Прогнозирование временных рядов при обработке потоковых данных в реальном времени

Р.А. Ельченков, М.Е. Дунаев, К.С. Зайцев

***Аннотация*. Целью данной работы является исследование методов предсказания значений временных рядов при обработке потоковых данных в распределенных системах в режиме реального времени. Для этого авторами предлагается модификация модели авторегрессии с заданным порядком AR путем добавления в нее функции наследования предыдущих значений временного ряда. Результаты сравнительных экспериментов предложенной модификации, названной Real-Time AR, с классическими AR и ARIMA подтвердили эффективность модификации. Это особенно явно проявляется при наличии аномалий поведения реального временного ряда. Предложенная модификация алгоритма позволяют не только распараллеливать вычисления, но и выполнять настройку модели "на лету" в экосистеме Apache Spark. Для проведения экспериментов с алгоритмами был построен специальный массив данных (датасет) - срез данных из 1000 измерений лога метрик работы сервера Apache Kafka с одной темой, двумя производителями и одним потребителем. В массив были искусственно добавлены аномальные фрагменты, отличающиеся большим числом сообщений в секунду и/или размером сообщения. Значения предложенного массива данных были нормализованы и смещены на среднее значение по тренировочной выборке предобучения модели. Результаты применения предложенного алгоритма при решении задач прогнозирования значений временных рядов показали, что наличие аномалий поведения объектов не вносит значительных искажений в результаты предсказания значений.**

***Ключевые слова* – журнал событий, технологическая платформа, машинное обучение, статистические методы, Apache Software Foundation, Apache Spark, онлайн-обучение**

I. ВВЕДЕНИЕ

Сегодня потоковые данные являются важнейшей составляющей современных информационных систем: от показателей датчиков в умных объектах и на производственных предприятиях до высоко нагруженных (highload) сервисов в IT-компаниях.

Для контроля подобных систем необходимо своевременно анализировать текущее состояние и превентивно оценивать динамику этого состояния в ближайшем будущем и на основе полученной картины данных принимать решения по управлению системой. Таким образом, ощущается всё большая необходимость в средствах анализа и обработки данных в потоковом, real time режиме. Акцент данной работы смещен в сторону решения задачи предсказания значений временных рядов.

Задача анализа временных рядов активно изучалась на протяжении десятков лет. Уже существует ряд успешных подходов, считающихся "классическими". К подобным моделям можно отнести статистические алгоритмы (ARMA, ARIMA, GARCH, модель Хольта-Винтерса и т.д.), модель линейной регрессии [1], нейронные сети различных архитектур (RNN, LSTM и т.д.) [2, 3].

Эти подходы хорошо описаны, всесторонне исследованы, и нашли применение в производственной среде. Тем не менее, одним из их главных недостатков является быстрое «устаревание модели». Для любого управляемого (с обучением) алгоритма Machine Learning требуется настраивать параметры модели по заранее подготовленной обучающей выборке, тестировать корректность предсказаний на тестовой выборке и затем интерполировать результаты на более широкую выборку реальных данных. Однако данные временных рядов могут не быть (а на практике в большинстве случаев и не бывают) стационарными: их статистические характеристики зависят от времени, и могут меняться произвольным образом в связи с чередой непредсказуемых внешних факторов.

Для того чтобы параметры модели были актуальны на период ее эксплуатации применяется систематическое переучивание модели через определенные интервалы времени. К сожалению, такое решение не эффективно, так как принуждает постоянно использовать ресурсы не для эффективной работы системы, а для ее реконфигурации, что усложняет поддержку системы силами инженеров.

Тем не менее, существует класс моделей, позволяющий не обучать их заново через определённые промежутки времени, а «дообучать» (модифицировать текущие параметры модели) в режиме реального

времени. Такие алгоритмы обсуждаются в публикациях в области online learning. Отметим, что в моделях такого рода скорость обработки поступающих данных должна быть достаточно велика, чтобы необработанные данные не начали накапливаться. В качестве решения для высокоинтенсивной работы предлагается использовать фреймворки, позволяющие производить распределенные вычисления над большими объемами данных. Одним из наиболее популярных фреймворков является Apache Spark, предназначенный для распределенной обработки структурированных и неструктурированных данных, и входящий в экосистему проектов Hadoop [4, 5].

В данной работе исследуются подходы к решению задачи предсказания значений временных рядов из журналов событий при обработке потоковой информации в реальном масштабе времени в распределенных вычислительных системах. Априорно предполагается, что возможны резкие изменения значений отдельных показателей временных рядов, вызванные разбалансировкой нагрузки, программными сбоями, попытками незаконного вмешательства в работу и т.п.

II. АЛГОРИТМЫ ПРЕДСКАЗАНИЯ ЗНАЧЕНИЙ

Для предсказания значений временных рядов рассмотрим два часто используемых алгоритма.

а) Авторегрессионная модель. Это такая модель, в которой значения временного ряда в текущий момент времени линейно зависят от предыдущих значений. Ее можно представить уравнением вида:

$$X_t = c + \sum_{i=1}^{p} \alpha_i X_{t-i} + \varepsilon_t \quad , \quad (1)$$

где α_i - коэффициенты авторегрессии,
c - постоянный коэффициент,
ε_t - стационарный шум, т.е. последовательность случайных величин, распределенных, как правило, по нормальному закону со средним, равным нулю,
p - порядок модели.

Задача оптимизации заключается в определении как параметра p, т.е. числа предыдущих значений используемых для прогнозирования текущего значения временного ряда, так и коэффициентов авторегрессии.

б) ARMA. Модель авторегрессии скользящего среднего – это математическая модель, которая используется для предсказания значений переменной стационарных временных рядов.

Моделью ARMA (p, q), где p и q - целые числа, задающие порядок модели, принято называть процесс генерации временного ряда вида:

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \alpha_i X_{t-i} + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} \quad , \quad (2)$$

где *c* - константа, ε_t - стационарный шум, α_i и β_i - авторегрессионные коэффициенты и коэффициенты скользящего среднего.

Задача оптимизации заключается в определении порядка модели (чисел p и q) и коэффициентов α_i и β_i.

III. ПОДГОТОВКА ДАННЫХ

Для тестирования и сравнения результатов работы построенных моделей локально была развернута система с архитектурой, представленной на рис. 1.

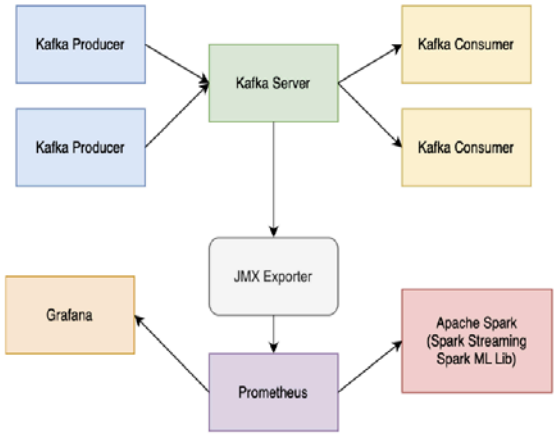

Рис. 1. Архитектура системы.

Для подготовки экспериментальных данных использовались метрики логов Open Source продукта Apache Kafka. В качестве производителей (producers) сообщений Kafka выступали написанные на высокоуровневом языке программирования Python программы, посылающие сообщения с определенной периодичностью в заранее заданные темы (topics) Kafka.

Эти же программы генерировали “аномалии”, представленные потоком сообщений с количеством сообщений в секунду, превышающим среднее значение в 2500 раз по сравнению с аналогичной метрикой для любых других временных отрезков (см. рис. 2 и 3).

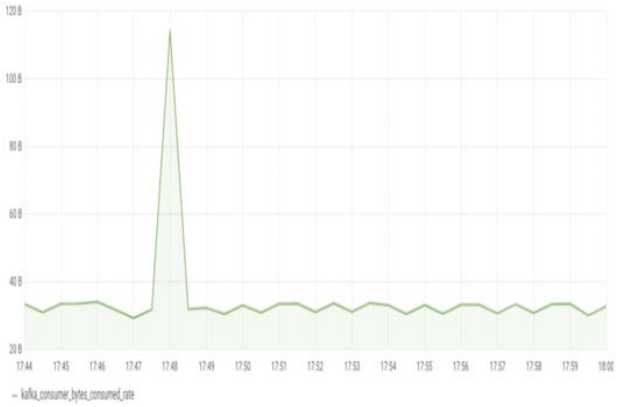

Рис. 2. Фрагмент метрики kafka_consumer_bytes_ consumed_rate в Graphana (с пиком).

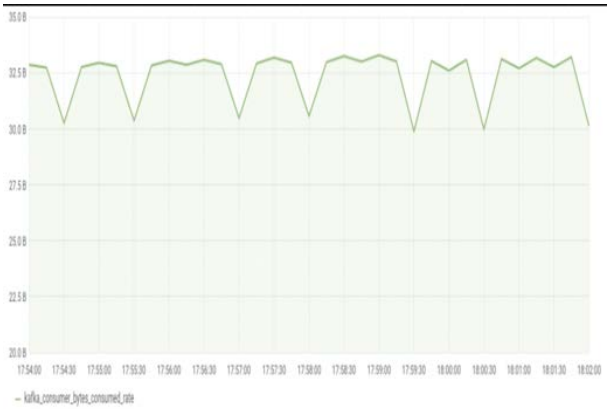

Рис.3. Фрагмент метрики kafka_consumer_bytes_ consumed_rate в Graphana (без пика).

Использовались два производителя (см. табл.1).

Таблица 1. Трафик производителей Kafka

Producer 1	Producer 2
формирует сообщения длиной 70 байт и отправляет их по плану: • одно сообщение раз две секунды, • два сообщение раз в десять секунд, • пять сообщений каждую пятидесятую секунду, • сто сообщений каждые пять минут и двадцать секунд	формирует сообщения переменного размера, отправляет их, и дополнительно генерирует два аномальных сообщения в соответствии с планом: • одно сообщение длиной 70 байт каждые две секунды, • два аномальных сообщения длиной 960 килобайт

Получаемый датасет представляет собой набор из более, чем двух тысяч временных рядов - метрик, собранных JMX-экспортером в количестве 1000 строк. Часть метрик, некоррелированных или слабо коррелированных с количеством сообщений в секунду, таких как jvm_threads_state, jmx_scrape_duration_seconds, была исключена из датасета. Данные были нормализованы и смещены относительно среднего тренировочной выборки.

В настоящем исследовании не рассмотрены вопросы появления новых метрик (при заведении новой темы в Kafka) и иррелевантности старых метрик (при удалении тем или отдельных потребите-лей/производителей Kafka) потому, что они имеют исключительно технический характер.

В реальных условиях данные будут поступать в реальном времени из сервисов мониторинга работы приложений, однако для исследовательских целей достаточно использовать заранее подготовленный датасет, поступающий в потоковом режиме, например, с помощью Apache Streaming.

Для решения задачи предсказания значений временных рядов было выбрано несколько наиболее информативных метрик продукта kafka (табл.2)

Таблица 2. Используемые метрики Kafka.

Название	Значение
kafka_consumer_records_consumed_rate	Среднее количество записей, потребляемых в секунду
kafka_consumer_bytes_consumed_rate	Среднее количество байтов, потребляемых потребителем в секунду
kafka_consumer_fetch_rate	Количество запросов на выборку в секунду.
kafka_consumer_fetch_latency_max	Максимальное время, необходимое для запроса на выборку

Проиллюстрируем решение этой задачи прогнозирования значений временного ряда на примере метрики *kafka_consumer_bytes_consumed_rate*, которая показывает среднее количество байтов, потребляемых в секунду. Эта метрика сильно коррелирует с количеством сообщений в секунду, что при росте объема входного трафика позволяет использовать ее в текущей задаче. Исходный и нормализованный виды этой метрики представлены на рис. 4 и 5.

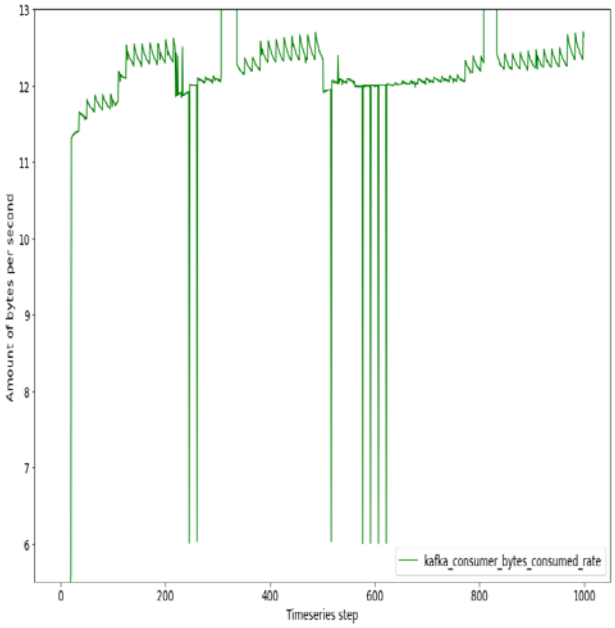

Рис. 4. Данные метрики kafka_consumer_bytes_consumed_rate.

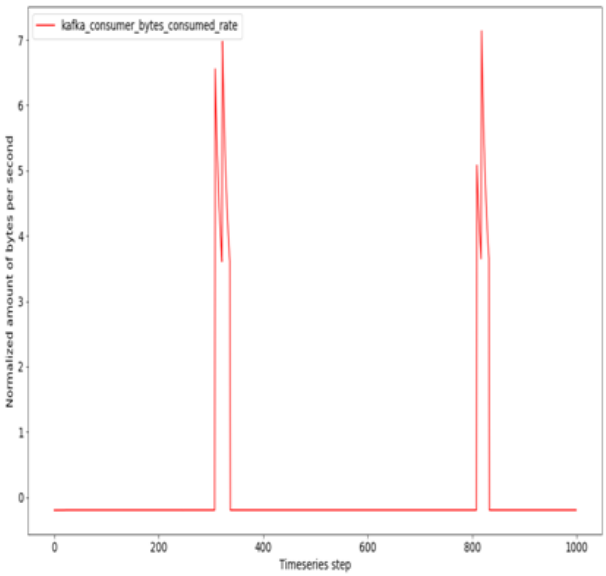

Рис. 5. Нормализованные данные метрики kafka_consumer_bytes_consumed_rate.

На приведенных графиках четко выделяются пики аномалий. В реальной практике такие пики получаются при DoS и DDoS атаках. Имеющийся датасет был разделен на две части: три четверти – обучающая выборка и одна четверть – тестирующая выборка. Такой подход позволяет избежать проблемы холодного старта модели, так как для нахождения локального минимума на каждой итерации требуется значительно меньше времени и вычислительных ресурсов.

IV. МОДИФИКЦИЯ МОДЕЛИ AR

В качестве основы для создания модели реального времени была выбрана Streaming Linear Regression, встроенная в библиотеку Spark MLlib, Модификация, названная авторами Real-Time AR, представляет собой модель линейной регрессии, которая при каждой новой партии поступивших данных модифицирует свои веса с помощью метода градиентного спуска. Иными словами Real-Time AR – это модель авторегрессии AR(p), где p - порядок авторегрессии, (количество членов в модели, отвечающих предыдущим p значениям временного ряда). Подобная модель с некоторой точностью при определенных значениях p аппроксимирует ARMA(p,q) - модель, помимо авторегрессии, включающую в себя еще и модель скользящего среднего, где p - порядок авторегрессии, а q - порядок модели скользящего среднего [8].

Техническая реализация модели Real-Time AR была выполнена в виде написанного на языке программирования Scala класса, наследующего от класса Streaming Linear Regression, и реализующего механизм сдвига используемых для обучения старых p значений временного ряда к новым, полученным за последний отрезок времени (см. рис. 6). С помощью такого подхода временной ряд конвертируется в массив данных с пространством признаков, состоящим из значений, смещенных относительно текущего значения временного ряда (рис. 7). И, задача предсказания сводится к задаче регрессии, которую можно решать с помощью встроенных в Streaming Linear Regression методов.

```
abstract class AR(p: Int) extends StreamingLinearRegressionWithSGD {
  var featuresHistoric: ListBuffer[Double] = ListBuffer.empty

  def addToFeatures(v: Double): ListBuffer[Double]
}
```

Рис. 6. Интерфейс класса авторегрессии.

Metric
1
2
3

→

Feature 1	Feature 2	Feature 3
1	null	null
2	1	null
3	2	1

Рис. 7. Преобразование временного ряда в датасет для линейной регрессии.

V. РЕЗУЛЬТАТЫ СРАВНЕНИЯ ПРЕДСКАЗАНИЙ

Для проведения экспериментов и последующего сравнительного анализа были выбраны следующие модели предсказаний:

предложенная авторами модификация Real-Time AR авторегрессии, с порядком авторегрессии p=5,

классическая модель авторегрессии, обученная на тренировочном датасете с порядком p=5,

интегрированная модель авторегрессии - скользящего среднего ARIMA с порядком авторегрессии p=5, степенью дифференцирования d=2, и порядком скользящего среднего q=1.

Ниже на рис.8 представлены графики результатов предсказаний моделей.

Несложно заметить, что в случае, когда значения датасета имеют периодические выбросы, т.е. сильную динамику изменений классические модели не справляются с задачей предсказания временных рядов. Такое поведение легко объясняется и связано с тем, что аномалии носят непериодический характер, а значит, не вписываются в паттерн, к которому была обучена классическая модель

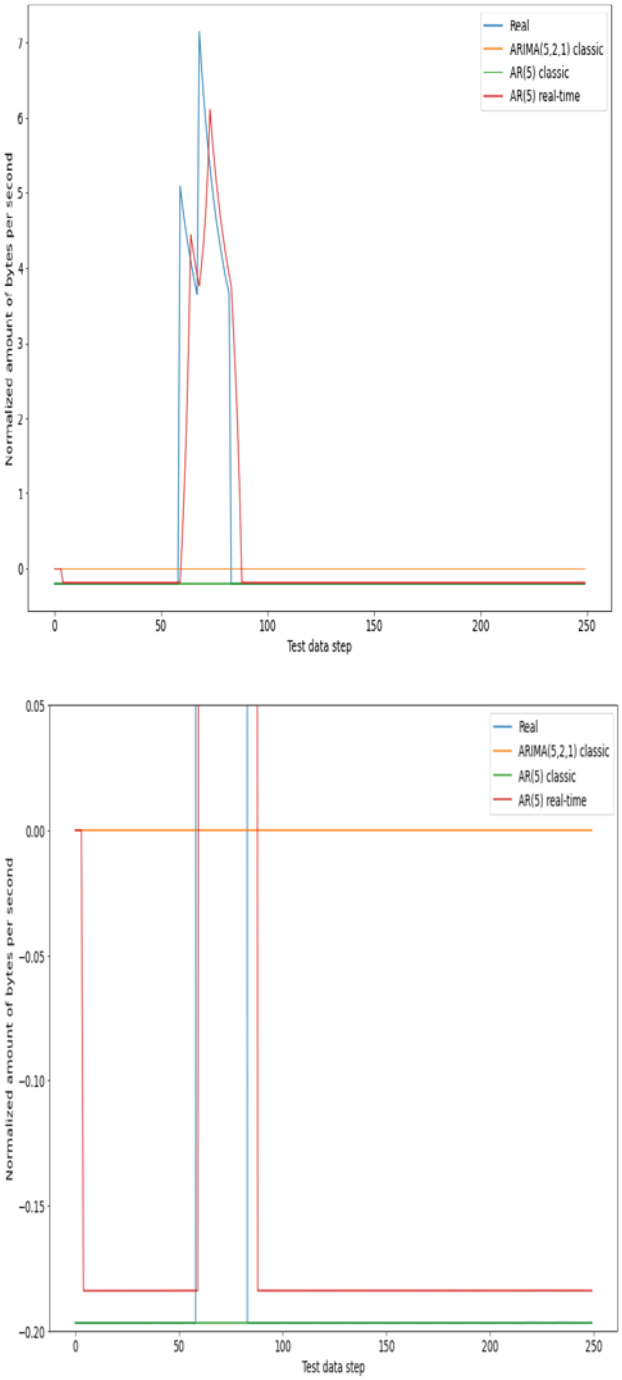

Рис. 8. Результаты предсказаний моделей на тестовой выборке.

.

VI. ДИСКУССИЯ ПО ТЕМЕ ИССЛЕДОВАНИЙ

В настоящий момент стандартом в индустрии предсказания являются модели, предобученные на выборках из данных, поступающих напрямую из производственный среды. Для повышения достоверности ответов модели, используемые в задачах обработки временных рядов, переобучаются

через определенные промежутки времени [9]. Такой подход, очевидно, приводит к дополнительным финансовым издержкам, связанным как с поддержанием работы серверов для повторного обучения моделей, так и с затратами рабочего времени сотрудников компаний.

Напрашивается решение из сферы online learning. В этом направлении сегодня ведутся и теоретические разработки [10, 11], и некоторые популярные фреймворки уже поддерживают ряд online learning алгоритмов. Например, Apache Spark предоставляет имплементацию real-time версий K-Means, Linear Regression [12, 13]. В библиотеке scikit-learn присутствует аналогичный ряд моделей, а также некоторые дополнительные модели для задач классификации, не рассматриваемые в рамках настоящей работы. В отличие от scikit-learn, Apache Spark предоставляет возможность распределенных вычислений, что может значительно ускорить процесс обработки больших данных. Однако в Apache Spark отсутствуют статистические модели, предназначенные для предсказания временных рядов, такие как AR, ARMA, ARIMA, GARCH и т.д.

В статье [14] предложена реализация ARIMA в режиме реального времени, более того авторами предлагается даже open-source библиотека предсказания с помощью ARIMA-моделей в реальном времени, однако предлагаемый в статье фреймворк не пригоден для обработки данных в распределенных средах.

Применение нейронных сетей демонстрирует хороший результат, в областях с известными шаблонами значений, однако нейронные сети занимают много времени для переобучения, поэтому мало пригодны для работы с неизвестными шаблонами изменения данных временных рядов в реальном времени. Кроме этого, в работе [15] отмечается негативный аспект нейронных сетей как “черного ящика” для выявления причин неудачной работы.

VII. ЗАКЛЮЧЕНИЕ

В данной работе исследовались подходы к решению задачи предсказания значений временных рядов из журналов событий при обработке потоковой информации в реальном масштабе времени в распределенных вычислительных системах. При этом предполагается, что в реальных процессах возможны резкие изменения значений отдельных показателей временных рядов, вызванные разными причинами.

В исследовании проведен обзор имеющихся публикаций и open-source решений в области online learning.

Выявлены недостатки классических моделей предсказания значений временных рядов при работе с потоковыми данными в реальном времени.

Предложена и реализована Real Time AR - модификация классической модели AR(ARMA) для работы в режиме реального времени с большими данными в экосистеме Apache Spark.

Для проведения испытаний алгоритмов подготовлен специальный датасет, состоящий из метрик JVM и Apache Kafka при передаче сообщений двумя производителями Kafka. При создании датасета изменялись такие величины, как количество сообщений в секунду и размер передаваемых сообщений. В датасет искусственно включены два выброса значений временного ряда в виде сообщений с размером, многократно превышающими средний размер стандартного сообщения.

Были проведены эксперименты с классическими моделями AR(5), ARIMA(5,2,1) и предложенной модификацией Real-Time AR(5) для определения эффективности работы моделей на подготовленном датасете с выбросами значений. Классические модели не справились с предсказанием аномального участка датасета. Real-Time AR предсказала и аномальный и вне-аномальный участки, однако точность модели требует совершенствования.

Подводя итог, можно сказать, что разработанная real-time модель для работы с временными рядами демонстрируют удовлетворительные результаты и может являться основой для дальнейших исследований и развития online learning моделей в экосистеме Apache Spark.

БЛАГОДАРНОСТИ

Авторы выражают благодарность Высшей инжиниринговой школе НИЯУ МИФИ за помощь в возможности опубликовать результаты выполненной работы.

БИБЛИОГРАФИЯ

[1] Peter J Brockwell, Peter J Brockwell, Richard A Davis, and Richard A Davis. Introduction to time series and forecasting. Springer, 2016.

[2] A. Aldweesh, A. Derhab, and A. Z. Emam, “Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues, Knowledge-Based Systems, vol. 189, p. 105124, 2020

[3] Yagmur Gizem Cinar, Hamid Mirisaee, Parantapa Goswami, ´Eric Gaussier, AliAit-Bachir, and France Vadim Strijov. Time series forecasting using rnns: anextended attention mechanism to model periods and handle miss-ing values.CoRR, abs/1703.10089, 2017.

[4] Unified engine for large-scale data analytics [электронный ресурс] https://spark.apache.org/ Дата обращения 01.10.2021

[5] Apache Hadoop https://hadoop.apache.org/ [электронный ресурс] Дата обращения 01.10.2021

[6] Shumway R.H., Stoffer D.S. TimeSeries Analysis and Its Applications:With R Examples, 3rd Edition. -Springer, 2011. - 609 p

[7] E. J. Hannan. Multiple Time Series. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons Inc., New York, 2009

[8] Anava, Oren, et al. "Online learning for time series prediction." Conference on learning theory. PMLR, 2013.
[9] Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Mel-bourne, Australia.
[10] Vitaly Kuznetsov, Mehryar Mohri. Time series prediction and online learning. 29th Annual Conference on Learning Theory, PMLR 49:1190-1213, 2016.
[11] Dimitris Fotakis, Thanasis Lianeas, Georgios Piliouras, and Stratis Skoulakis. Efficient online learning of opti-mal rankings: Dimensionality reduction via gradient descent. In Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, 2020.
[12] Streaming linear regression https://spark.apache.org/docs/latest/mllib-linear-methods.html#streaming-linear-regression [электронный ресурс], Дата обращения 01.10.2021
[13] Clustering - RDD-based API https://spark.apache.org/docs/latest/mllib-clustering.html#streaming-k-means. [электронный ресурс] Дата обращения 01.10.2021
[14] Kozitsin V, Katser I, Lakontsev D. Online Forecasting and Anomaly Detection Based on the ARIMA Model. Applied Sciences. 2021; 11(7):3194. https://doi.org/10.3390/app11073194
[15] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. 2020. Deep Learning for Anomaly Detection: A Review. ACM Comput. Surv. 1, 1, Article 1 (January 2020), 36 pages. https://doi.org/10.1145/3439950

Статья получена 18 марта 2022.

Ельченков Роман Александрович, Национальный Исследовательский Ядерный Университет МИФИ, магистрант, Roma57995@gmail.com
Дунаев Максим Евгеньевич, Национальный Исследовательский Ядерный Университет МИФИ, аспирант, Max.dunaev@mail.ru
Зайцев Константин Сергеевич, Национальный Исследовательский Ядерный Университет МИФИ, профессор, KSZajtsev@mephi.ru

Time series forecasting in real-time streaming data processing

R.A. Elchenkov, M.E. Dunaev, K.S. Zaytsev

Annotation **— The purpose of this work is to study methods for predicting the values of time series when processing streaming data in distributed systems in real time. To do this, the authors propose a modification of the autoregressive model with a given AR order by adding to it the inheritance function of the previous values of the time series. The results of comparative experiments of the proposed modification, called Real-Time AR with classical AR and ARIMA, confirmed the effectiveness of the modification. This is especially evident in the presence of anomalies in the behavior of the real time series. The proposed modification of the algorithm allows not only to parallelize calculations, but also to configure the model on the fly in the Apache Spark ecosystem. To conduct experiments with the algorithms, a special data array was built - a data slice from 1000 measurements of the Apache Kafka server metrics log with one topic, two producers and one consumer. Anomalous fragments were artificially added to the array, differing in a large number of messages per second and/or message size. The values of the proposed data array were normalized and shifted by the average value over the training sample of the model pre-training. The results of applying the proposed algorithm in solving problems of predicting the values of time series showed that the presence of anomalies in the behavior of objects does not introduce significant distortions in the results of predicting values.**

Keywords **— Logs, Technological Platform, Machine learning, Statistical Methods, Apache Software Foundation, Apache Spark, Online Learning.**

REFERENCES

[1] Peter J Brockwell, Peter J Brockwell, Richard A Davis, and Richard A Davis. Introduction to time series and forecasting. Springer, 2016.

[2] A. Aldweesh, A. Derhab, and A. Z. Emam, “Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues, Knowledge-Based Systems, vol. 189, p. 105124, 2020

[3] Yagmur Gizem Cinar, Hamid Mirisaee, Parantapa Goswami, ´Eric Gaussier, AliAit-Bachir, and France Vadim Strijov. Time series forecasting using rnns: anextended attention mechanism to model periods and handle miss-ing values.CoRR, abs/1703.10089, 2017.

[4] Unified engine for large-scale data analytics https://spark.apache.org/ Reviewed 01.10.2021

[5] Apache Hadoop https://hadoop.apache.org/ Reviewed 01.10.2021

[6] Shumway R.H., Stoffer D.S. TimeSeries Analysis and Its Applications:With R Examples, 3rd Edition. -Springer, 2011. - 609 p

[7] E. J. Hannan. Multiple Time Series. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons Inc., New York, 2009

[8] Anava, Oren, et al. "Online learning for time series prediction." Conference on learning theory. PMLR, 2013.

[9] Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Mel-bourne, Australia.

[10] Vitaly Kuznetsov, Mehryar Mohri. Time series prediction and online learning. 29th Annual Conference on Learning Theory, PMLR 49:1190-1213, 2016.

[11] Dimitris Fotakis, Thanasis Lianeas, Georgios Piliouras, and Stratis Skoulakis. Efficient online learning of opti-mal rankings: Dimensionality reduction via gradient descent. In Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, 2020.

[12] Streaming linear regression https://spark.apache.org/docs/latest/mllib-linear-methods.html#streaming-linear-regression Reviewed 01.10.2021

[13] Clustering - RDD-based API https://spark.apache.org/docs/latest/mllib-clustering.html#streaming-k-means Reviewed 01.10.2021

[14] Kozitsin V, Katser I, Lakontsev D. Online Forecasting and Anomaly Detection Based on the ARIMA Model. Applied Sciences. 2021; 11(7):3194. https://doi.org/10.3390/app11073194

[15] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. 2020. Deep Learning for Anomaly Detection: A Review. ACM Comput. Surv. 1, 1, Article 1 (January 2020), 36 pages. https://doi.org/10.1145/3439950)