

DOI: 10.26794/2587-5671-2020-24-1-14-23

УДК 330.43(045)

JEL C01

## Сравнительный анализ прогнозных моделей ARIMA и LSTM на примере акций российских компаний

А.В. Алжеев<sup>а</sup>, Р.А. Кочкаров<sup>б</sup>

Финансовый университет, Москва, Россия

<sup>а</sup> <https://orcid.org/0000-0001-8944-5679>; <sup>б</sup> <https://orcid.org/0000-0003-3186-3901>

### АННОТАЦИЯ

**Цель** статьи – поиск лучшей модели для прогноза временных рядов с учетом минимизации ошибок и высокой точности прогноза. Использован **метод** сравнительного анализа наиболее популярной «традиционной» эконометрической модели ARIMA и модели глубокого обучения LSTM (Long short-term memory) на основе рекуррентной нейронной сети. Приведено математическое описание этих прогнозных моделей. Авторы разработали алгоритмы для прогноза временных рядов, основанные на подходе “Rolling forecasting origin” («прогнозирование происхождения»). Алгоритмы реализованы в среде программирования Python с подключенными библиотеками Keras, Theano и Statsmodels. В качестве входных наборов данных импортированы значения котировок акций российских компаний: Алроса, Газпром, КамАЗ, НЛМК, Киви, Роснефть, ВТБ и Яндекс за период с 02.06.2014 по 11.11.2019 г. с разбивкой по неделям. **Результаты** исследования подтверждают превосходство модели LSTM, при которой среднеквадратическая ошибка RMSE на 65% меньше, чем при использовании модели ARIMA. Сделан **вывод**, что для повышения качества прогноза временных рядов предпочтительно применять алгоритм на основе модели LSTM.

**Ключевые слова:** ARIMA; LSTM; прогнозные модели; акции; анализ; прогнозирование котировок; алгоритмы

**Для цитирования:** Алжеев А.В., Кочкаров Р.А. Сравнительный анализ прогнозных моделей ARIMA и LSTM на примере акций российских компаний. *Финансы: теория и практика*. 2020;24(1):14-23. DOI: 10.26794/2587-5671-2020-24-1-14-23

## Comparative Analysis of ARIMA and LSTM Predictive Models: Evidence from Russian Stocks

A.V. Alzheev<sup>а</sup>, R.A. Kochkarov<sup>б</sup>

Financial University, Moscow, Russia

<sup>а</sup> <https://orcid.org/0000-0001-8944-5679>; <sup>б</sup> <https://orcid.org/0000-0003-3186-3901>

### ABSTRACT

The article **aims** to find the best time series predictive model, considering the minimization of errors and high accuracy of the prediction. The authors performed the comparative analysis of the most popular “traditional” econometric model ARIMA and the deep learning model LSTM (Long short-term memory) based on a recurrent neural network. The study provides a mathematical description of these predictive models. The authors developed algorithms for predicting time series based on the “Rolling forecasting origin” approach. These are Python-based algorithms using the Keras, Theano and Statsmodels libraries. Stock quotes of Russian companies Alrosa, Gazprom, KamAZ, NLMK, Kiwi, Rosneft, VTB and Yandex for the period from June 2, 2014 to November 11, 2019, broken down by week, served as input data. The research **results** confirm the superiority of the LSTM model, where the RMSE error is 65% less than with the ARIMA model. Therefore, an LSTM model-based algorithm is more preferable for the better quality of time series prediction.

**Keywords:** ARIMA; LSTM; predictive models; stocks; analysis; stock quote prediction; algorithms

**For citation:** Alzheev A.V., Kochkarov R.A. Comparative analysis of ARIMA and LSTM predictive models: Evidence from Russian stocks. *Finance: Theory and Practice*. 2020;24(1):14-23. (In Russ.). DOI: 10.26794/2587-5671-2020-24-1-14-23

## ВВЕДЕНИЕ

Прогнозирование временных рядов — сложная задача, во многом обусловленная неожиданными изменениями в экономике и асимметрией информации. Из-за волатильности рынка в последнее время серьезную обеспокоенность вызывает качество прогнозов регрессионных моделей, которое необходимо тщательно оценивать.

Временной ряд — это последовательность значений наблюдений какой-либо величины, взятых через относительно равные промежутки времени. Примерами временного ряда могут быть: объем выпуска продукции, число проданных страховых продуктов, наводнения и т.п. Временные ряды широко используются в статистике, обработке сигналов, распознавании образов, эконометрике, финансах, астрономии, инженерии и в большинстве других видов человеческой деятельности, где измеряются значения в течение времени. Анализ временных рядов объединяет в себе ряд моделей для извлечения значимой информации из данных с целью предсказывать будущие значения, основываясь на прошлых.

Временной ряд всегда является упорядоченным во времени. Это отличает анализ временного ряда от поперечных исследований (к примеру, связь между полом человека и заработной платой). Также имеются отличия и от исследования пространственных выборок (зависимость заработной платы от региона). Модели временных рядов отражают факт того, что близкие во времени наблюдения будут связаны сильнее, чем наблюдения, более далекие друг от друга во времени.

Целью данной работы является поиск адекватной модели для прогноза временных рядов с учетом минимизации ошибок и высокой точности.

Одной из распространенных и часто применяемых в контексте прогнозирования финансовых временных рядов является модель ARIMA (от англ. Autoregressive moving average — интегрированная модель авторегрессии скользящего среднего, иногда ее называют моделью Бокса-Дженкинса) [1]. Популярными моделями глубокого обучения являются RNN (от англ. Recurrent neural network — рекуррентная нейронная сеть) [2] и ее модификация LSTM (от англ. Long short-term memory — сеть долгой краткосрочной памяти), предложенная в 1997 г. З. Хохрайтером и Ю. Шмидхубером [3]. Глубокое обучение позволяет находить сложные закономерности в последовательных пространственных цепочках. Рекуррентная нейронная сеть, например, применяется для распознавания рукописного текста и речи. LSTM модель, в частности, была использована для предсказания волатильности индекса S&P 500 [4–6].

В данной статье сравниваются модели ARIMA и LSTM, а критерием сравнения является минимизация ошибки прогноза. Модель ARIMA выбрана, исходя из ее способности работать с данными нестационарного характера. Из спектра моделей глубокого обучения выбрана LSTM как наиболее подходящая модель прогнозирования временных рядов с возможностью сохранения памяти в течение длительного периода времени.

## АЛГОРИТМИЧЕСКАЯ ТОРГОВЛЯ

Алгоритмическая торговля в последние десятилетия активно развивается благодаря совокупности факторов: стремительному развитию методов машинного обучения, развитию технологий работы с данными и их анализа, росту возможностей хранения и обработки большего объема данных. Кроме того, сложность применяемых участниками рынка алгоритмов торговых систем растет, поскольку они конкурируют не только с теми, кто не использует автоматизированные системы, но и друг с другом. В связи с этими тенденциями исследование применимости различных алгоритмов машинного обучения к задачам алгоритмической торговли является актуальной задачей. В подобных исследованиях есть интерес не только со стороны компаний, занимающихся алгоритмической торговлей, таких как хедж-фонды, но и со стороны научного сообщества, поскольку применение алгоритмов машинного обучения к рассматриваемой области способно принести новые знания в развитие машинного обучения как раздела компьютерных наук, которые могут быть применены к другим предметным областям.

Теоретической ценностью работы является развитие исследований по применению методов машинного обучения к алгоритмической торговле. Кроме того, подобные методы, могут быть перенесены на другие предметные области при поиске стратегии наиболее точного прогноза следующего значения временного ряда для максимизации прибыли. В качестве практической ценности работы можно рассматривать возможность создания программных продуктов для торговли на бирже на основании алгоритмов, исследуемых в работе.

## МАШИННОЕ ОБУЧЕНИЕ НА ФОНДОВОМ РЫНКЕ

Фондовый рынок — публичная торговая площадка для покупки и продажи ценных бумаг различных компаний. Попытки точно предсказать будущие значения цены той или иной акции являются важной экономической задачей для широкого круга компаний, так как успешный прогноз может вылиться

в большой финансовый успех. С самого зарождения фондовых рынков люди искали и развивали способы максимально точно предсказывать котировки ценных бумаг. В итоге сформировались три основных подхода: фундаментальный анализ, технический анализ и технологические методы. Несмотря на порой взаимоисключающие результаты использования этих подходов в рамках одной задачи (одни предсказывают будущее значение, другие — направление движения для заданного периода времени), все они используются для торговли на фондовых рынках. При этом задача повышения точности прогноза продолжает оставаться актуальной для всех компаний, работающих на фондовых рынках. Данная работа посвящена применению машинного (включая глубокое) обучения к проблеме прогнозирования на фондовом рынке. И классическое машинное обучение, и нейронные сети используются в качестве прогнозных инструментов, тем не менее обучение с подкреплением и генетические алгоритмы не обрели той популярности и степени разработанности, как многие классические алгоритмы (случайный лес, метод опорных векторов). Одними из известных видов нейронных сетей является нейронная сеть прямого распространения вместе с методом обратного распространения ошибки (как метода обучения). В торговле на фондовом рынке нейронные сети могут имитировать действия агента, выполняющего те или иные задачи на рынке акций. Важной задачей при построении нейронных сетей является предобработка данных, и многие исследователи решают ее по-своему, предлагая новые методы, так как не существует устоявшихся практик однозначной предобработки входных массивов данных ввиду большого разнообразия как самих данных, так и ее типов. Выходные данные в рамках конкретной задачи, напротив, могут быть однозначно определены, принимая вид финансовых или экономических показателей. Что касается оценки результатов, то наиболее часто употребляемым методом является сравнение результатов работы алгоритма со значениями из тестовой выборки.

### СПОСОБЫ МАШИННОГО ОБУЧЕНИЯ

Наиболее традиционным способом применения машинного обучения к задаче алгоритмической торговли является обучение с учителем (supervised learning). В качестве целевой переменной, значение которой предсказывается алгоритмом, выступает изменение цены актива за некоторый период времени или, альтернативно, волатильность рынка. В начале исследований по применению машинного обучения к прогнозированию событий на финансовых биржах часто используемыми алгорит-

мами были простые и хорошо изученные деревья решений машины опорных векторов. Для таких алгоритмов входными признаками обычно являются технические индикаторы рынка. С началом успешного применения нейронных сетей к другим задачам, таким, как задачи обработки естественного языка (natural language processing), нейронные сети стали применяться и в области алгоритмической торговли, например, для предсказания движений цен на рынке под влиянием появляющихся политических или экономических новостей. Появилась возможность извлекать сигналы рынка из неструктурированной информации. Обучение с подкреплением (reinforcement learning) является одним из быстро развивающихся разделов машинного обучения, с помощью этого метода достигнуты такие результаты, как способность компьютера играть в различные игры на симуляторе Atari на уровне, превосходящем уровень человека, ничего изначально не зная о правилах игры.

Применение обучения с подкреплением к задачам алгоритмической торговли имело место и ранее, но более распространенным было применение обучения с учителем, так как позволяло достичь лучших результатов, однако в последнее время в связи с успехами, достигнутыми с помощью обучения с подкреплением в других предметных областях, возрос интерес к исследованиям возможности применения этого класса алгоритмов к задачам алгоритмической торговли.

Генетические алгоритмы успешно применяются к широкому спектру прикладных областей, например к робототехнике. Можно применять генетические алгоритмы к обобщенным задачам, например таким, как построение ансамблей алгоритмов машинного обучения для повышения точности работы отдельных алгоритмов и построение деревьев решений. В таких задачах генетические алгоритмы также показывают свою эффективность. В задачах алгоритмической торговли изначально они применялись в основном для генерации так называемых торговых правил. Этот метод заключается в том, что с помощью генетических алгоритмов создаются функции от технических индикаторов рынка, которые наиболее точно способны предсказывать изменение цены актива на некотором промежутке времени. В настоящее время генетические алгоритмы применяются на практике для построения автоматизированных торговых систем.

### ИССЛЕДОВАНИЯ ВРЕМЕННЫХ РЯДОВ

Анализ временных рядов представляет собой интересную область исследования с большим количеством прикладных задач в бизнесе, экономике, финансах и компьютерных науках. Целью анализа

временного ряда является изучение его динамики, построение модели, описывающей структуру данных и, наконец, прогноз будущих значений ряда. Крайне важно построение эффективной модели с получением максимально возможной точности.

Прогнозированием временных рядов традиционно занимается эконометрика, используя различные методы. Модель ARIMA в течение долгого времени считается стандартом в данной области. Однако, несмотря на крайне широкое использование, модель ARIMA имеет некоторые ограничения. Например, с помощью модели ARIMA сложно моделировать нелинейные зависимости между переменными. Более того, предполагается постоянное стандартное отклонение ошибок в модели, что на практике не наблюдается. Использование интегрированного подхода совместно с моделью GARCH (от англ. Generalized auto-regressive conditional heteroscedasticity — авторегрессионная условная гетероскедастичность) позволяет ослабить это ограничение, но построение и оптимизация модели становятся более сложной задачей. Модель GARCH, в этом смысле, предназначена для разъяснения кластеризации волатильности на финансовых рынках [7].

Традиционное машинное обучение привнесло новые подходы к задачам прогнозирования. Такие алгоритмы, как Метод опорных векторов (Support Vector Machines — SVM) и Случайный лес (Random Forest — RF) пользуются заслуженным вниманием специалистов во множестве областей, включая финансы. Следующим шагом являлось применение алгоритмов глубокого обучения (Deep learning — глубокое обучение).

В области глубокого обучения для прогнозирования временных рядов наиболее часто применяется модель LSTM — особый тип сверточной нейронной сети. Несмотря на относительную новизну этого метода, подход со стороны глубокого обучения обрел широкую популярность среди исследователей. Например, К. Краусс и его коллеги использовали ряд прогнозных подходов — глубокое обучение, градиентный бустинг над деревьями и случайный лес для моделирования S&P 500 [8]. Результатами являются выводы о низкой производительности моделей глубокого обучения и сложности их обучения. Сравнение различных моделей проводилось на основе экономических данных [9, 10].

## МОДЕЛИ ARIMA И LSTM И ПРОГНОЗНЫЕ ВЫЧИСЛЕНИЯ

Далее приведено математическое описание прогнозных моделей ARIMA и LSTM.

ARIMA — это обобщенная модель ARMA (Autoregressive moving average), которая соединяет в себе

AR (Autoregressive) и MA (Moving average) процессы. Как указано в акрониме модели, ARIMA( $p, d, q$ ) состоит из следующих частей [1, 11]:

AR — autoregressive. Регрессионная модель, которая использует зависимость между наблюдениями и число интегрированных наблюдений ( $p$ );

I — integrated. Для обеспечения стационарности путем взятия разностей ( $d$ );

MA — moving average. Подход, анализирующий зависимость между наблюдениями и остатками при применении модели к интегрированным наблюдениям ( $q$ ).

Простая форма AR модели порядка  $p$ , т.е. AR( $p$ ) может быть записана как линейный процесс следующим образом:

$$x_t = c + \sum_{i=1}^p \varphi_i x_{t-i} + \varepsilon_t, \quad (1)$$

где  $x_t$  — стационарная переменная;  $c$  — константа;  $\varphi_i$  — коэффициенты автокорреляции;  $\varepsilon_t$  — остатки модели, белый шум с нулевым средним.

Простая форма MA модели порядка  $q$ , т.е. MA( $q$ ) может быть записана так:

$$x_t = \mu + \sum_{i=0}^q \theta_i \varepsilon_{t-i}, \quad (2)$$

где  $\mu$  — математическое ожидание процесса (обычно предполагается равным нулю);  $\theta_i$  — веса;  $\theta_0$  — предполагается равным 1;  $\varepsilon_t$  — белый шум с нулевым средним. Комбинация этих двух моделей дает нам модель ARIMA порядка ( $p, q$ ):

$$x_t = c + \sum_{i=1}^p \varphi_i x_{t-i} + \varepsilon_t + \sum_{i=0}^q \theta_i \varepsilon_{t-i}. \quad (3)$$

Параметры  $p$  и  $q$  называются порядком AR и MA, соответственно. Использование ARIMA позволяет делать прогноз на нестационарных данных из-за введения в модель интегрированности. Это достигается с помощью взятия разностей — вычитания уровней временного ряда друг из друга.

Учитывая сезонность временного ряда, с большой вероятностью весомый вклад в модель внесут краткосрочные компоненты. Таким образом в модели также необходимо учитывать сезонность — сезонную ARIMA (seasonal ARIMA — SARIMA). Наиболее важными этапами являются оценка коэффициентов модели. Если дисперсия растет с течением времени, необходимо использовать стабилизирующие дисперсию трансформации и взятие разностей. Используя автокорреляционную функцию и частную автокорр-



реляционную функцию, следует измерять линейную зависимость между наблюдениями и величину  $q$ .

LSTM — это подвид сверточной нейронной сети со способностью «запоминания» значений наблюдений ранних этапов для использования их в будущем. Для ознакомления с LSTM необходимо дать некоторые определения и пояснения [2, 3].

Artificial neural network (искусственная нейронная сеть) — нейронная сеть, состоящая из по крайней мере трех слоев: входного, скрытого и выходного. Количество переменных в наборе данных определяет размерность или число узлов во входном слое. Эти узлы соединены ребрами, каждое из которых несет какой-либо вес, от которого зависит, может ли сигнал пройти через слой или нет. Нейронная сеть обучается, изменяя эти веса на основе данных. В скрытых слоях узлы используют функцию активации (например, сигмоида) на взвешенной сумме входных данных, чтобы преобразовать их в выходные (в нашем случае предсказания). Выходной слой генерирует вектор вероятностей для различных предсказанных значений и выбирает один с наименьшей ошибкой. Веса не могут быть оптимальны сразу, поэтому нейронная сеть обучается с учетом результата предыдущей итерации обучения. Эти итерации называются эпохами и веса изменяются до тех пор, пока не достигнут оптимального (заданного) значения.

Recurrent neural network (сверточная нейронная сеть) — особая нейронная сеть, целью которой является предсказание следующего наблюдения в серии. Идеей, стоящей за RNN, является желание извлечь полезную информацию из серии наблюдений для совершения предсказаний. Соответственно, более ранние наблюдения необходимо запоминать. В модели RNN внутренний слой служит для запоминания информации с предыдущих наблюдений серии. Главной проблемой является запоминание небольшого количества предшествующих наблюдений, что не подходит для длинных (финансовых, экономических) периодов. Для решения этой задачи были разработаны LSTM сети.

Long short-term memory нейронная сеть (долгая краткосрочная память) — это искусственная сверточная нейронная сеть, используемая в области глубокого обучения. В отличие от обычных нейронных сетей с прямой связью, LSTM сети имеют обратные связи. Подобные сети способны обрабатывать не только отдельные единичные данные (например, изображения), но и целые последовательности данных (например, аудиозапись речи или видеозаписи). Поэтому LSTM сети способны решать такие задачи, как распознавание рукописного ввода, распознавание речи, определение аномалий в больших потоках данных (сетевой

трафик, банковские транзакции). LSTM нейронные сети хорошо подходят для классификации, обработки и построения прогнозов на основе временных рядов, где взаимосвязанные явления могут происходить с неопределенным временным лагом. Этот временной лаг приводит к затруднениям в использовании классических нейронных сетей в решении данных проблем из-за затухания градиента, в то время как LSTM сети нечувствительны к величине временного лага.

В LSTM сети существуют 3 типа слоев:

1. Слой «забывания» (Forget gate) — на выход подается число от 0 до 1, где 1 обозначает необходимость полного запоминания, а 0 полностью стирает из памяти.
2. Слой «запоминания» (Memory gate) выбирает, какие данные необходимо сохранить. В первую очередь с помощью сигмоидного слоя выбираются значения, которые затем запоминаются.
3. Выходной слой (Output gate) выбирает информацию из каждой «ячейки», в которой произведено запоминание.

### СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ ARIMA И LSTM

Поставив целью сравнение моделей ARIMA и LSTM, авторами была проведена серия экспериментов на финансовых данных в виде временных рядов. Главным вопросом исследования — какой алгоритм ARIMA или LSTM позволяет более качественно производить прогнозы временных рядов.

Авторами собраны<sup>1</sup> исторические данные по ценам акций российских предприятий: Алроса, Газпром, КамАЗ, НЛМК, Киви, Роснефть, ВТБ и Яндекс. Каждый набор данных содержит такие значения, как 'Open', 'High', 'Low', 'Close', 'Volume'. Для анализа выбран набор данных по значению 'Close' за период с 02.06.2014 по 11.11.2019 г. с разбивкой по неделям. Также произведено разделение данных на тренировочный и тестовый набор в соотношении 70% к 30% без перемешивания. Тренировочный набор данных был использован для обучения модели, а тестовый — для оценки качества ее обучения. Графики собранных данных представлены на *рис. 1*.

Показатель RMSE (от англ. Root Mean Square Error, RMS Error — среднеквадратичная ошибка) используется для оценки качества предсказания модели. Он измеряет разницу между истинным и предсказанным значениями. Формула показателя задается формулой [12]:

<sup>1</sup> URL: <https://www.finam.ru/profile/moex-akcii/gazprom/export/> (дата обращения: 14.11.2019).

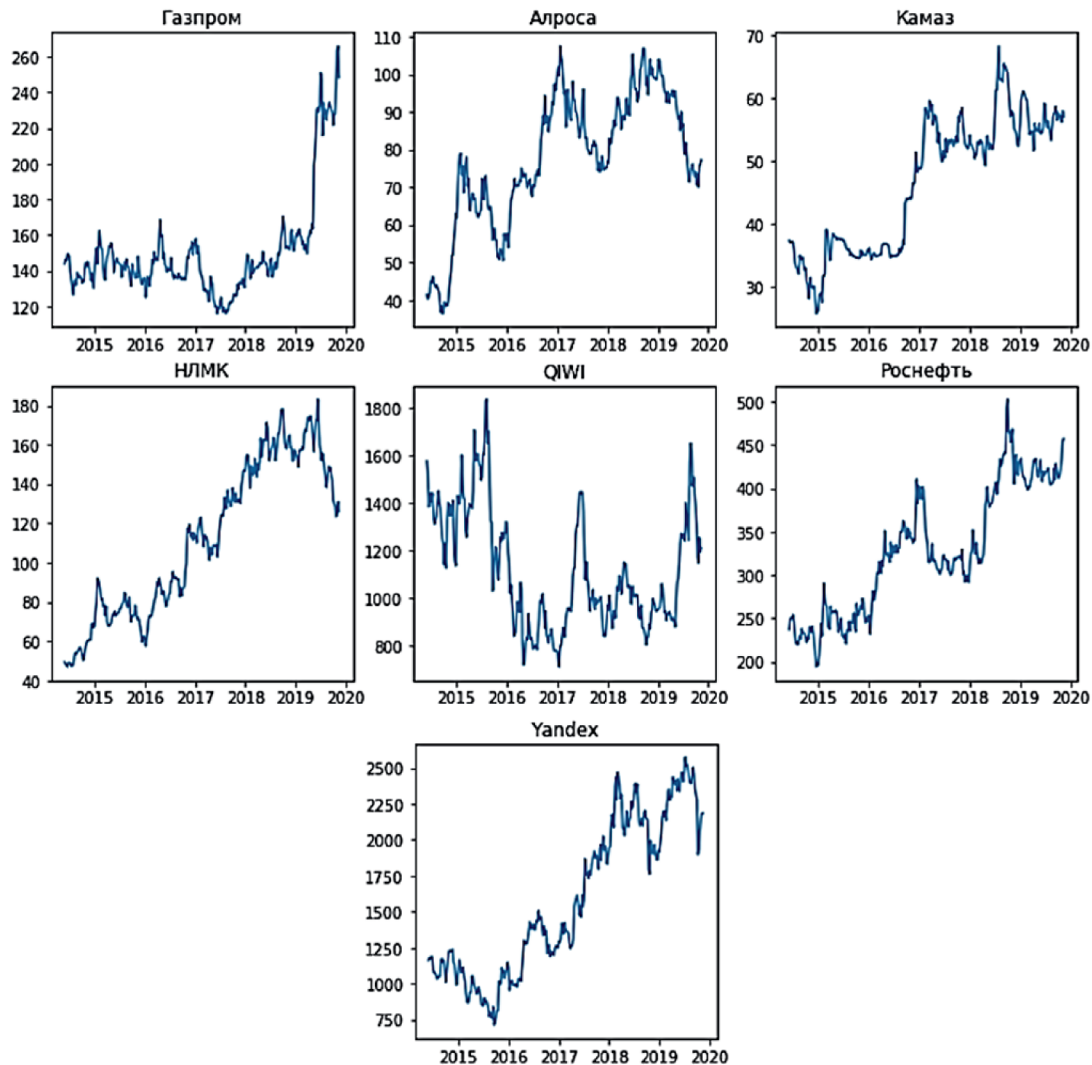


Рис. 1 / Fig. 1. Динамика изменения акций российских компаний / Stock performance of Russian companies

Источник / Source: составлено авторами / compiled by the authors.

$$RMSE = \sqrt{\frac{1}{N} \sum_1^N (x_i - \hat{x}_i)^2}, \quad (4)$$

где  $N$  — общее число наблюдений; а  $\sum_1^N (x_i - \hat{x}_i)$

представляет собой квадрат суммы разностей истинного значения и предсказания модели. Главным достоинством использования метрики RMSE является «штрафование» модели за большие ошибки.

### РАЗРАБОТКА АЛГОРИТМОВ ARIMA И LSTM

Алгоритмы разработаны авторами для прогноза временных рядов и основаны на подходе “Rolling forecasting origin” («прогнозирование происхождения») [13]. В основе алгоритма лежит фокус на пред-

сказании следующей точки временного ряда каждого набора данных. Подход использует тренировочные наборы данных, каждый из которых содержит на одно наблюдение больше, чем предыдущий. Существует несколько различных вариаций данного алгоритма, среди которых известны следующие:

1. Одношаговый без переоценки. Модель оценивает один набор тренировочных данных и затем вычисляет прогноз на один шаг вперед.
2. Многошаговый без переоценки. Аналогичный предыдущему, однако прогноз дается на несколько следующих шагов.
3. Многошаговый с переоценкой. Прогноз вычисляется на несколько шагов вперед, однако модель обучается заново на каждой итерации перед прогнозом.

В данной статье используется алгоритм с построением новой модели каждый раз, когда добавляются

```

# Скользящая ARIMA
Вход: временной ряд
Выход: RMSE прогноза
# Разбиение данных на тренировочный и тестовый наборы:
# 70% для тренировочного и 30% для тестового
1. size ← length(series) * 0.70
2. train ← series[0...size]
3. test ← series[size...length(size)]
# Подготовка структуры данных
4. history ← train
5. predictions ← empty
# Прогноз
6. for each t in range(length(test)) do
7. model ← ARIMA(history, order=(5, 1, 0))
8. model_fit ← model.fit()
9. hat ← model_fit.forecast()
10. predictions.append(hat)
11. observed ← test[t]
12. history.append(observed)
13. end for
14. MSE = mean_squared_error(test, predictions)
15. RMSE = sqrt(MSE)
16. Return RMSE
    
```

Рис. 2 / Fig. 2. Алгоритм скользящего окна для модели ARIMA / Sliding window algorithm for ARIMA model

Источник / Source: составлено авторами / compiled by the authors.

новые данные. Алгоритмы реализованы в среде программирования Python<sup>2</sup> с подключенными библиотеками Keras, Theano и Statsmodels.

На рис. 2 представлен алгоритм скользящего окна для модели ARIMA. Алгоритм ARIMA разбит на следующие этапы:

1) входные данные (каждый финансовый временной ряд) разбиваются на тренировочный и тестовый наборы;

2) создаются дополнительные структуры (массивы, в которых хранятся записи и вносятся изменения о «текущих» исторических данных, прогнозы) для обучения модели;

3) модель обучается на текущем наборе данных и производит прогноз, который далее записывается в массив predictions, изменяя массив history;

4) вычисляется метрика RMSE для модели ARIMA.

На рис. 3 представлен алгоритм скользящего окна для модели LSTM.

В отличие от моделей регрессии, в анализе временных данных существует последовательность зависимостей между входными переменными. Сверточные нейронные сети очень хорошо улавливают подобные зависимости.

Алгоритм LSTM разбит на следующие этапы:

1) данные разбиваются на тренировочный и тестовый наборы;

2) фиксируется параметр random\_state для воспроизведения результатов;

3) определяется функция fit\_lstm, в качестве аргументов принимается набор данных, число эпох и число нейронов;

4) создается скрытый слой;

5) задается оптимизационный алгоритм и функция потерь (Adam и MSE);

6) после этого модель обучается, при этом ее внутреннее состояние каждый раз восстанавливается (код в строке 12) на исходное;

7) производится прогноз и вычисляется метрика RMSE для модели LSTM.

## РЕЗУЛЬТАТЫ ИСПОЛНЕНИЯ АЛГОРИТМОВ

Результаты исполнения алгоритмов представлены на рис. 4. На финансовых временных рядах модель LSTM показала себя лучше, чем модель ARIMA. Значения RMSE моделей равны 10,8 и 26,7, соответственно. Применение модели LSTM позволяет уменьшить значение RMSE на 65% по сравнению с моделью ARIMA.

## ВЫВОДЫ

Сравнительный анализ двух алгоритмов на основе моделей LSTM и ARIMA определил превосходство

<sup>2</sup> URL: <https://www.python.org/> (дата обращения: 15.01.2020).

```

# Скользящий LSTM
Вход: временной ряд
Выход: RMSE прогноза
# Разбиение данных на тренировочный и тестовый наборы:
# 70% для тренировочного и 30% для тестового
1. size ← length(series) * 0.70
2. train ← series[0...size]
3. test ← series[size...length(size)]
# Фиксируем random seed для воспроизводимости результата
4. set random.seed(7)
# Создание и обучение модели
Procedure fit_lstm(train, epoch, neurons)
5. X ← train
6. y ← train - X
7. model = Sequential()
8. model.add(LSTM(neurons), stateful=True))
9. model.compile(loss='mean_squared_error',
optimizer='adam')
10. for each i in range(epoch) do
11. model.fit(X, y, epochs=1, shuffle=False)
12. model.reset_states()
13. end for
return model
# Прогноз на шаг вперед
Procedure forecast_lstm(model, X)
14. yhat ← model.predict(X)
return yhat
15. epoch ← 1
16. neurons ← 4
17. predictions ← empty
# Обучение LSTM модели
18. lstm_model = fit_lstm(train, epoch, neurons)
# Прогноз на тренировочном наборе данных
19. lstm_model.predict(train)
# Валидация на тестовых данных
20. for each i in range(length(test)) do
21. # прогноз
22. X ← test[i]
23. yhat ← forecast_lstm(lstm_model, X)
24. # сохранить прогноз
25. predictions.append(yhat)
26. expected ← test[i]
27. end for
28. MSE ← mean_squared_error(expected, predictions)
29. Return (RMSE ← sqrt(MSE))

```

Рис. 3 / Fig. 3. Алгоритм скользящего окна для модели LSTM / Sliding window algorithm for LSTM model

Источник / Source: составлено авторами / compiled by the authors.

модели LSTM над моделью ARIMA в части минимизации среднеквадратической ошибки RMSE. Значение ошибки RMSE модели LSTM меньше соответствующей ошибки для модели ARIMA в среднем на 65%.

Развивающиеся в настоящее время технологии анализа больших данных, алгоритмы машинного обучения и, в частности, алгоритмы глубокого обучения набирают популярность среди исследовате-

лей в различных областях науки. Главным вопросом является точность и адекватность новых подходов по сравнению с традиционными методами. В статье проведено сравнение точности прогнозирования моделей ARIMA и LSTM, представляющих два разных метода. Обе модели были применены на различных наборах финансовых данных — временных рядах цены закрытия акции на конец недели — и результаты показывают превосходство модели LSTM.



	Имя компании	LSTM RMSE	ARIMA RMSE	Уменьшение RMSE, %
0	Алроса	1,12	3,04	63,158
1	Газпром	3,81	8,17	53,366
2	Камаз	0,38	1,76	78,41
3	НЛМК	1,19	5,34	77,715
4	QIWI	25,32	63,13	59,892
5	Роснефть	4,61	13,73	66,423
6	Yandex	39,19	91,67	57,249
7	Среднее	10,803	26,691	65,173

Рис. 4 / Fig. 4. Результаты исполнения алгоритмов ARIMA и LSTM / ARIMA and LSTM algorithm execution results

Источник / Source: составлено авторами / compiled by the authors.

Существующие на данный момент задачи оценки качества моделей весьма многочисленны. Авторы затронули лишь некоторые их аспекты. Представляется целесообразным исследование и применение новых алгоритмов обработки больших данных на осно-

ве нейронных сетей, больших графов и машинного обучения для исследования инструментов финансового рынка российских компаний. В этом смысле перспективным видится агрегированный подход, комбинирующий разные подходы и алгоритмы.

### БЛАГОДАРНОСТЬ

Работа выполнена при поддержке гранта РФФИ 18–00–01103. Финансовый университет, Москва, Россия.

### ACKNOWLEDGEMENTS

The article was supported by the RFBR grant 18–00–01103. Financial University, Moscow, Russia.

### СПИСОК ИСТОЧНИКОВ

1. Магнус Я.Р., Катышев П.К., Пересецкий А.А. Эконометрика. Начальный курс. М.: Дело; 2007. 504 с.
2. Schmidhuber J. Habilitation. System modeling and optimization. Postdoctoral thesis. Munich: Technical University of Munich; 1993. 209 p. (In German).
3. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Computation*. 1997;9(8):1735–1780. DOI: 10.1162/neco.1997.9.8.1735
4. Brownlee J. Time series prediction with LSTM recurrent neural networks in Python with Keras. Machine Learning Mastery. 2016. URL: <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras>
5. Gers F.A., Schmidhuber J., Cummins F. Learning to forget: Continual prediction with LSTM. *Neural Computation*. 2000;12(10):2451–2471. DOI: 10.1162/089976600300015015
6. Кратович П.В. Нейронные сети и модели ARIMA для прогнозирования котировок. *Программные продукты и системы*. 2011;(1):95–98.
7. Garcia F., Guijarro F., Moya I., Oliver J. Estimating returns and conditional volatility: A comparison between the ARMA-GARCH-M models and the backpropagation neural network. *International Journal of Complex Systems in Science*. 2012;1(2):21–26.
8. Krauss C., Do X.A., Huck N. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*. 2017;259(2):689–702. DOI: 10.1016/j.ejor.2016.10.031
9. Chung J., Lee D., Seo Y., Yoo C.D. Deep attribute networks. NIPS Workshop on Deep Learning and Unsupervised Feature Learning. 2012;3. URL: [http://www.eng.uwaterloo.ca/~jbergstr/files/nips\\_dl\\_2012/Paper%2011.pdf](http://www.eng.uwaterloo.ca/~jbergstr/files/nips_dl_2012/Paper%2011.pdf)
10. Fischer T., Krauss C. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*. 2018;270(2):654–669. DOI: 10.1016/j.ejor.2017.11.054

11. Эконометрика. Елисеева И.И., ред. М.: Финансы и статистика; 2006. 576 с.
12. Armstrong J.S., Collopy F. Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting*. 1992;8(1):69–80. DOI: 10.1016/0169–2070(92)90008-W
13. Hyndman R.J., Athanasopoulos G. Forecasting: Principles and practice. 2<sup>nd</sup> ed. Melbourne: OTexts; 2018. 382 p.

## REFERENCES

1. Magnus Ya.R., Katyshev P.K., Peresetsky A.A. Econometrics. Beginner course. Moscow: Delo; 2007. 504 p. (In Russ.).
2. Schmidhuber J. Habilitation. System modeling and optimization. Postdoctoral thesis. Munich: Technical University of Munich; 1993. 209 p. (In German).
3. Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Computation*. 1997;9(8):1735–1780. DOI: 10.1162/neco.1997.9.8.1735
4. Brownlee J. Time series prediction with LSTM recurrent neural networks in Python with Keras. Machine Learning Mastery. 2016. URL: <https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras>
5. Gers F.A., Schmidhuber J., Cummins F. Learning to forget: Continual prediction with LSTM. *Neural Computation*. 2000;12(10):2451–2471. DOI: 10.1162/089976600300015015
6. Kratovich P.V. Neural networks and ARIMA models for predicting quotes. *Programmnye produkty i sistemy = Software & Systems*. 2011;(1):95–98. (In Russ.).
7. Garcia F., Guijarro F., Moya I., Oliver J. Estimating returns and conditional volatility: A comparison between the ARMA-GARCH-M models and the backpropagation neural network. *International Journal of Complex Systems in Science*. 2012;1(2):21–26.
8. Krauss C., Do X.A., Huck N. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*. 2017;259(2):689–702. DOI: 10.1016/j.ejor. 2016.10.031
9. Chung J., Lee D., Seo Y., Yoo C.D. Deep attribute networks. NIPS Workshop on Deep Learning and Unsupervised Feature Learning. 2012;3. URL: [http://www.eng.uwaterloo.ca/~jbergstr/files/nips\\_dl\\_2012/Paper%2011.pdf](http://www.eng.uwaterloo.ca/~jbergstr/files/nips_dl_2012/Paper%2011.pdf)
10. Fischer T., Krauss C. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*. 2018;270(2):654–669. DOI: 10.1016/j.ejor.2017.11.054
11. Eliseeva I.I., ed. Econometrics. Moscow: Finansy i statistika; 2006. 576 p. (In Russ.).
12. Armstrong J.S., Collopy F. Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting*. 1992;8(1):69–80. DOI: 10.1016/0169–2070(92)90008-W
13. Hyndman R.J., Athanasopoulos G. Forecasting: Principles and practice. 2<sup>nd</sup> ed. Melbourne: OTexts; 2018. 382 p.

## ИНФОРМАЦИЯ ОБ АВТОРАХ / ABOUT THE AUTHORS



**Андрей Вадимович Алжеев** — студент магистратуры Департамента анализа данных, принятия решений и финансовых технологий, Финансовый университет, Москва, Россия  
**Andrei V. Alzheev** — Master’s student, Department of Data Analysis, Decision Making and Financial Technology, Financial University, Moscow, Russia  
 alzheev@gmail.com



**Расул Ахматович Кочкаров** — кандидат экономических наук, доцент Департамента анализа данных, принятия решений и финансовых технологий, Финансовый университет, Москва, Россия  
**Rasul A. Kochkarov** — Cand. Sci. (Econ.), Assoc. Prof., Department of Data Analysis, Decision Making and Financial Technology, Financial University, Moscow, Russia  
 rasul\_kochkarov@mail.ru

*Статья поступила в редакцию 18.11.2019; после рецензирования 17.12.2019; принята к публикации 20.12.2019.  
 Авторы прочитали и одобрили окончательный вариант рукописи.  
 The article was submitted on 18.11.2019; revised on 17.12.2019 and accepted for publication on 20.12.2019.  
 The authors read and approved the final version of the manuscript.*