

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ
имени И.Т. ТРУБИЛИНА»

Кафедра информационных систем и технологий

КУРСОВАЯ РАБОТА

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ
ТРЕХЦВЕТНОЙ КЛЕТОЧНО-АВТОМАТНОЙ
ПРОГНОЗНОЙ МОДЕЛИ

ВЫПОЛНИЛ:

магистрант направления подготовки
09.04.03 Прикладная информатика
профиль «Менеджмент проектов
в области информационных систем»
Макаров Юрий Юрьевич

РУКОВОДИТЕЛЬ:

доктор экономических наук, кандидат физико-математических наук
Попова Елена Витальевна

Краснодар 2026

РЕФЕРАТ

Работа 80 с., 10 рис., 2 табл., 7 источников, 1 прил.

КЛЕТОЧНЫЙ АВТОМАТ, ВРЕМЕННОЙ РЯД, ПРОГНОЗИРОВАНИЕ, ДОХОДНОСТЬ, ДИСКРЕТИЗАЦИЯ, СТОХАСТИЧЕСКАЯ МОДЕЛЬ, РЫНОЧНАЯ ПАМЯТЬ

Объектом исследования являются финансовые временные ряды котировок акций российских компаний.

Цель работы — разработка и программная реализация трёхцветной стохастической клеточно-автоматной модели прогнозирования финансовых временных рядов.

В процессе работы использовались методы анализа временных рядов, дискретизации доходностей, вероятностного моделирования переходов состояний, а также методы оценки точности прогноза на основе метрик MAE, MSE, RMSE и MAPE.

В результате исследования разработан алгоритм трёхсостояний клеточного автомата с памятью, реализовано программное приложение для загрузки данных с Московской биржи и проведения прогнозирования, выполнен сравнительный анализ результатов для пяти временных рядов.

Полученные результаты показали, что предложенная модель способна воспроизводить основные тенденции движения цены, при этом точность прогнозирования зависит от глубины памяти и характеристик конкретного финансового инструмента.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АНАЛИЗА ВРЕМЕННЫХ РЯДОВ	5
2 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КЛЕТОЧНЫХ АВТОМАТОВ	6
3 СТОХАСТИЧЕСКИЙ ТРЁХСОСТОЯНИЕ КЛЕТОЧНЫЙ АВТОМАТ С ПАМЯТЬЮ m	7
3.1 Обоснование выбора	7
3.2 Дискретизация доходностей	7
3.3 Память длины m	8
3.4 Вероятностное правило перехода	9
3.5 Правило прогноза	10
3.6 Переход к цене	10
4 ПОРЯДОК РАБОТЫ С ПРОГРАММОЙ	12
4.1 Подготовка данных и выбор источника	12
4.2 Настройка параметров модели	13
4.3 Выполнение прогноза	13
4.4 Оценка точности и экспорт результатов	14
4.5 Подготовка релизных сборок для запуска	17
5 ПРОГНОЗИРОВАНИЕ 5 ВРЕМЕННЫХ РЯДОВ	18
5.1 Подготовка исходных данных	18
5.2 Проведение прогнозирования	19
5.3 Формирование сравнительной таблицы	19
5.4 Анализ полученных результатов	20
ЗАКЛЮЧЕНИЕ	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	24
Приложение А	25

ВВЕДЕНИЕ

Финансовые рынки представляют собой сложные динамические системы, поведение которых изменяется во времени под воздействием множества факторов. Котировки акций характеризуются нелинейностью, высокой волатильностью и наличием случайной составляющей, что усложняет задачу их прогнозирования. В связи с этим актуальным является поиск и исследование альтернативных методов анализа временных рядов, способных учитывать вероятностный характер рыночной динамики.

Одним из таких подходов является использование клеточных автоматов - дискретных динамических систем, в которых эволюция состояния определяется локальными правилами перехода. Применение трёхсостояний модели (рост, падение, нейтральное состояние) позволяет преобразовать непрерывный временной ряд цен в дискретную последовательность состояний и построить вероятностный механизм прогнозирования.

Целью данной курсовой работы является разработка и программная реализация трёхцветной стохастической клеточно-автоматной модели для прогнозирования финансовых временных рядов.

Для достижения поставленной цели были решены следующие задачи: изучены теоретические основы анализа временных рядов и клеточных автоматов, реализован алгоритм модели с памятью, проведено прогнозирование на основе данных Московской биржи и выполнен сравнительный анализ результатов для пяти акций российских компаний.

Объектом исследования являются временные ряды котировок акций, предметом - клеточно-автоматная модель их прогнозирования.

1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АНАЛИЗА ВРЕМЕННЫХ РЯДОВ

В рамках дисциплины “Нелинейная динамика экономических процессов” объектом исследования выступают временные ряды котировок акций.

Временной ряд представляет собой последовательность значений показателя, упорядоченных по времени и полученных через равные промежутки времени.

Финансовые временные ряды представляют собой нелинейные, нестационарные динамические системы, обладающие шумовой компонентой и эффектом кластеризации волатильности.

Формально временной ряд может быть представлен как $x_t = f(x_{t-1}, \dots, x_{t-m}) + \varepsilon_t$, где f - нелинейная функция, m - глубина памяти, ε_t - случайная ошибка.

2 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КЛЕТОЧНЫХ АВТОМАТОВ

Клеточный автомат - дискретная динамическая система, определяемая четверкой $CA = (L, S, N, F)$, где L - множество клеток, S - конечное множество состояний, N - окрестность, F - функция перехода.

Эволюция описывается соотношением:

$$s_i^{t+1} = F(s_j^t, j \in N(i))$$

Клеточные автоматы классифицируются по размерности (1D, 2D), по числу состояний (двух-, трёх-, многосостояние) и по типу правила (детерминированные, стохастические, с памятью).

В классическом одномерном клеточном автомате правило фиксировано:

$$s_i^{t+1} = F(s_{i-1}^t, s_i^t, s_{i+1}^t)$$

В данной работе используется трёхсостояние автомат с множеством состояний $S = \{-1, 0, 1\}$, интерпретируемых как рост, стабильность и падение.

Переход определяется временной функцией:

$$s_t = F(s_{t-1}, \dots, s_{t-m}).$$

3 СТОХАСТИЧЕСКИЙ ТРЁХСОСТОЯНИЕ КЛЕТОЧНЫЙ АВТОМАТ С ПАМЯТЬЮ m

3.1 Обоснование выбора

В предыдущих разделах было показано, что финансовые временные ряды характеризуются высокой степенью неопределённости и наличием случайной составляющей. Поведение рынка нельзя считать полностью детерминированным, поскольку на динамику цен влияет большое количество факторов: экономические новости, макроэкономические показатели, поведение инвесторов и спекулятивная активность.

Поэтому использование полностью детерминированного клеточного автомата может быть недостаточно гибким. В связи с этим в работе применяется стохастическая модификация клеточного автомата, в которой правило перехода между состояниями формируется на основе вероятностей.

Иными словами, модель не жёстко задаёт следующее состояние, а выбирает его на основе статистики наблюдений в исторических данных.

3.2 Дискретизация доходностей

На первом этапе выполняется преобразование исходного временного ряда цен в ряд доходностей.

Доходность рассчитывается по формуле:

$$r_t = \ln(x_t / x_{t-1})$$

x_t - цена акции в текущий момент времени t

x_{t-1} - цена акции в предыдущий момент времени $t - 1$

r_t - относительное изменение цены (доходность)

Экономический смысл данной формулы заключается в том, что она показывает, на сколько процентов изменилась цена по сравнению с предыдущим значением.

После вычисления доходностей выполняется их дискретизация, то есть переход от непрерывных значений к трём фиксированным состояниям.

Вводится пороговое значение k , которое определяет границу между значимым и незначительным изменением.

Состояние определяется следующей системой уравнений:

$$s_t = 1 \text{ при } r_{it} > k$$

$$s_t = -1 \text{ при } r_t < -k$$

$$s_t = 0 \text{ при } |r_{it}| \leq k$$

s_t - состояние системы в момент времени t

k - порог дискретизации

Экономическая интерпретация:

1 - существенный рост;

-1 - существенное падение;

0 - незначительное изменение (боковое движение).

Таким образом, исходный временной ряд преобразуется в последовательность из трёх состояний.

3.3 Память длины m

Следующим важным параметром модели является глубина памяти m .

Под глубиной памяти понимается количество предыдущих состояний, которые учитываются при прогнозировании следующего состояния.

Формируется шаблон:

$$\text{parent}_t = (s_{t-m}, t_{t-m+1}, \dots, s_{t-1})$$

parent_t - последовательность предыдущих m состояний;

s_{t-1} - последнее известное состояние.

s_{t-m} - состояние системы m шагов назад

Количество возможных шаблонов определяется как 3^m

Это связано с тем, что каждое состояние может принимать три значения (-1, 0, 1), и для каждого из m шагов существует три варианта.

Например:

при $m=2$ $3^m = 2^m = 2$ возможных комбинаций 9;

при $m=3$ $3^m = 3^m = 3$ - уже 27.

Таким образом, увеличение глубины памяти значительно увеличивает количество возможных переходов.

3.4 Вероятностное правило перехода

После формирования всех шаблонов производится подсчёт частоты появления следующего состояния.

Для каждого шаблона определяется вероятность перехода:

$$P(s \mid \text{pattern}) = (\text{count}_s + a) / N + 3a$$

count_s - количество случаев, когда после данного шаблона наблюдалось состояние s

a - параметр Лапласовского сглаживания

$P(s \mid \text{pattern})$ - условная вероятность перехода к состоянию s при заданном шаблоне

Параметр α используется для того, чтобы избежать ситуации нулевых вероятностей. Это особенно важно при малом количестве наблюдений.

Проще говоря, если некоторый переход не встречался в обучающих данных, модель всё равно оставляет ему небольшую вероятность.

3.5 Правило прогноза

После вычисления вероятностей для каждого возможного состояния выбирается наиболее вероятное:

$$\tilde{s}_{t+1} = \arg \max P(s \mid \text{pattern})$$

\tilde{s}_{t+1} - прогнозируемое состояние на следующий момент времени

$\arg \max$ - операция выбора значения аргумента, при котором функция вероятности принимает максимальное значение

$P(s \mid \text{pattern})$ - условная вероятность перехода к состоянию s при заданном шаблоне предыдущих состояний

То есть модель выбирает то состояние, которое чаще всего следовало за данной комбинацией состояний в прошлом.

3.6 Переход к цене

После того как определено прогнозное состояние \tilde{s}_{t+1} , необходимо получить числовой прогноз цены.

Для этого для каждого состояния рассчитывается средняя доходность:

$$\mu_s = 1/n_s \sum r_t$$

μ_s - средняя доходность для состояния s

N_s - количество наблюдений этого состояния.

Прогнозная доходность в момент времени t равна средней доходности того состояния, которое спрогнозировала модель:

$$r_t = \mu_s * s$$

s - состояние клеточного автомата (-1 , 0 или $+1$), выбранное как наиболее вероятное.

μ_s - средняя доходность для состояния s

r_t - прогнозная доходность

Далее прогнозная цена определяется по формуле:

$$\tilde{x}_{t+1} = \tilde{x}_t * e^{r_t}$$

\tilde{x}_t - текущая прогнозная цена;

\tilde{x}_{t+1} - следующая прогнозируемая цена;

r_t - прогнозная доходность

При первой итерации \tilde{x}_t берется как реальная цена, а не прогнозируемая.

Таким образом, модель сначала прогнозирует направление движения, а затем переводит это направление в числовой прогноз с использованием соответствующей средней доходности.

4 ПОРЯДОК РАБОТЫ С ПРОГРАММОЙ

4.1 Подготовка данных и выбор источника

Программа поддерживает два источника временных рядов: импорт CSV файла и загрузку котировок через MOEX ISS API (API московской биржи). Перед запуском прогнозирования пользователь выбирает источник данных, задаёт диапазон дат (для API) либо указывает файл CSV. После загрузки данные приводятся к единому формату (дата–цена закрытия) и сортируются по времени.

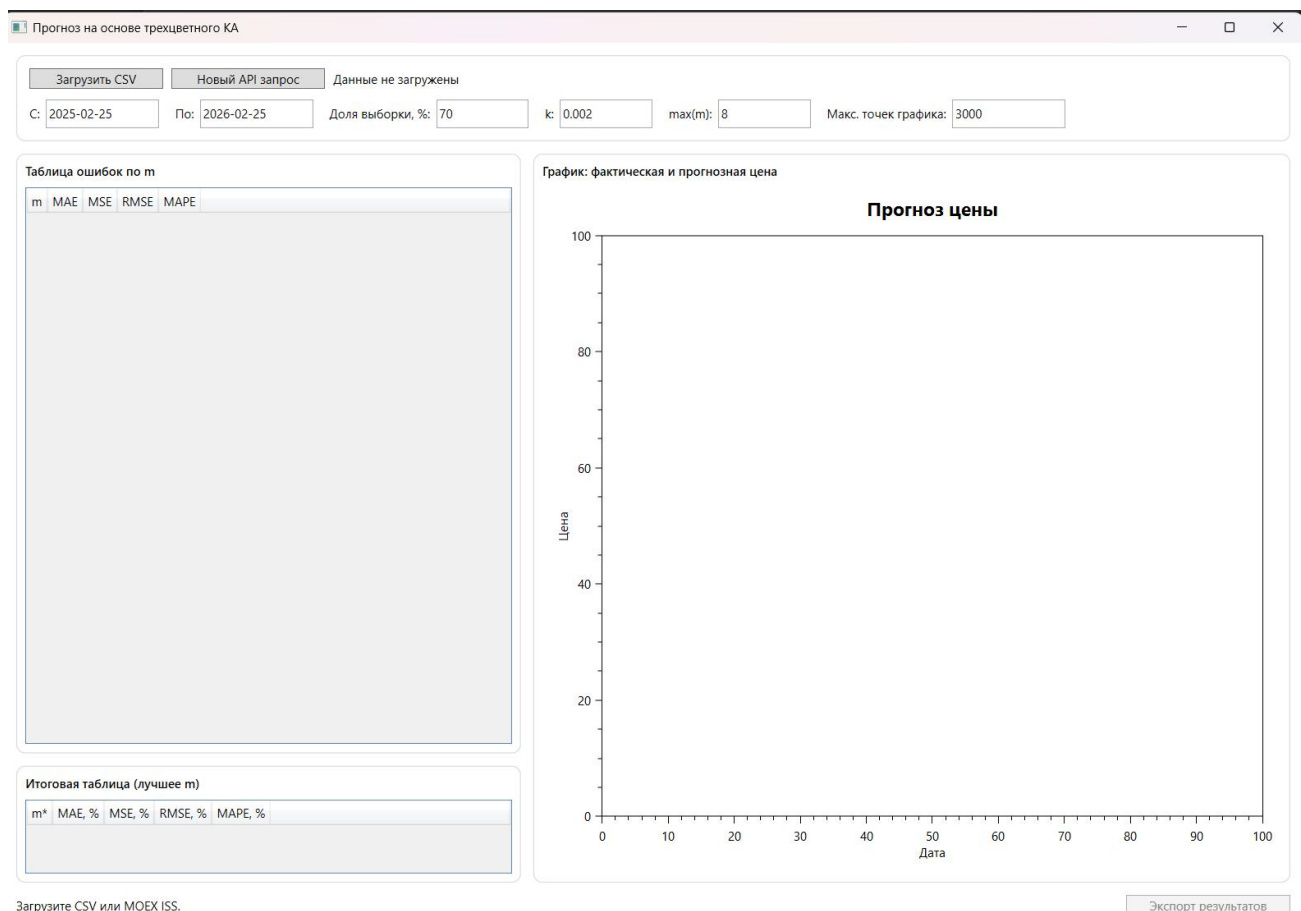


Рисунок 1 - Интерфейс приложения

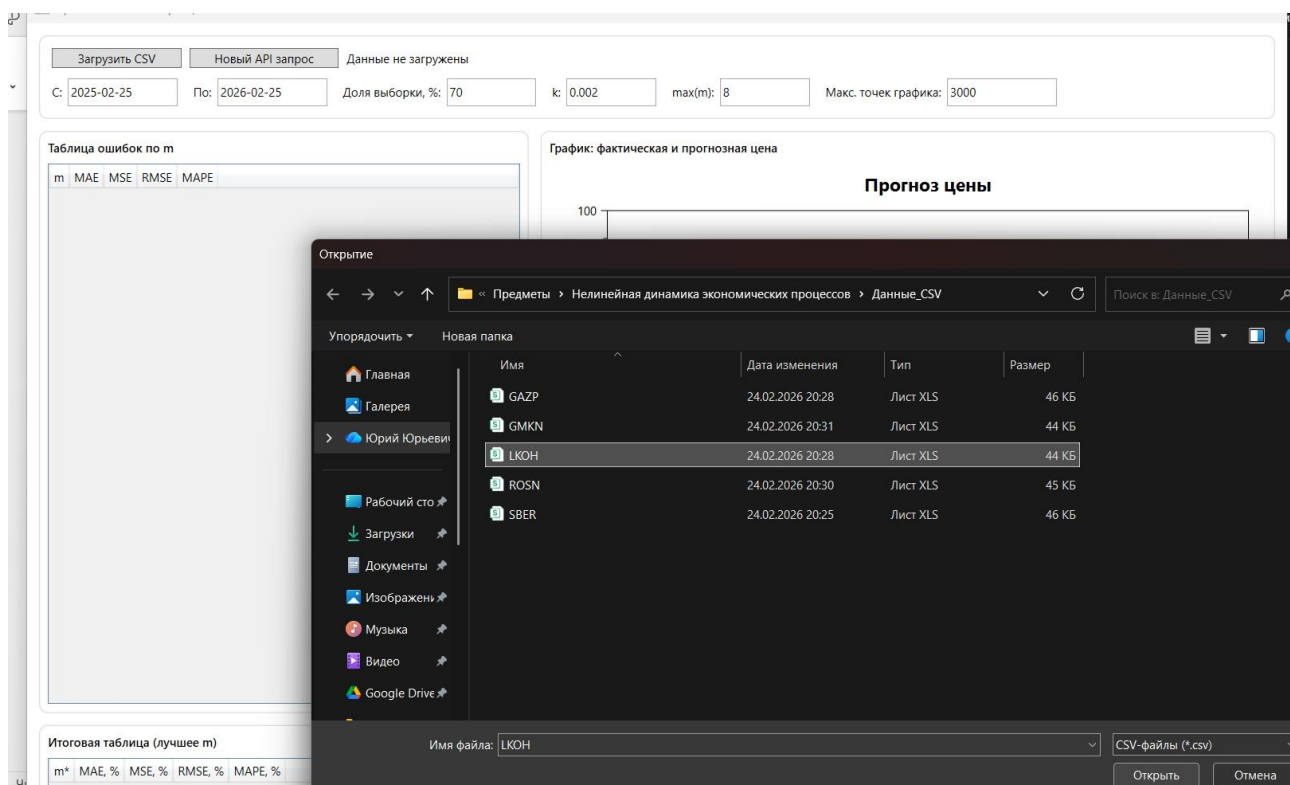


Рисунок 2 - Импорт данных

4.2 Настройка параметров модели

Далее задаются ключевые параметры трёхсостояний модели: порог дискретизации k , глубина памяти m и параметр Лапласовского сглаживания a . Порог k определяет, какие изменения доходности относятся к росту, падению или нейтральному состоянию; параметр m задаёт длину временного контекста (шаблона), по которому оценивается вероятностный переход

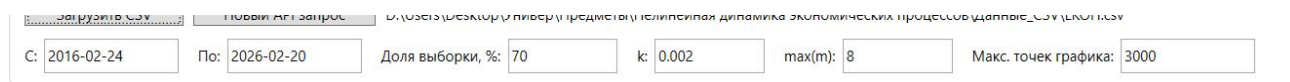


Рисунок 3 - Установка значений ключевых параметров

4.3 Выполнение прогноза

После нажатия кнопки запуска выполняются последовательные этапы:

1) расчёт доходностей r_t по исходным ценам;

- 2) кодирование доходностей в состояния $s_t \in \{-1, 0, 1\}$;
- 3) формирование обучающей выборки шаблонов длины m и оценка частот переходов;
- 4) получение вероятностей со сглаживанием;
- 5) выбор наиболее вероятного состояния на каждом шаге и перевод состояния в прогнозируемую доходность по средним;
- 6) рекурсивное восстановление ценового прогноза.

Результат отображается на графике, где сравниваются фактические и прогнозные значения.

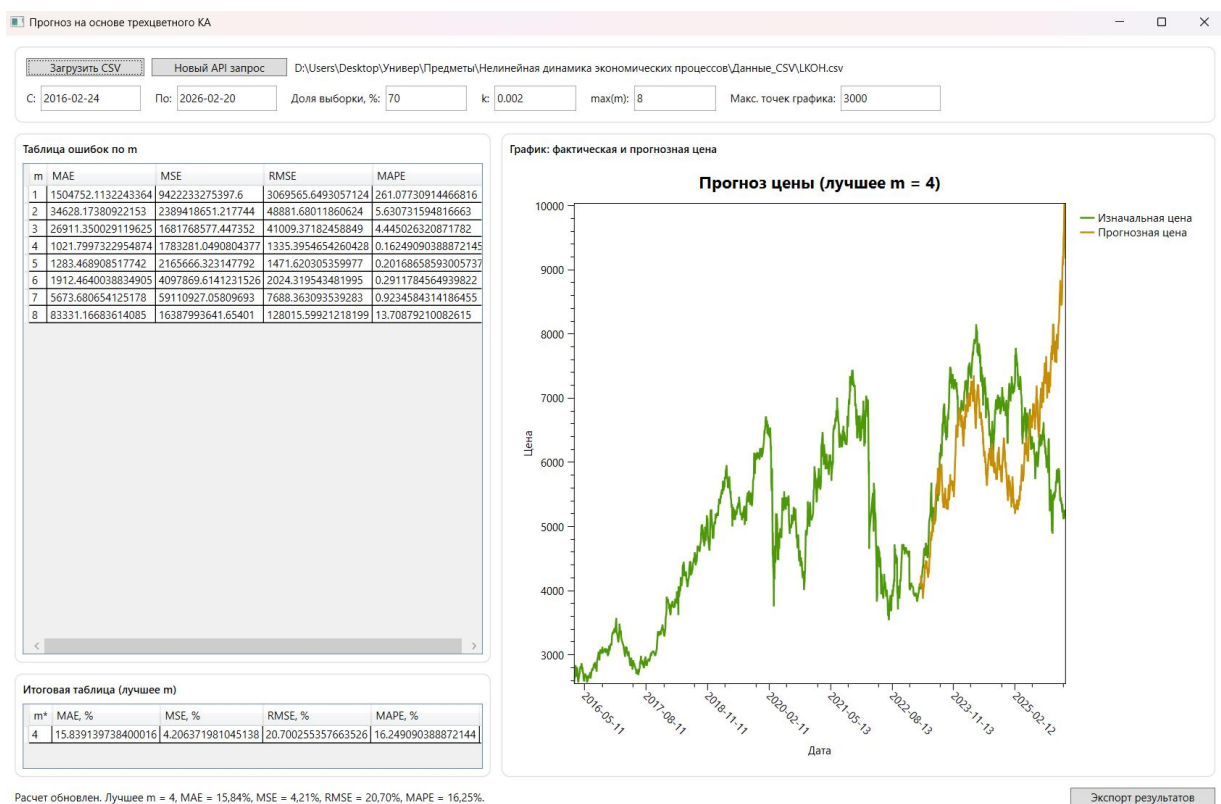


Рисунок 4 - Результаты прогноза

4.4 Оценка точности и экспорт результатов

Точность прогноза оценивается по метрикам MAE, MSE, RMSE и MAPE. После расчёта метрик пользователь может сохранить результаты в CSV-файл: прогнозные цены по датам, значения ошибок и параметры модели. Это обеспечивает воспроизводимость экспериментов и возможность сравнения результатов для разных временных рядов.

Расчет обновлен. Лучшее $m = 4$, MAE = 15,84%, MSE = 4,21%, RMSE = 20,70%, MAPE = 16,25%.

Рисунок 5 - Краткая сводка результатов по ошибкам прогноза

Таблица ошибок по m				
m	MAE	MSE	RMSE	MAPE
1	1504752.1132243364	9422233275397.6	3069565.6493057124	261.07730914466816
2	34628.17380922153	2389418651.217744	48881.68011860624	5.630731594816663
3	26911.350029119625	1681768577.447352	41009.37182458849	4.445026320871782
4	1021.7997322954874	1783281.0490804377	1335.3954654260428	0.16249090388872145
5	1283.468908517742	2165666.323147792	1471.620305359977	0.20168658593005737
6	1912.4640038834905	4097869.6141231526	2024.319543481995	0.2911784564939822
7	5673.680654125178	59110927.05809693	7688.363093539283	0.9234584314186455
8	83331.16683614085	16387993641.65401	128015.59921218199	13.70879210082615

Итоговая таблица (лучшее m)				
m^*	MAE, %	MSE, %	RMSE, %	MAPE, %
4	15.839139738400016	4.206371981045138	20.700255357663526	16.249090388872144

Рисунок 6 - Подробная сводка результатов по ошибкам прогноза

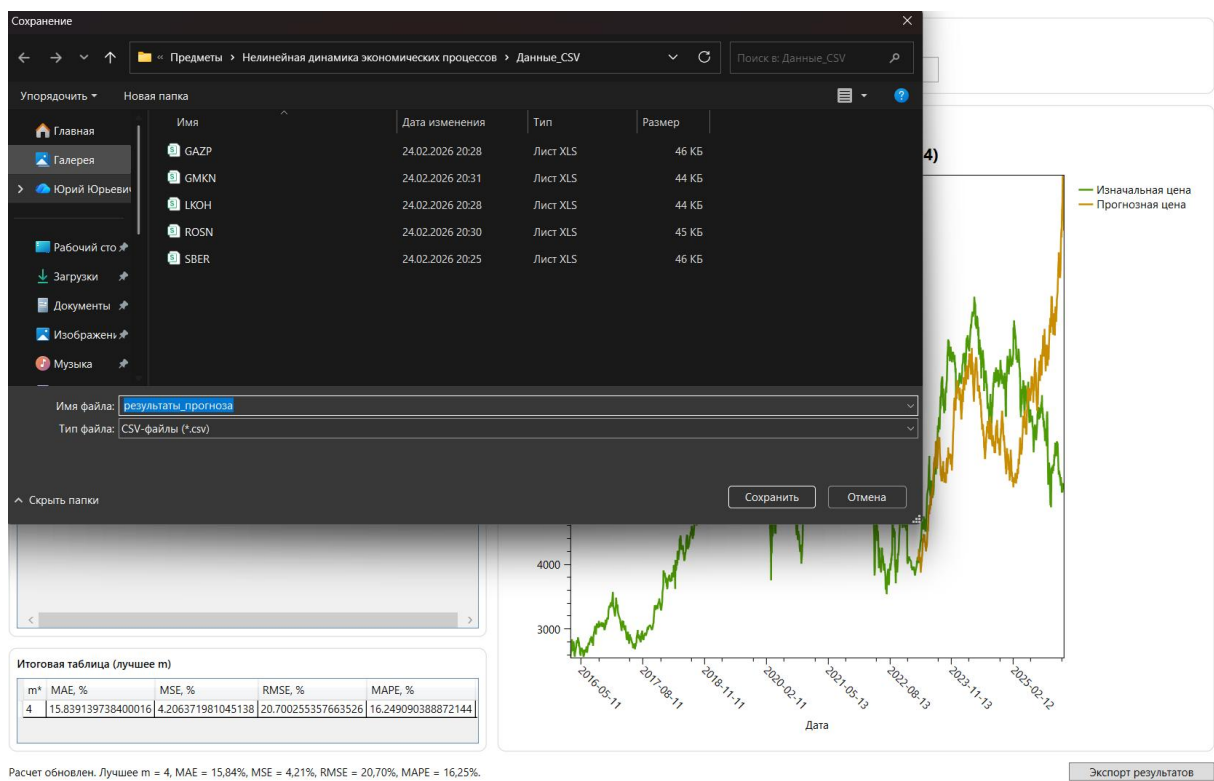


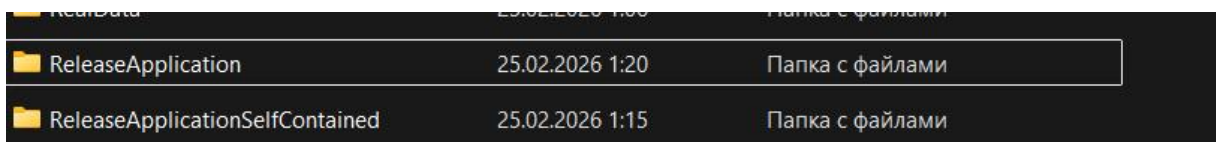
Рисунок 7 - Выбор сохранения экспорт файла

	A	B	C	D	E
1	ЛучшаяПамять	4			
2					
3	Метрики				
4	Память	MAE	MSE	RMSE	MAPE
5		1 1504752,1	9,422E+12	3069565,6	261,07731
6		2 34628,174	2,389E+09	48881,68	5,6307316
7		3 26911,35	1,682E+09	41009,372	4,4450263
8		4 1021,7997	1783281	1335,3955	0,1624909
9		5 1283,4689	2165666,3	1471,6203	0,2016866
10		6 1912,464	4097869,6	2024,3195	0,2911785
11		7 5673,6807	59110927	7688,3631	0,9234584
12		8 83331,167	1,639E+10	128015,6	13,708792
13					
14	Прогнозная цена				
15	Дата	Изначальн	Прогнозная цена		
16	09.03.2023	4105	4063,6362		
17	10.03.2023	4085	4122,9167		
18	13.03.2023	4066	4061,5838		
19	14.03.2023	4063	4120,8344		
20	15.03.2023	4022	4180,9492		
21	16.03.2023	4084	4241,941		

Рисунок 8 - Данные в экспорт файле

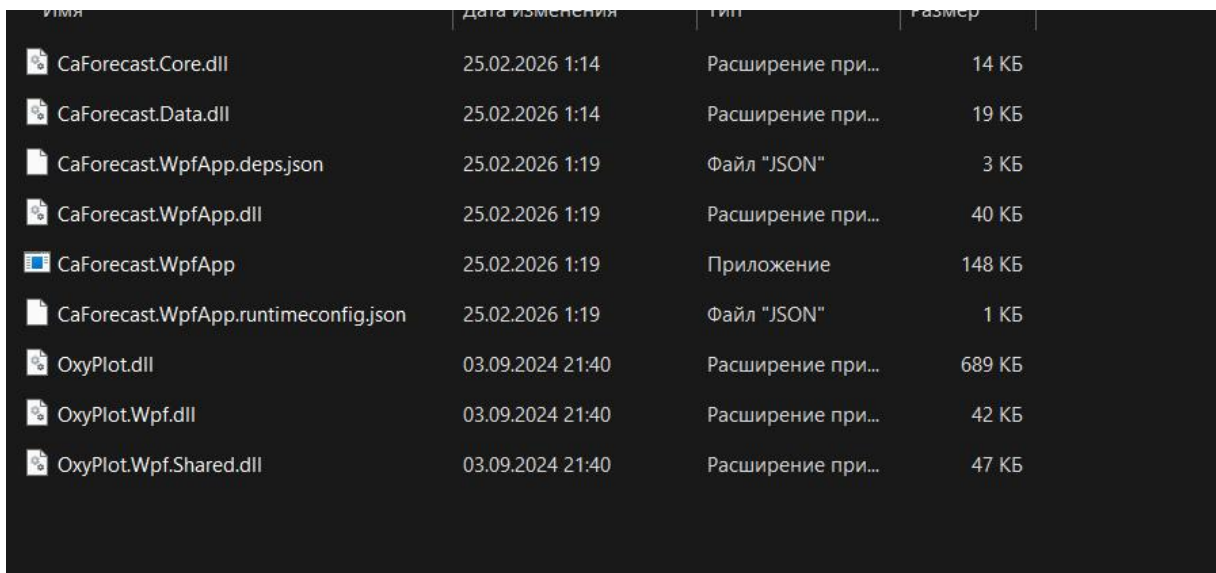
4.5 Подготовка релизных сборок для запуска

Для распространения приложения сформированы две релизные сборки: framework-dependent (требует установленного .NET 8 Runtime на целевой машине) и self-contained (содержит все необходимые компоненты платформы и не требует установки .NET). На рисунке 6.1 показана структура каталогов релизных сборок, а на рисунке 6.2 - состав framework-dependent сборки.



ReleaseApplication	25.02.2026 1:20	Папка с файлами
ReleaseApplicationSelfContained	25.02.2026 1:15	Папка с файлами

Рисунок 9 - Структура каталогов релиза



Имя	дата изменения	тип	размер
CaForecast.Core.dll	25.02.2026 1:14	Расширение при...	14 КБ
CaForecast.Data.dll	25.02.2026 1:14	Расширение при...	19 КБ
CaForecast.WpfApp.deps.json	25.02.2026 1:19	Файл "JSON"	3 КБ
CaForecast.WpfApp.dll	25.02.2026 1:19	Расширение при...	40 КБ
CaForecast.WpfApp	25.02.2026 1:19	Приложение	148 КБ
CaForecast.WpfApp.runtimeconfig.json	25.02.2026 1:19	Файл "JSON"	1 КБ
OxyPlot.dll	03.09.2024 21:40	Расширение при...	689 КБ
OxyPlot.Wpf.dll	03.09.2024 21:40	Расширение при...	42 КБ
OxyPlot.Wpf.Shared.dll	03.09.2024 21:40	Расширение при...	47 КБ

Рисунок 10 - Файлы релиза

5 ПРОГНОЗИРОВАНИЕ 5 ВРЕМЕННЫХ РЯДОВ

5.1 Подготовка исходных данных

Практическая часть работы началась с формирования набора временных рядов для последующего прогнозирования. В качестве источника данных использовалась информационная система Московской биржи (MOEX ISS API), позволяющая выгружать исторические котировки акций в формате CSV/XLS.

Были выбраны пять ликвидных российских компаний:

- 1) GAZP - ПАО “Газпром” (газовая отрасль)
- 2) GMKN - ПАО «ГМК “Норильский никель” (цветная металлургия)
- 3) LKOH - ПАО “ЛУКОЙЛ” (нефтегазовая отрасль)
- 4) ROSN - ПАО “Роснефть” (нефтяная отрасль)
- 5) SBER - ПАО “Сбербанк России” (финансовый сектор)

Для каждой компании был сформирован временной ряд цен закрытия за длительный период наблюдений.

Использованный диапазон дат представлен в таблице:

Ряд	Диапазон дат
GAZP	24.02.2016 — 20.02.2026
GMKN	24.02.2016 — 01.04.2024 (до дробления акций)
LKOH	24.02.2016 — 20.02.2026
ROSN	24.02.2016 — 20.02.2026
SBER	24.02.2016 — 20.02.2026

Таблица 1 - Диапазон дат для расчётов

Для GMKN период был ограничен датой дробления акций, чтобы избежать структурного разрыва ряда и некорректного влияния на модель.

После выгрузки данные были сохранены в отдельных файлах и загружены в разработанное программное приложение.

5.2 Проведение прогнозирования

Далее для каждого временного ряда выполнялись одинаковые этапы обработки:

- 1) Загрузка исходного файла в программу
- 2) Расчёт доходностей r_t
- 3) Дискретизация доходностей в трёхсостояний ряд $st \in \{1, 0, -1\}$
- 4) Формирование шаблонов длины m
- 5) Оценка вероятностей переходов с использованием Лапласовского сглаживания.
- 6) Подбор оптимальной глубины памяти mmm .
- 7) Расчёт прогнозной цены.
- 8) Вычисление метрик точности.

Для каждого ряда программа автоматически перебирала различные значения глубины памяти и определяла то значение m , при котором ошибка прогнозирования была минимальной.

После выполнения расчётов результаты сохранялись в отдельные экспортные файлы:

- 1) GAZP_export
- 2) GMKN_export
- 3) LKOH_export
- 4) ROSN_export
- 5) SBER_export

5.3 Формирование сравнительной таблицы

На основе экспортированных данных вручную была составлена итоговая сравнительная таблица.

Ряд	Лучшая память m	MAE	MSE	RMSE	MAPE (%)
GAZP	4	20.45	760.08	27.57	15.53
GMKN	6	3100.77	18314899.93	4279.59	20.24
LKOH	4	1021.80	1783281.05	1335.40	16.25
ROSN	6	83.43	10843.10	104.13	16.14
SBER	4	77.97	9092.31	95.35	26.89

Таблица 2 - сравнительная таблица 5 компаний

5.4 Анализ полученных результатов

Оптимальная глубина памяти различается для разных акций.

Для GAZP, LKOH и SBER оптимальным оказалось $m = 4$, тогда как для GMKN и ROSN - $m = 6$. Это говорит о том, что разные финансовые инструменты обладают различной глубиной “рыночной памяти”.

Наименьшие относительные ошибки (MAPE) показали GAZP и ROSN (около 15–16%). Это свидетельствует о том, что модель лучше адаптируется к более стабильным рядам.

Для GMKN наблюдается высокая абсолютная ошибка, что связано с высокой стоимостью акции и значительной волатильностью.

Наибольшая относительная ошибка наблюдается для SBER, что может быть связано с повышенной чувствительностью акции к рыночным событиям.

Увеличение глубины памяти выше оптимального значения приводило к росту ошибки, что подтверждает эффект переобучения при чрезмерно большом числе шаблонов 3^m .

В целом результаты показывают, что трёхцветный стохастический клеточный автомат способен воспроизводить основные тенденции движения цены, однако точность прогноза зависит от характера конкретного временного ряда.

ЗАКЛЮЧЕНИЕ

В рамках данной курсовой работы была разработана и исследована трёхцветная стохастическая клеточно-автоматная модель прогнозирования финансовых временных рядов.

В теоретической части были рассмотрены особенности финансовых временных рядов как нелинейных и нестационарных динамических систем, а также основные принципы построения клеточных автоматов. Показано, что использование дискретной модели с ограниченным числом состояний позволяет упростить структуру анализа и перейти от непрерывных ценовых значений к последовательности состояний, отражающих направление движения рынка.

В ходе работы была реализована программная система, обеспечивающая:

- загрузку исторических котировок через MOEX ISS API или из CSV-файлов;
- преобразование цен в доходности;
- дискретизацию доходностей в трёхсостояний ряд;
- построение вероятностной модели переходов с использованием Лапласовского сглаживания;
- подбор оптимальной глубины памяти;
- вычисление прогнозных значений цены;
- расчёт метрик точности и экспорт результатов.

Практическое исследование было проведено на пяти временных рядах российских компаний: ПАО “Газпром”, ПАО «ГМК „Норильский никель“», ПАО “ЛУКОЙЛ”, ПАО “Роснефть” и ПАО “Сбербанк России”. Для каждого инструмента определена оптимальная глубина памяти модели и рассчитаны показатели точности прогнозирования.

Результаты показали, что оптимальная глубина памяти различается для разных финансовых инструментов, что свидетельствует о различной степени временной зависимости их динамики. Это подтверждает гипотезу о наличии различной «рыночной памяти» у исследуемых активов.

Наименьшие относительные ошибки продемонстрировали ряды с более устойчивой динамикой, тогда как более волатильные инструменты показали большую погрешность прогнозирования. Также было установлено, что чрезмерное увеличение глубины памяти приводит к росту ошибки, что связано с экспоненциальным увеличением числа шаблонов $3m^3$ и эффектом переобучения модели.

В целом можно сделать вывод, что предложенная трёхцветная клеточно-автоматная модель способна воспроизводить основные тенденции движения финансовых временных рядов и может использоваться в качестве инструмента анализа динамики котировок. При этом точность прогноза зависит от характера исследуемого актива и выбранных параметров модели.

Дальнейшее развитие работы может быть связано с:

- расширением числа состояний модели;
- адаптивным выбором порога дискретизации;
- использованием скользящего окна обучения;
- сравнением с нейросетевыми и классическими эконометрическими методами.

Таким образом, поставленные в работе цели и задачи были достигнуты, а разработанная программная реализация продемонстрировала практическую применимость предложенного подхода.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Тюрин, Ю.Н. Анализ данных на компьютере: учеб. пособие / Ю.Н. Тюрин, А.А. Макаров – Москва: ИНФРА-М, 2002. – 528 с.
2. Нильсен, Э. Практический анализ временных рядов: прогнозирование со статистикой и машинным обучением / Э. Нильсен – Москва: ДМК Пресс, 2021. – 544 с.
3. Воривода, М.А. Клеточные автоматы / М.А. Воривода, О.А. Григорьева // 58-я Научная конференция аспирантов, магистрантов и студентов БГУИР – Минск, 2022. – С. 117–119.
4. Эконометрика: учебник для студентов вузов / Н.Ш. Кремер, Б.А. Путко – Москва: ЮНИТИ-ДАНА, 2010. – 328 с.
5. Воскобойников, Ю. Е. Эконометрика в Excel. Часть 2. Анализ временных рядов: учеб. пособие / Ю. Е. Воскобойников – Новосибирск: НГАСУ (Сибстрин), 2008. – 155 с. 3.
6. Курс лекций по экономико-математическому моделированию / И.В. Орлова, В.А. Половников, В.В. Федосеев – Москва: ВЗФИ, 1993-148 с.
7. Компьютерное экономико-математическое моделирование: Учебное пособие / А.А. Горчаков, И.В. Орлова – Москва: ЮНИТИ, 1995-170 с.

Приложение А

Фрагмент кода программы

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace CaForecast.Core;

public class CaForecaster
{
    public ForecastResult Forecast(
        IReadOnlyList<double> prices,
        IReadOnlyList<double> returns,
        IReadOnlyList<int> encodedStates,
        int trainReturnsCount,
        int memory,
        double alpha,
        CaRuleTrainer trainer,
        MetricsService metricsService)
    {
        if (prices is null)
        {
            throw new ArgumentNullException(nameof(prices));
        }

        if (returns is null)
        {
            throw new ArgumentNullException(nameof(returns));
        }

        if (encodedStates is null)
        {
            throw new ArgumentNullException(nameof(encodedStates));
        }

        if (trainer is null)
        {
            throw new ArgumentNullException(nameof(trainer));
        }

        if (metricsService is null)
        {
            throw new ArgumentNullException(nameof(metricsService));
        }

        if (returns.Count != encodedStates.Count)
        {
            throw new ArgumentException("Длины массивов доходностей и закодированных состояний должны совпадать.");
        }

        if (trainReturnsCount <= memory || trainReturnsCount >= returns.Count)
        {
            throw new ArgumentOutOfRangeException(nameof(trainReturnsCount), "Размер обучающей выборки должен быть больше памяти и меньше общего числа доходностей.");
        }
    }
}
```

```

var trainStates = encodedStates.Take(trainReturnsCount).ToArray();
var trainReturns = returns.Take(trainReturnsCount).ToArray();
var model = trainer.Train(trainStates, trainReturns, memory, alpha);

var actualReturns = new List<double>();
var predictedReturns = new List<double>();
var actualPrices = new List<double>();
var predictedPrices = new List<double>();

var currentPredictedPrice = prices[trainReturnsCount];

for (var t = trainReturnsCount; t < returns.Count; t++)
{
    var pattern = BuildPattern(encodedStates, t - memory, memory);
    var probabilities = model.TransitionProbabilities.TryGetValue(pattern, out var p)
        ? p
        : model.GlobalDistribution;

    var predictedState = ArgMaxState(probabilities);
    var predictedReturn = model.MeanReturnsByState[predictedState];

    actualReturns.Add(returns[t]);
    predictedReturns.Add(predictedReturn);

    var actualPrice = prices[t + 1];
    actualPrices.Add(actualPrice);

    currentPredictedPrice *= Math.Exp(predictedReturn);
    predictedPrices.Add(currentPredictedPrice);
}

var mae = metricsService.CalculateMae(actualPrices, predictedPrices);
var mse = metricsService.CalculateMse(actualPrices, predictedPrices);
var rmse = metricsService.CalculateRmse(actualPrices, predictedPrices);
var mapePercent = metricsService.CalculateMapePercent(actualPrices, predictedPrices);

return new ForecastResult
{
    Memory = memory,
    TrainReturnsCount = trainReturnsCount,
    ActualReturns = actualReturns,
    PredictedReturns = predictedReturns,
    ActualPrices = actualPrices,
    PredictedPrices = predictedPrices,
    Mae = mae,
    Mse = mse,
    Rmse = rmse,
    MapePercent = mapePercent
};
}

private static string BuildPattern(IReadOnlyList<int> states, int start, int memory)
{
    return string.Join('|', states.Skip(start).Take(memory));
}

private static int ArgMaxState(IReadOnlyList<double> probs)
{
    var index = 0;

```

```

var max = probs[0];
for (var i = 1; i < probs.Count; i++)
{
    if (probs[i] > max)
    {
        max = probs[i];
        index = i;
    }
}

return index switch
{
    0 => -1,
    1 => 0,
    2 => 1,
    _ => 0
};
}
}

```