финансовых графиков и анализа. Приложение использует современные технологии для загрузки, обработки и анализа финансовых данных, а также для предсказания волатильности с помощью различных моделей временных рядов и машинного обучения.

Пользователи могут выбирать тикеры, временные интервалы и модели предсказания, а также пользоваться функцией автоматического обновления данных и смены тем оформления.

Проект можно развивать, добавляя новые модели предсказания, улучшая интерфейс и добавляя новые функции для более глубокого анализа и визуализации данных. Выполненная работа продемонстрировала успешное сочетание технологий веб-разработки и методов анализа временных рядов, что позволило создать функциональное и удобное в использовании приложение для предсказания волатильности финансовых инструментов.

Основные источники информации:

1 Hu, Yan, Ni, Jian, Wen, Liu: A hybrid deep learning approach by integrating LSTM-ANN networks with GARCH model for copper price volatility prediction , Physica A: Statistical Mechanics and its Applications (2020)

2 Бриллинджер Д. Временные ряды. Обработка данных и теория / М. : ИНФРА-М 2015. 420 с.

3 Mills T. C. The Econometric Modelling of Financial Time Series / Cambridge University Press, 1993. p. 545.

4 Damodar N. Gujarati. Basic Econometrics. // The McGraw-Hill Companies. 2004. P. 1002.

5 Schibata R. Selection of the Order on an Autoregressive Model by Akaike's Information Criterion // Biometrika. 1996. Vol. 63. P. 147-164.

6 Andersen T., Bollerslev T, Diebold F. X., Labys P. (2003). Modeling and forecasting realized volatility. Econometrica, 71 (2), 579–625.

7 Bollerslev T. Generalized autoregressive conditional heteroskedasticity // Journal of Econometrics. 1986. Vol. 31. P. 307-327.

8 Кротова, Ю. И., Файзлиев, А. Р., Луньков, А. Д. 2021. Прогнозирование волатильности доходности индекса ММВБ с помощью комбинации ARMA и GARCH моделей. Саратовский национальный исследовательский государственный университет им. Н. Г. Чернышевского, Россия.

9 Corsi F. (2009). A simple approximate long-memory model of realized volatility. Journal of Financial Econometrics, 7 (2), 174–196

10 Hansen P., Lunde A. (2004).A forecast comparison of volatility models: Does anything beat a GARCH(1,1)? Journal of Applied Econometrics, 20 (7), 873–889

11 S. Hochreiter and J. Schmidhuber: Long Short Term Memory, NeuralComput., vol. 9, no. 8, pp. 1735–1780 (1997)

12 M. Qiu, Y. Song, and F. Akagi: Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market, Chaos, Solitons and Fractals, vol. 85, pp. 1–7 (2016)

13 E. Chong, C. Han, and F. C. Park: Deep learning networks for stockmarket analysis and prediction: Methodology, data representations, andcase studies, Expert Syst. Appl., vol. 83, pp. 187–205 (2017)

14 J. Z. Wang, J. J. Wang, Z. G. Zhang, and S. P. Guo: Forecasting stock indices with back propagation neural network,Expert Syst. Appl., vol. 38,no. 11, pp. 14346–14355 (2011)

15 T. Fischer and C. Krauss: Deep learning with long short-term memorynetworks for financial market predictions, Eur. J. Oper. Res., vol. 270,no. 2, pp. 654–669, Oct. (2018)

16 J. Cao, Z. Li, and J. Li: Financial time series forecasting model based on CEEMDAN and LSTM, Phys. A Stat. Mech. its Appl., vol. 519, pp.127–139, Apr. (2019)

17 Y. Baek and H. Y. Kim: ModAugNet: A new forecasting framework for stock market index value with an over fitting prevention LSTM module and a prediction LSTM module, Expert Syst. Appl., vol. 113, pp. 457–480, Dec. (2018)

18 A. Sagheer and M. Kotb, Time series forecasting of petroleum produc-tion using deep LSTM recurrent networks, Neurocomputing, vol. 323,pp. 203–213, Jan. (2019)

19 Sagheer, Alaa and Kotb, Mostafa: Unsupervised Pre-training of a Deep LSTM-based Stacked Autoencoder for Multivariate Time Series Forecast-ing Problems, Expert Systems with Applications, Scientific Reports, Vol9, (2019)

20 Corsi F., Reno R. (2012). Discrete-time volatility forecasting with persistent leverage effect and the link with continuous-time volatility modeling. Journal of Business and Economic Statistics, 30 (3), 120–380.