УДК: 004.054
DOI: 10.25559/SITITO.15.201904.945-953

Прогнозирование сбоев приложений по журналам состояния программных объектов

Р. Г. Бледнов[1], Н. А. Скворцов[2*]
[1] Московский государственный университет имени М.В. Ломоносова, г. Москва, Россия
119991, Россия, г. Москва, ГСП-1, Ленинские горы, д. 1
[2] Федеральный исследовательский центр «Информатика и управление» Российской академии наук, г. Москва, Россия
119333, Россия, г. Москва, ул. Вавилова, д. 44
[*] nskv@mail.ru

Аннотация

Программные сбои являются ощутимой и неизбежной проблемой при работе корпоративных программных систем. Обычно сбои обнаруживают при помощи слежения за превышением пороговых значений критических показателей системы. При этом предпринимать меры по предотвращению сбоев или их последствий часто не удаётся из-за недостатка времени на эти действия. Необходимо своевременно прогнозировать сбои, основываясь на журналах состояния приложений. Для этого были изучены различные подходы к прогнозированию сбоев, один из которых основан на обнаружении предшествующих им аномалий в данных состояния приложений. В работе предлагается несколько подходов, основанных на машинном обучении, для прогнозирования сбоев при помощи обнаружения аномалий. Лучшие результаты прогнозирования сбоев достигнуты при использовании метода градиентного бустинга над деревьями решений с применением метода скользящего окна и исключаемой преданомальной областью временного ряда в данных журналов. Это позволяет находить сбои в используемом наборе данных за приемлемое время до момента отказа системы. На случай отсутствия размеченных экспертом данных для обучения предложен подход к обучению без учителя с использованием изолирующих лесов и подход к автоматической разметке данных.

Ключевые слова: прогнозирование сбоев, программные системы, журналы состояния объектов, обнаружение аномалий.

Финансирование: Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований в рамках научных проектов № 18-07-01434_а «Методы и средства организации экспериментов в движимых гипотезами исследованиях в областях с интенсивным использованием данных» и № 18-29-22096_мк «Методы и средства решения задач анализа данных в распределенных вычислительных инфраструктурах в области нейрофизиологии».

Для цитирования: *Бледнов Р. Г., Скворцов Н. А.* Прогнозирование сбоев приложений по журналам состояния программных объектов // Современные информационные технологии и ИТ-образование. 2019. Т. 15, № 4. С. 945-953. DOI: 10.25559/SITITO.15.201904.945-953

Application Failure Prediction in Program Object State Logs

R. G. Blednov[1], N. A. Skvortsov[2*]
[1] Lomonosov Moscow State University, Moscow, Russia
1, Leninskie gory, Moscow 119991, Russia
[2] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia
44 Vavilov Str., Moscow 119333, Russia
[*] nskv@mail.ru

Abstract

Software failures are a tangible and imminent problem in enterprise software systems. Failures are usually detectable via monitoring threshold values of some critical indicators. At the same time, failure prevention or mitigation is often not possible due to a lack of time for any actions before a failure. It is necessary to predict failures in a timely manner using application status logs. For this purpose, different approaches to failure prediction have been studied, and one of them is based on the detection of foregoing anomalies in data on states of applications. The paper proposes several machine learning approaches to anomaly detection for failure prediction. The best results of failure prediction have been achieved with the gradient boosting method over decision trees with application of the sliding window method and excluding pieces of time series prior to anomalies in log data. This allows finding failures in considered data at a reasonable time before the system fails. In case of a lack of labeled data for training, an unsupervised approach using isolating forests and an automatic data labeling approach are proposed.

Keywords: failure prediction, software systems, object state logs, anomaly detection.

Funding: The research was carried out with the financial support of the Russian Foundation for Basic Research in as a part of scientific projects 18-07-01434_a «Methods and Means of Organizing Experiments in the Field of Hypothetical Research in Fields with Intensive Use of Data» and 18-29-22096_ мк «Methods and Tools for Solving Data Analysis Problems in Distributed Computational Infrastructures in the Field of Neurophysiology».

For citation: Blednov R. G., Skvortsov N. A. Application Failure Prediction in Program Object State Logs. *Sovremennye informacionnye tehnologii i IT-obrazovanie* = Modern Information Technologies and IT-Education. 2019; 15(4):945-953. DOI: 10.25559/SITITO.15.201904.945-953

Введение

Сбои в работе информационных систем являются большой проблемой в современном мире. В частности, во время работы программного обеспечения могут возникать сбои, которые мешают системе в целом функционировать должным образом. Современные методы борьбы со сбоями, в основном, направлены на локализацию причины сбоя в программе и служат для сбора дополнительной информации и исследования проблемы постфактум. Современные методы прогнозирования сбоев [1] срабатывают с недопустимо большой вероятностью ошибки первого рода и в момент, когда сбой уже начался. При этом не остаётся времени, чтобы предпринять меры по предотвращению сбоя или его последствий, такие как отключение проблемных программных компонентов, запуск резервных вычислительных ресурсов. Если для обработки и локализации сбоев существуют разные подходы, такие как непрерывный мониторинг значений основных показателей с сигнализаций о выходе их за порог максимальных или минимальных допустимых значений [1], то прогнозирование чаще всего связано с машинным обучением. Данными для обучения машины задачам прогнозирования сбоев могут быть не только отчеты о сбоях, иерархические текстовые описания сбоев, отчеты тестирования о шагах для воспроизведения сбоев, клиентские заявки об ошибках, но и автоматически генерируемые журналы разных видов о работе программ. В работах по теме обнаружения сбоев предлагаются различные подходы к классификации данных на нормальные и аномальные. Аномальные данные содержат признаки возникновения ситуаций, ведущих к сбоям. Методы обнаружения аномалий применяются как к отдельным шагам временных рядов в данных журналов, так и к временным окнам различной длины. В целом, задача прогнозирования программных сбоев изучена недостаточно.

Целью работы является разработка подхода к прогнозированию сбоев приложений по журналам состояния объектов. Прогнозирование сбоев осуществляется по журналам, содержащим данные о состоянии программных объектов, записанные с определенной периодичностью. Журналы являются единственным подробным и структурированным источником данных о состоянии работы приложений. По ним можно получать данные в реальном времени.

Заблаговременные информативные предупреждения о значительных изменениях в работе приложений должны помочь поставщикам услуг своевременно выявлять и предотвращать проблемы с производительностью и фатальные ошибки приложений и систем. При этом важно прогнозировать как можно большую часть реальных сбоев.

Для достижения поставленной цели были исследованы существующие методы и средства прогнозирования сбоев, разработаны и реализованы подходы к прогнозированию сбоев с использованием методов машинного обучения с учителем и без учителя и проведено сравнение результатов прогнозирования при различных подходах и различными методами.

Результатом с практической точки зрения является реализация различных сценариев от загрузки данных журналов до получения прогнозов сбоев на тестовых данных с набором значений гиперпараметров для выбора конкретных видов испытаний. По теме, связанной с данным исследованием, была защищена магистерская диссертация на факультете ВМК МГУ.

Далее в разделе 2 представлен обзор родственных работ с оценкой достоинств и недостатков подходов и обзор применимых методов машинного обучения для прогнозирования сбоев. Раздел 3 посвящён описанию и реализации разработанного подхода. В разделе 4 описаны данные, необходимые для исследования, и связанные с ними проблемы. В разделе 5 представлены результаты тестирования.

Обзор родственных работ и методов

Родственные работы

Для обнаружения и прогнозирования сбоев известным подходом является мониторинг, основанный на периодических запросах о состоянии работы приложения (health check), либо на периодически отсылаемой приложением информации о своей работоспособности (heartbeat)[1] [2][3]. При таких подходах приложения обычно разрабатываются с поддержкой такого мониторинга. Проверка состояния приложений при этом обычно основана на наборе правил, контролирующих выход показателей состояния, полученных от приложений, за рамки разрешённых интервалов. Недостатками таких подходов является то, что приложения должны поддерживать специальные функции для возможности мониторинга, а также то, что выход показателей состояния за пределы разрешённых интервалов происходит непосредственно перед сбоями, и не остаётся времени на противодействие сбою или его последствиям.

Известны работы, в которых задачи прогнозирования сбоев разного рода путем решают обнаружения аномалий во временных рядах. Например, популярен подход на основе архитектуры рекуррентных нейронных сетей с долгой краткосрочной памятью [4][5]. Такие модели делают прогноз, в основном, на один шаг вперед, что недостаточно в случае прогнозирования сбоев. Существуют подходы к обнаружению аномалий при помощи методов, основанных на функции расстояния между элементами данных [6] с разбиением временных рядов на окна различной длины [7]. Недостатками метрических методов являются их подверженность проклятию размерности и некоторые предположения относительно распределения данных. Из достоинств таких подходов можно отметить, что метод скользящего окна при анализе временных рядов позволяет захватить аномалии, протяжённые и развивающиеся во времени.

Среди опубликованных работ единицы посвящены прогнозированию сбоев приложений с использованием журналов состояния объектов систем. В статье [1] описывается подобная проблема, однако технические детали реализации, к сожалению, не раскрыты. Подход заключается в смещении наблюдения от прикладного уровня в сторону системного уровня, таким образом учитываются данные состояния как сервера приложений, таки и операционной системы. Авторами проектируется сервис мониторинга сбоев для физически распределенных систем и приложений. В качестве данных для анализа используются журналы состояний программных объектов. Данные внутренних состояний объектов собираются в журнал при помощи ин-

[1] Sun M., Tonn J.A. Network Health Monitoring Through Real-Time Analysis of Heartbeat Patterns From Distributed Agents. Patent no. US 2003/0061340 A1. Mar. 27, 2003. [Электронный ресурс]. URL: https://patentimages.storage.googleapis.com/49/4e/5b/34ef21acb10ac8/US20030061340A1.pdf (дата обращения: 12.09.2019).

фраструктуры управления и мониторинга Java-приложений JMX [8]. Обнаружение аномалий используется в качестве возможного подхода для обнаружения сбоев. Производится наблюдение по множеству показателей JVM [9] и нескольким показателям состояния операционной системы. С каждым показателем связан вес для большего или меньшего влияния на результат прогноза сбоев. Мониторинг основан на проверке значений по пороговым значениям [1], которые обновляются со временем, чтобы соответствовать эволюции системы или изменениям в прикладной среде. Однако такой подход малоэффективен для сложных систем. В частности, система не обнаруживает контекстуальные аномалии, в которых показатели могут быть аномальными, не выходя за пороговые значения.

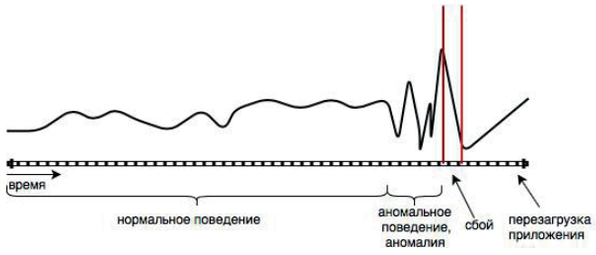
В работе [10] для поиска аномалий используется обучение различных моделей, основанных на нейронных сетях: развёртывающих (DNN), рекуррентных (RNN) и сетях с долгой краткосрочной памятью (LSTM). Для обнаружения аномалий прогнозируются ожидаемые значения показателей во временных рядах. Правила для обнаружения аномалий основываются на пороговых значениях разницы между фактическими и прогнозируемыми значениями. Отличительной чертой прикладной области работы является то, что временные ряды, для которых строился этот подход, обладают высокой периодичностью и автокоррелированностью. Прикладная область настоящей работы не обладает подобными характеристиками.

В работе [11] под понятием аномалия понимают любое непреднамеренное состояние системы, которое оказывает или может оказать негативное влияние на ее нормальную работу. Они включают фатальные состояния, приводящие к остановке узлов, например, ошибки нехватки памяти, а также более тонкие проблемы, такие как зависание потоков, которые продолжают потреблять ресурсы, в то время как остальная часть системы все еще работает в условиях ограниченных ресурсов. Для прогнозирования сбоев и обнаружения предшествующих им аномалий используются методы классификации, в частности, случайный лес и разновидность рекуррентных нейронных сетей. Недостатками этой работы является отсутствие какой-либо предобработки данных, направленной на улучшение качества выборки, и отсутствие подбора гиперпараметров для настройки используемых методов машинного обучения. Произведено и ниже представлено сравнение результатов этой работы с настоящим исследованием.

Анализ доступных методов прогнозирования

По результатам анализа родственных работ по теме прогнозирования сбоев можно сделать вывод о том, что задача прогнозирования сбоев по журналам состояния объектов в литературе освещена слабо. Поэтому при построении подхода были рассмотрены различные методы машинного обучения, применимые к задаче прогнозирования сбоев.

Для прогнозирования сбоев применяют поиск аномалий, предшествующих сбою. Штатная работа системы сопровождается нормальными значениями показателей, отражающих состояние её объектов. Сбою, которым может являться фатальное состояние, ведущее к остановке и требующее перезагрузки системы или приложения, предшествуют аномалии в поведении показателей состояния объектов (см. Рис. 1). Исследуя поведение системы, можно обнаруживать характерные аномалии в её показателях, и тем самым прогнозировать сбои.



Р и с. 1. Пример данных с аномалией и сбоем
F i g. 1. Example of data with anomaly and malfunction

Методы поиска аномалий включают подходы к обучению с учителем и без учителя [12]. Обнаружение аномалий методами обучения без учителя (unsupervised anomaly detection) [13] обычно используется для работы с данными, в которых отсутствует разметка экспертов предметной области, то есть нет информации об участках временных рядов, на которых система вела себя аномально.

Задачу прогнозирования сбоев по значениям некоторых показателей состояния системы можно рассматривать как вариант задачи двухклассовой классификации, в которой данные разбиваются метками классов на «нормальные» и «аномальные». В случае, если нет разметки данных для обучения, вначале проводится кластеризация. Так как количество нормальных примеров очевидно намного превосходят число аномальных, и информация о сбоях отсутствует, можно считать, что весь набор данных относится к нормальным (хотя, возможно, и зашумленным). Таким образом, на основе всего множества данных формируется кластер, который приобретает метку класса нормальных данных. Далее выбросам из кластера будет присваиваться метка класса аномальных данных.

Для обнаружения аномалий с обучением без учителя могут использоваться разные методы машинного обучения с различными достоинствами и недостатками, скоростью обучения и прогноза и точностью. Метод эллиптической оборачивающей кривой (Elliptic Envelope) [14] является легковесным методом. К сожалению, он имеет предположение о нормальном распределении данных, что делает этот метод непригодным для некоторых задач, в которых не известно начальное распределение данных. Метод выделения выбросов, основанный на локальной плотности точек (Local Outlier Factor) [15], не использует параметрических предположений и не нуждается во предварительном обучении, но это же является и недостатком, так как на этапе прогноза необходимо проанализировать много попарных расстояний, метод чувствителен к размерности данных. Одноклассовый метод опорных векторов (One Class SVM) [16] основан на построении гиперплоскости в пространстве признаков, отделяющей аномальные данные от нормальных и также не содержит параметрических предположений. Однако метод имеет немалую вычислительную сложность, а его точность зависит от выбора функции ядра классификатора, которую подобрать непросто. Метод изолирующего леса (Isolation Forest) [17] основан на случайных разбиениях выборки. Чем меньше требуется разбиений, чтобы изолировать наблюдение, тем более оно аномально. Эта процедура проводится несколько раз (bagging) для усреднения ошибки и получения устойчивых результатов. Метод быстро обучается и даёт устойчивые результаты.

При наличии информации о том, в какое время происходили сбои, можно рассмотреть модификации подхода, представ-

ленного выше. Из класса нормальных данных убирается часть данных с возможным сбоем в работе и аномалией передним, то есть определенный отрезок временного ряда перед сбоем и непосредственно во время сбоя. Делается предположение, что последние n тиков временного ряда перед сбоем аномальны, и данные за этот промежуток времени размечаются как аномальные. Данный подход равнозначен разметке выборки на основе известных сбоев, но без экспертной оценки реальной продолжительности аномалий в данных. Такая выборка может быть в немалой степени зашумленной, поскольку и класс нормальных данных может захватывать начало аномалий, и класс аномальных данных – участки нормальных данных без признаков аномалий. Тем не менее, такая автоматическая разметка позволяет получать данные для применения методов обучения с учителем. Также в настоящем исследовании для сравнения методов обнаружения аномалий без учителя и с учителем использовался размеченный экспертом набор данных. Качественная разметка данных является дорогостоящей процедурой. Если результаты прогнозирования сбоев без учителя или автоматическая разметка оказываются приемлемого качества, то потребность в экспертной разметке данных в реальных системах отпадает.

Методы поиска аномалий посредством обучения с учителем (supervised anomaly detection) [18] в прогнозировании сбоев используют в качестве обучающих данных примеры нормального и аномального поведения приложений для создания моделей классификации. Проблема классификации хорошо изучена [13] [18][19]. В [20] приведено сравнение наиболее часто используемых методов классификации с описанием выбора их настроек и значений суперпараметров и оценкой их работы с использованием различных подходов к оценке качества классификации. Следуя результатам, изложенным в [20], в данном исследовании в качестве методов классификации были применены случайный лес на деревьях решений, градиентный бустинг на деревьях, рекуррентные нейронные сети и метод опорных векторов.

Подход к решению задачи прогнозирования сбоев

В результате анализа родственных работы для прогнозирования сбоев используется поиск аномалий, предшествующих сбою. Ниже представлено подробное описание этапов подхода к прогнозированию сбоев.

Метод скользящего окна

Обычно в качестве объектов для обучения и тестирования алгоритмов поиска аномалий передаются единичные по временному шагу вектора признаков. Для учета временной составляющей данных применяется метод скользящего окна для перехода к новому признаковому пространству. Временной ряд разбивается на окна заданной длины n с перекрытием m < n значений, и в качестве признаков используются все данные за отрезок времени n (см. Рис. 2).

Р и с. 2. Пример разбиения временного ряда на окна

Fi g. 2. An example of splitting a time series into windows

В этом подходе важную роль играет подбор параметров длины окна n и размера перекрытия соседних окон m, а также способ присваивания окну метки нормальных или аномальных данных. Можно выделить несколько способов классификации окна как нормального или аномального. Например, окно размечается как аномальное, если в него входит хотя бы одно аномальное измерение. Согласно другому способу, окно аномально, если все измерения внутри него аномальны. Значения гиперпараметров ширины окна, ширины области их перекрытия и способа разметки окон определяются опытным путем.

Выбор методов поиска аномалий

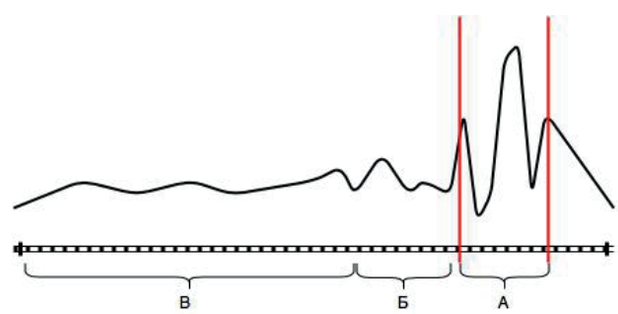
По результатам обзора доступных методов поиска аномалий эксперименты проводились для размеченных данных с применением методов градиентного бустинга и случайного леса. Для данных, на которых отсутствует разметка, применялся метод изолирующего леса. Также для данных без разметки применён подход с автоматической разметкой данных. В этом подходе для известных моментов сбоя делается предположение о продолжительности аномалии перед сбоем и производится соответствующая разметка данных.

Предварительная обработка и очистка данных

Вначале производится нормализация набора данных таким образом, чтобы математическое ожидание μ равнялось нулю, а дисперсия σ равнялась единице:

$$X_N = \frac{X - \mu}{\tilde{A}}.$$

Замечено, что во временных рядах размеченных данных на границе меток нормальных и аномальных данных бывают значения, которые уже существенно отличаются от предшествующих данных, однако эксперт по какой-то причине не посчитал их аномальными. На рисунке 3 приведён пример временного ряда, в котором область А размечена предметным экспертом как аномальные данные, а области Б и В – как нормальные. Для улучшения качества обучения были исключены из рассмотрения области Б, непосредственно предшествующие размеченным аномальным областям В. Таким образом, избегаются возможные ошибки разметки времени начала аномальных данных, и классы нормальных и аномальных данных лучше разграничиваются. Приносит ли это ощутимый положительный результат в точности прогнозирования сбоев, можно оценить экспериментальным путём.



Р и с. 3. Пример временного ряда с областями нормальных «В», аномальных «А» и пограничных значений «Б»

F i g. 3. An example of a time series with areas of normal “B”, anomalous “A” and borderline values “Б”

Выбор признаков для прогнозирования сбоев

Для качественного прогнозирования сбоев необходимо, чтобы в набор данных входили признаки, отвечающие за разносторонние критические показатели работы программных объектов и операционных систем. К ним относятся группы признаков, связанные с:

- подключениями к базам данных;
- состоянием потоков приложений;
- задержкой подключения к различным источникам данных;
- нагрузкой на процессор,
- состоянием различных областей памяти;
- количеством открытых файловых дескрипторов;
- статистикой сбора мусора (garbage collectors);
- количеством и видом ошибок.

На основе этого списка групп показателей выбирались необходимые показатели из размеченных и неразмеченных наборов данных о состоянии объектов приложений и операционной системы.

Выбор метрик для оценки качества методов прогнозирования

Для оценки точности прогнозирования использовались стандартные для двухклассовых задач метрики [21] количества верных положительных (TP — True Positive), верных отрицательных (TN — True Negative), ложных положительных (FP — False Positive), ложных отрицательных (FN — False Negative) прогнозов на тестовой выборке данных.

Минимизация количества верных отрицательных (FN) прогнозов в данной задаче является ключевой, так как с прикладной точки зрения это случаи, когда алгоритм пропускает аномалию, и при этом происходит реальный сбой, который может нести за собой существенные негативные последствия. Метрика количества ложных положительных (FP) прогнозов менее критична, поскольку ложный сигнал тревоги может повлечь расходы на излишние превентивные действия во избежание последствий возможных сбоев, но других последствий такие прогнозы не имеют.

Точность (precision) прогнозирования и его полнота (recall) в данном случае говорит о том, как часто прогноз сбоя оказывается верным, полнота – какую долю реальных сбоев удаётся спрогнозировать:

$$precision = \frac{TP}{TP + FP}, recall = \frac{TP}{TP + FN}.$$

Так как полнота прогнозирования зависит от количества ложных отрицательных прогнозов, ключевым в данной задаче является достижение её максимальных значений.

На базе метрик точности и полноты прогнозирования строится кривая их соотношения (PRC – Precision-Recall Curve) [22], и для оценки качества прогнозов используется площадь под этой кривой (AUC PRC – Area under Curve PRC).

Более популярной оценкой качества прогнозов является ROC-кривая – соотношение доли верных положительных прогнозов (TPR) и доли ложных положительных прогнозов (FPR).

$$TPR = recall = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN}.$$

При построении ROC-кривой количество верных и ложных срабатываний делится на объёмы соответствующих классов, поэтому площадь под ROC-кривой (ROC AUC) при дисбалансированных классах может давать видимость качественных результатов даже при слабой точности. Значит, в задаче прогнозирования сбоев, где классы нормальных и аномальных данных существенно дисбалансированы, лучше использовать оценку, основанную на кривой соотношения точности и полноты AUC PRC.

Подбор гиперпараметров

Для поиска наилучших решений экспериментально производится подбор гиперпараметров применяемых методов на различных этапах решения задачи:

- ширина преданомальной области на этапе предобработки данных для исключения её из обучения на размеченных данных выбирается путём перебора, решения задачи при фиксированных остальных параметрах обучения и сравнения метрик качества полученных результатов;
- параметры метода скользящего окна выбираются перебором и решением задачи при фиксированном подходе к предварительной обработке и фиксированным методе машинного обучения со сравнением качества результатов.

Реализация подхода

Для тестирования предлагаемого подхода был реализован прототип системы прогнозирования сбоев на языке Python версии 3.7 с использованием библиотек NumPy, SciPy, Pandas, scikit-learn, LightGBM [23].

Проведение экспериментов

Набор данных

В настоящем исследовании используются низкоуровневые показатели состояния приложений и операционной системы. Так как распространенным языком программирования корпоративных приложений является Java, то в качестве данных для прогнозирования сбоев выбраны журналы, хранящие значения показателей состояния объектов виртуальных машин Java. Это возможно благодаря тому, что показатели состояния любого объекта могут быть собраны при помощи инфраструктуры управления и мониторинга JMX (Java Manager Extension)[2] [24] посредством менеджеров MBean или MXBean.

Для экспериментов использовался сторонний набор данных, который предоставлен в свободный доступ в 2018 году[3] [25] и соответствует требованиям, предъявляемым к признакам для прогнозирования сбоев в настоящем исследовании. В наборе данных присутствует 831 показатель и 7.5 миллионов временных измерений с интервалом в одну минуту. Двадцать экземпляров одного приложения были активны на нескольких узлах одновременно. Поскольку приложения были одинаковы на всех рабочих узлах, их не различали при сборе данных. Требуемый набор групп признаков, отвечающих за различные аспекты работы приложений и операционных систем, описан выше. Исходя из него, для проведения экспериментов в насто-

[2] Todorova T., Opgenorth J., Weller J. Monitor viewer for an enterprise network monitoring system. Patent no. US 2005/0216585 A1. Sep. 29, 2005. [Электронный ресурс]. URL: https://patentimages.storage.googleapis.com/c2/b7/cf/49e62e4c7b326e/US20050216585A1.pdf (дата обращения: 12.09.2019).

[3] Huch F. Repository for feature data [Электронный ресурс]. URL: https://www.kaggle.com/anomalydetectionml/features (дата обращения: 12.09.2019).

ящей работе из стороннего набора данных был выбран 231 признак.

Следовательно, собранные данные могут содержать пропущенные значения из-за перезапусков системы и времени, когда система перегружена и не отвечает. Кроме того, набор измеряемых показателей со временем меняется. В качестве предобработки данных авторы набора выбирали подмножество показателей, для которых присутствовало большинство значений, и заполняли отсутствующие значения с помощью линейной интерполяции [11] [26]. Информация о том, что измерения были недоступны в определенные моменты времени, сохранялась с помощью специального флага.

Также до появления в свободном доступе описанного набора данных использовался синтетический набор данных 1999 года KDD-99 [27].

Тестирование методов прогнозирования сбоев без учителя

Рассмотренные выше алгоритмы обучения без учителя были применены на примере набора данных KDD-99. Для оценки качества была использована метрика площади под кривой точности/полноты (AUC PRC). В таблице 1 приведены результаты тестирования. По итогам исследования для тестирования на реальном наборе данных и сравнения с методами обучения с учителем выбран метод изолирующего леса. Все дальнейшие испытания проводились на реальном наборе данных, описанном выше.

Таблица 1. Результаты тестирования методов без учителя

Table 1. Test results of methods without a teacher

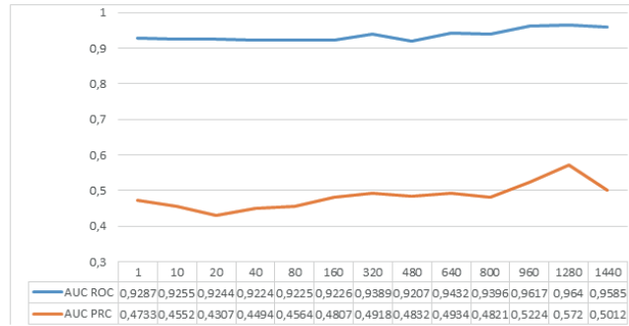
Метод обучения без учителя	Время обучения	Время прогноза	AUC PRC
Эллиптическая оборачивающая кривая	1 мин 7 сек	1 сек	0,9
Локальный уровень выброса	1 сек	40 мин	0,984
Одноклассовый метод опорных векторов	1 ч 40 мин	1 ч	0,999
Изолирующий лес	22 сек	28 сек	0,994

Тестирование методов прогнозирования сбоев с учителем

В случае присутствия размеченных экспертом данных для обнаружения сбоев могут применяться методы машинного обучения с учителем. Эксперименты проводились с целью ответа на вопрос, насколько обучение с учителем влияет на результаты прогнозирования по сравнению с методами над неразмеченными данными. В данном исследовании использовались методы градиентного бустинга и случайного леса.

Применена реализация метода градиентного бустинга над деревьями [28] на платформе LightGBM. Реализация обладает множеством гиперпараметров, которые подбираются под конкретную задачу эмпирическим путем. Прогнозирование сбоев при настройках гиперпараметров по умолчанию показало метрики качества AUC ROC = 0.8612 и AUC PRC = 0.4062. В результате подбора гиперпараметров в ряде испытаний были достигнуты результаты AUC ROC = 0.9287 и AUC PRC = 0.4733.

Далее были проведены эксперименты с удалёнными из рассмотрения перед аномальными областями временных рядов в соответствии с описанным ранее подходом к предобработке данных.



Р и с. 4. Результаты тестирования для подбора ширины исключаемой преданомальной области

F i g. 4. Test results for selection of width of the excluded preanomalous area

Предположение о целесообразности удаления областей, которые эксперты посчитали нормальными, но могут содержать начало аномальных данных, подтвердилось. На рисунке 4 представлены зависимости качества модели от ширины исключаемой преданомальной области. С увеличение ширины преданомальной области результаты улучшались. При ширине области в 1280 был получен прирост в 3 процентных пункта для AUC ROC и 9 для AUC PRC.

Подбор ширины скользящего окна производился с гиперпараметрами, давшими лучшие результаты в предшествующих экспериментах для подхода без исключения преданомальных областей данных и с использованием этого подхода. В экспериментах варьировались ширина окна (1, 2 и 4) и способ присваивания окну метки нормальных или аномальных данных. Способ, обозначенный «один» рассматривает окно как аномальное, если хотя бы один временной тик в окне имеет аномальные данные, и «все», если для этого все тики в окне должны быть аномальны.

Таблица 2. Результаты подбора параметров скользящего окна

Table 2. Results of sliding window settings

Ширина окна	Аномальность окна	Преданомальная область	AUC ROC	AUC PRC
1	-	0	0,9287	0,4733
2	Один	0	0,9263	0,4290
2	Все	0	0,9032	0,4430
4	Один	0	0,9499	0,4492
4	Все	0	0,9038	0,4941
1	-	1280	0,9640	0,5727
2	Все	960	0,9645	0,5244
2	Все	1280	0,9667	0,5828
4	Все	960	0,8982	0,4546
4	Все	1280	0,9640	0,5304
4	Все	1440	0.9226	0,4889

Исходя из таблицы 2, без использования подхода с исключением преданомальных областей окна шириной 4 дают хорошие результаты. Способ присваивания аномальности окон «все» достигает лучшего качества по сравнению со способом «один» для всех вариантов ширины окна. Проведены эксперименты, учитывающие исключение преданомальных областей шириной 960 и 1280, в сочетании с параметрами, дающими лучшие результаты без исключения преданомальных областей. По итогам этих экспериментов лучшие результаты получаются при ширине исключаемой преданомальной области 1280 тиков, способе присваивания аномальности окон «все» и ширине окна 1 и 2. Оценки качества моделей для окон шириной 1 со-

ставляют AUC ROC: 0.9640, AUC PRC: 0.5719, а для окон шириной 2 – соответственно AUC ROC: 0.9663, AUC PRC: 0.5828.

Таким образом, для метода градиентного бустинга на деревьях при применении подхода к прогнозированию сбоев с учителем, предложенного в данной работе, основной показатель качества - площадь под кривой точности и полноты - вырос с 0.406 до 0.572 и 0.582 для окон шириной 1 и 2 соответственно. Детальные эксперименты проводились также с применением метода случайного леса. В качестве реализации использовалась библиотека sklearn. Основываясь на результатах подбора ширины исключаемой преданомальной области при применении градиентного бустинга над деревьями, эксперименты для метода случайного леса были проведены только с учётом исключения преданомальных областей шириной 960 и 1280 тиков. Лучший достигнутый результат применения метода случайного леса составляет AUC ROC: 0.9763 и AUC PRC: 0.5017. Качество модели случайного леса оказалось, в среднем, хуже качества модели градиентного бустинга при тех же условиях.

Тестирование подхода к прогнозированию сбоев на данных с автоматической разметкой

В рамках предлагаемого подхода был рассмотрен случай отсутствия разметки, но известных моментах реальных сбоев. В этом случае возможна автоматическая разметка данных с фиксированным предполагаемым размером аномальной области. Проведённые эксперименты включали сравнение работы метода изолирующего леса [17] (реализации в библиотеке sklearn) в качестве подхода к обучению без учителя и метода градиентного бустинга над деревьями в качестве подхода к обучению с учителем. Оба подхода протестированы на одном и том же реальном наборе данных, причём в первом случае, он использовался как неразмеченный, то есть его разметка не использовалась вообще, а во втором случае использовались только данные о моментах сбоев и автоматическое разметка с автоматической разметкой фиксированной длины аномальной зоны перед сбоями. В экспериментах предполагалось, что аномалия продолжается в течение 30 минут перед сбоем.

Таблица 3. Результаты экспериментов с метода изолирующего леса на данных без

Table 3. The results of experiments with the isolating forest method on data

Ширина окна	Аномальность окна	Преданомальная область	Обучение без учителя без разметки		Обучение с учителем при авторазметке	
			AUC ROC	AUC PRC	AUC ROC	AUC PRC
1	-	1280	0,55	0,1275	0,62	0,2111
2	все	960	0,5724	0,1423	0,6173	0,2314
2	все	1280	0,5689	0,1401	0,6102	0,2294
4	все	960	0,6152	0,1612	0,6092	0,2192
4	Все	1280	0,5914	0,1593	0,6051	0,2025

Результаты экспериментов приведены в таблице 3. Они показывают, что подход с авторазметкой аномальных областей по данным о моментах реальных сбоев не даёт высоких результатов качества прогнозирования сбоев.

Выводы и заключение

Были исследованы подходы к прогнозированию сбоев на основе методов машинного обучения с учителем и без учителя. В результате проведённых экспериментов можно заключить, что прогнозирование сбоев методами машинного обучения без учителя на неразмеченных данных качество даёт слишком слабые результаты для реального его применения. При пороге 0.03 метод изолирующего леса даёт долю ложных отрицательных прогнозов 0.3236 и долю ложных положительных прогнозов 0.4632. Таким образом, в 32% случаев этот подход к прогнозированию пропускает реальные сбои.

Градиентный бустинг показал лучший результат из протестированных методов обучения с учителем. При пороге в 0.025 доля ложных отрицательных прогнозов составляет всего 0.0291, а доля ложных положительных прогнозов - 0.3264. Это означает, что система прогнозирования пропускает не больше 3 % реальных сбоев. Учитывая, что в данном наборе данных ширина аномальной области перед сбоем не бывает меньше 22 временных тиков по минуте, можно сделать вывод, что вероятность обнаружить аномалию за как минимум 20 минут до сбоя составляет 99,91%.

Также сделан вывод о том, методы обучения с учителем на автоматически размеченных данных работают лучше, чем методы обучения без учителя на неразмеченных данных. Такой подход требует подбора фиксированной предполагаемой ширины области аномальных данных перед сбоями. В условиях отсутствия размеченных экспертами данных этот подход может использоваться, однако результаты прогнозирования при этом существенно хуже, чем могли бы быть при экспертной разметке данных.

References

[1] Ceccarelli A., Zoppi T., Itria M., Bondavalli A. A Multi-layer Anomaly Detector for Dynamic Service-Based Systems. In: Koornneef F., van Gulijk C. (Eds.) *Computer Safety, Reliability, and Security. SAFECOMP 2014.* Lecture Notes in Computer Science, vol. 9337. Springer, Cham, 2015, pp. 166-180. (In Eng.) DOI: 10.1007/978-3-319-24255-2_13

[2] Wan J., Al-awlaqi M.A.A.H., Li M., O'Grady M., Gu X., Wang J., Cao N. Wearable IoT enabled real-time health monitoring system. *EURASIP Journal on Wireless Communications and Networking*. 2018; 2018:298. (In Eng.) DOI: 10.1186/s13638-018-1308-x

[3] Sterritt R. Pulse monitoring: extending the health-check for the autonomic grid. In: *IEEE International Conference on Industrial Informatics, 2003. INDIN 2003. Proceedings*, Banff, Alberta, Canada, 2003, pp. 433-440. (In Eng.) DOI: 10.1109/INDIN.2003.1300375

[4] Malhotra P., Vig L., Shroff G., Agarwal P. Long Short Term Memory Networks for Anomaly Detection in Time Series. In: *ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence, and Machine Learning*. Presses Universitaires de Louvain, Bruges (Belgium), 2015, p. 89. (In Eng.)

[5] Dau H.A., Ciesielski V., Song A. Anomaly Detection Using Replicator Neural Networks Trained on Examples of One Class. In: Dick G. et al. (Eds.) *Simulated Evolution and Learning. SEAL 2014.* Lecture Notes in Computer Science, vol 8886. Springer, Cham, 2014, pp. 311-322. (In Eng.) DOI: 10.1007/978-3-319-13563-2_27

[6] Goldstein M., Uchida S. A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *PloS ONE*. 2016; 11(4):e0152173. (In Eng.) DOI: 10.1371/journal.pone.0152173

[7] Ma J., Perkins S. Time-series novelty detection using one-

class support vector machines. In: *Proceedings of the International Joint Conference on Neural Networks, 2003,* vol. 3. Portland, OR, 2003, pp. 1741-1745. (In Eng.) DOI: 10.1109/IJCNN.2003.1223670

[8] Sullins B.G., Whipple M. JMX in Action. Manning Publications Co., 2002. (In Eng.)

[9] Meyer J., Downing T. Java Virtual Machine. O'Reilly in Cambridge, Mass., 1997. (In Eng.)

[10] Shipmon D.T., Gurevitch J.M., Piselli P.M., Edward S.T. Time Series Anomaly Detection; Detection of anomalous drops with limited features and sparse examples in noisy highly periodic data. *arXiv*. 2017; abs/1708.03665. (In Eng.)

[11] Huch F., Golagha M., Petrovska A., Krauss A. Machine learning-based run-time anomaly detection in software systems: An industrial evaluation. In: *2018 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE)*, Campobasso, 2018, pp. 13-18. (In Eng.) DOI: 10.1109/MALTESQUE.2018.8368453

[12] Shkodyrev V.P., Yagafarov K.I., Bashtovenko V.A., Ilyina E.E. The Overview of Anomaly Detection Methods in Data Streams. *CEUR Workshop Proceedings*. 2017; 1864:33. Available at: http://ceur-ws.org/Vol-1864/paper_33.pdf (accessed 12.09.2019). (In Russ. abstract in Eng.)

[13] Song J., Takakura H., Okabe Y., Nakao K. Toward a more practical unsupervised anomaly detection system. *Information Sciences.* 2013; 231:4-14. (In Eng.) DOI: 10.1016/j.ins.2011.08.011

[14] Aganagic M., Okounkov A. Elliptic stable envelopes. *arXiv*. 2016; 1604.00423. (In Eng.)

[15] Ma Y., Shi H., Ma H., Wang M. Dynamic process monitoring using adaptive local outlier factor. *Chemometrics and Intelligent Laboratory Systems*. 2013; 127:89-101. (In Eng.) DOI: 10.1016/j.chemolab.2013.06.004

[16] Li K-L., Huang H-K., Tian S-F., Xu W. Improving one-class SVM for anomaly detection. In: *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*, Xi'an, 2003, pp. 3077-3081, Vol. 5. (In Eng.) DOI: 10.1109/ICMLC.2003.1260106

[17] Liu F.T., Ting K.M., Zhou Z.H. Isolation forest. In: *2008 Eighth IEEE International Conference on Data Mining*, Pisa, 2008, pp. 413-422. (In Eng.) DOI: 10.1109/ICDM.2008.17

[18] Görnitz N., Kloft M., Rieck K., Brefeld U. Toward Supervised Anomaly Detection. *Journal of Artificial Intelligence Research.* 2013; 46:235-262. (In Eng.) DOI: 10.1613/jair.3623

[19] Aggarwal C.C. An Introduction to Outlier Analysis. *Outlier Analysis.* Springer, Cham, 2017, pp. 1-34. (In Eng.) DOI: 10.1007/978-3-319-47578-3_1

[20] Caruana R., Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. In: *Proceedings of the 23rd international conference on Machine learning (ICML '06).* Association for Computing Machinery, New York, NY, USA, 2006, pp. 161-168. (In Eng.) DOI: 10.1145/1143844.1143865

[21] Wang W., Zhang X., Gombault S., Knapskog S.J. Attribute Normalization in Network Intrusion Detection. In: *2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, Kaohsiung, 2009, pp. 448-453. (In Eng.) DOI: 10.1109/I-SPAN.2009.49

[22] Saito T., Rehmsmeier M. The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLoS ONE.* 2015; 10(3):e0118432. (In Eng.) DOI: 10.1371/journal.pone.0118432

[23] Oliphant T.E. Python for Scientific Computing. *Computing in Science and Engineering*. 2007; 9(3):10-20. (In Eng.) DOI: 10.1109/MCSE.2007.58

[24] Choi J-W., Lee K-H. A Web-based management system for network monitoring. In: *IEEE Workshop on IP Operations and Management*, Dallas, TX, 2002, pp. 98-102. (In Eng.) DOI: 10.1109/IPOM.2002.1045763

[25] Witten I.H., Frank E., Hall M.A., Pal C.J. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2016. (In Eng.)

[26] Moritz S., Bartz-Beielstein T. imputeTS: time series missing value imputation in R. *The R Journal.* 2017; 9(1):207-218. (In Eng.) DOI: 10.32614/RJ-2017-009

[27] Tavallaee M., Bagheri E., Lu W., Ghorbani A.A. A detailed analysis of the KDD CUP 99 data set. In: *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, 2009, pp. 1-6. (In Eng.) DOI: 10.1109/CISDA.2009.5356528

[28] Mason L., Baxter J., Bartlett P., Frean M. Boosting Algorithms as Gradient Descent. In: Solla S.A., Leen T.K., K.-R. Müller (Eds.) *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA, 2000, pp. 512-518. Available at: http://papers.nips.cc/paper/1766-boosting-algorithms-as-gradient-descent.pdf (accessed 12.09.2019). (In Eng.)

Submitted 12.09.2019; revised 20.11.2019; published online 23.12.2019.
Поступила 12.09.2019; принята к публикации 20.11.2019; опубликована онлайн 23.12.2019.

Об авторах:

Бледнов Роман Германович, студент факультета вычислительной математики и кибернетики, Московский государственный университет имени М.В. Ломоносова (119991, Россия, г. Москва, ГСП-1, Ленинские горы, д. 1), ORCID: http://orcid.org/0000-0002-2378-4050, blednovrg@gmail.com

Скворцов Николай Алексеевич, научный сотрудник Института проблем информатики, Федеральный исследовательский центр «Информатика и управление» Российской академии наук (119333, Россия, г. Москва, ул. Вавилова, д. 44), ORCID: http://orcid.org/0000-0003-3207-4955, nskv@mail.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Roman G. Blednov, student of the Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University (1, Leninskie gory, Moscow 119991, Russia), ORCID: http://orcid.org/0000-0002-2378-4050, blednovrg@gmail.com

Nikolay A. Skvortsov, Researcher of the Institute of Informatics Problems, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences (44 Vavilov Str., Moscow 119333, Russia), ORCID: http://orcid.org/0000-0003-3207-4955, nskv@mail.ru

All authors have read and approved the final manuscript.