**Детектирование аномалий и прогнозирование нагрузки в распределенных системах с использованием машинного обучения**

Игорь Юнаш, Александра Лежанкина

Аннотация:

В рамках данной работы была разработана и внедрена в серверную инфраструктуру отдела мобильных приложений НИУ ВШЭ система детектирования аномалий и прогнозирования нагрузки. Авторами работы были проанализированы несколько имеющихся академических реализаций этой задачи, использующих методы машинного обучения, и написана своя, на базе аналитической базы данных ClickHouse и бекенда на Kubernetes.

В продуктовой среде выполнения кода у разработчиков и владельцев продукта всегда возникает необходимость в отслеживании ошибок и прогнозировании нагрузки на серверы. В зависимости от результатов анализа может потребоваться увеличить количество серверов, или же наоборот, уменьшить их, чтобы сэкономить бюджет. Также в рамках анализа можно отследить некорректное исполнение кода, падение серверов или другие неожиданные ошибки, которые не были предусмотрены при разработке программы, то есть - аномалии. Аномалией считаем некое отклонение от обычного состояния в работе системы, чаще всего это связано с проблемой в некоторой области или редким событием, которое еще невозможно обработать. Таким образом, мы можем классифицировать аномалии на два случая: обнаружение выбросов, то есть некорректных данных, которые не соответствуют действительности, они могут возникать в результате ошибок в вычислениях, опечатках, могут являться шумовыми объектами, возникающими в результате неверной классификации; а также обнаружение новизны, то есть данных, которые кардинально отличаются своими свойствами от уже имеющихся объектов в выборке, но не противоречат логике системы. При выявлении выбросов алгоритм направлен на поиск слишком высоких/низкий или волатильных значений, относительно имеющихся в выборке. При выявлении новизны алгоритм оценивает, насколько новое значение похоже на то, что содержится в выборке.

Необходимость в оперативном обнаружении аномалий заключается в нескольких вещах: в первую очередь аномалия может являться большой технической проблемой, которая влияет на состояние продукта, возможность доступа к нему или на функционирование определенных его частей. В то же время, сам источник данных, на основании информации из которого была найдена аномалия, может функционировать корректно, косвенно указывая на ошибки другой части инфраструктуры. В отличие от ручных способов детектирования аномалий (это может быть простое визуальное наблюдение за состоянием сервисов, мониторинг сообщений в пользовательскую поддержку и другое), или методов мониторинга черным ящиком (например, с помощью автоматических средств проверки работоспособности сервисов, наподобие Uptime Robot), автоматизированные средства детектирования аномалий позволяют получать максимально оперативную и подробную информацию о проблеме, что позволяет локализовать ее быстрее и вернуть систему к нормальному состоянию.

Идея создания подобной системы не нова и мы можем посмотреть на существующие аналоги. Некоторые компании разрабатывают свои закрытые внутренние реализации подобных систем с тесной интеграцией во внутренние технические и организационные процессы. Также на рынке есть публичные системы, которыми может воспользоваться каждый. Например, модуль машинного обучения в системе Kibana, которая основывается на базе данных ElasticSearch. (CustomRules). В ней есть возможность как поиска аномалий на основе собранных метрик, так и прогнозирования нагрузки на срок до 8 недель. Проблема этого и некоторых других продуктов заключается в финансовой необходимости постоянной оплаты за количество анализируемых данных или за количество машин, на которых выполняется анализ. В то же время эти системы позволяют добиваться результата с наименьшими организационными затратами, которые заключаются в запуске основного координационного блока системы, модулей для специализированных задач, которые зачастую запускаются на отдельных машинах, чтобы иметь возможность масштабироваться, в конфигурации сбора логов, чтобы они попадали в соответствующие индексы баз данных со всех машин, и легковесной настройке непосредственного анализа методами машинного обучения.

Vk. Скользящее среднее.

Кластеризация

Eliptic Envelope

Объект для обнаружения промахов в гауссовом распределенном наборе данных.

Первый метод — это обычный [**SVM**](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BE%D0%BF%D0%BE%D1%80%D0%BD%D1%8B%D1%85_%D0%B2%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BE%D0%B2), который отделяет выборку от начала координат. Идея немного сомнительна, но оказалась довольно работоспособной (см. рис. 10). Здесь правда не так много разнообразия в выборе параметров, как при решении задач классификации, поскольку в качестве ядра подойдёт лишь rbf (радиальные базисные функции), все остальные ядра показывают феноменально отвратительный результат. Интересно, что многие годы задачи детектирования поломок сложных механизмов решались именно с помощью OneClassSVM, почему-то без рассмотрения альтернатив. Полезно помнить, что**OneClassSVM это скорее алгоритм поиска новизны**, а не выбросов, т.к. «затачивается» под обучающую выборку.

Если мы имеем дело с задачей novelty detection, где для тренировки нам доступны только "хорошие" наблюдения без аномалий, то мы можем воспользоваться этой моделью и научиться для каждого нового наблюдения говорить, является ли оно аномальным или нет.

Общая идея: преобразовать признаковое пространство и провести разделяющую гиперплоскость так, чтобы наблюдения лежали как можно дальше от начала координат:

Больше для новизны

[**Изолирующий лес**](https://cs.nju.edu.cn/zhouzh/zhouzh.files/publication/icdm08b.pdf)  (Isolation Forest) – это одна из вариаций идеи случайного леса. Как всегда: простая и надёжная.

* Состоит из деревьев
* Каждое дерево строится до исчерпании выборки
* Для построения ветвления в дереве: выбирается случайный признак и случайное расщепление
* Для каждого объекта мера его нормальности – среднее арифметическое глубин листьев, в которые он попал (изолировался)

Логика алгоритма простая: при описанном «случайном» способе построения деревьев выбросы будут попадать в листья на ранних этапах (на небольшой глубине дерева), т.е. выбросы проще «изолировать» (напомним, что дерево строится до тех пор, пока каждый объект не окажется в отдельном листе). Алгоритм хорошо отлавливает именно выбросы (см. рис. 12).

Нейросети автоэнкодеры (из доклада Павла Филимонова в лаборатории Касперского )

Говоря о теоретической составляющей работы, нельзя не упомянуть временные ряды. Временной ряд представляет собой статистический материал о значении какого-либо параметра исследуемого процесса, собранный в разные моменты времени. В нашем случае мы планируем рассмотреть датасет, содержащий логи работы системы: место обращения пользователя, его ip-адрес, время выполнения действия, код ответа сервера и факт наличия ошибки. Таким образом, данные подходят под определение временного ряда: имеем вектор, где каждый элемент является показанием логов работы системы в момент времени t, рассматриваем только тот случай, когда такие показания возможно получить. Необходимо сопоставить каждому моменту t в соответствие какое-то значение, которое бы показывало, насколько в данный момент нетипичны значения временного ряда. При этом сами аномалии могут не быть похожи друг на друга, то есть их не удастся сгруппировать в один класс. Мы будем использовать при финальном тестировании достаточно объемный датасет, при желании вмещающий в себя информацию за месяцы работы. В таких случаях, когда временной ряд оказывается очень длинным, он для удобства разбивается на подряды меньшего размера, исходя из вычислительных мощностей.

Перейдем к более техническому описанию нашего подхода к работе. В общем случае существует три подхода к обнаружению аномалий: supervised (“с учителем”), semi-supervised, unsupervised (“без учителя”) и неконтролируемые методы. В первом случае это чем-то напоминает бинарную классификацию. Мы имеем в качестве обучающей выборки уже размеченный датасет, содержащий однозначные метки на точках, являющихся выбросами. Глядя на наш датасет, можно уже сейчас сказать, что нам будет не очень удобно вручную размечать данные для обучения, ввиду их внушительного объема. Semi-supervised имеет достаточно интересную характеристику: такой метод подразумевает то, что выборка, на которой мы обучаемся, гарантированно не содержит аномальных данных, предполагается, что они будут встречаться далее при тестировании. В третьем случае мы заранее не располагаем информацией о характеристиках аномалий. При детектировании их используются введенные метрики “расстояния” или “плотности”. Тогда подходят, например, алгоритмы k-means, OneClassSVM, Isolation Forests. Среди самых популярных алгоритмов в этой области вообще можно выделить: метод опорных векторов, нейронные сети, метод ближайших соседей и решающие деревья. Подробный сравнительный анализ будет представлен уже в основной работе, сейчас можно в общем сказать, что методы детектирования могут отличаться, в зависимости от качества выборки, от объема данных, от желаемой скорости работы алгоритма, количества и характера потенциальных аномалий. Большинство разработанных систем отдают свое предпочтение нейронным сетям. Среди самых распространенных можно выделить: Long-Short Term Memory (LSTM) - востребован за счет умения запоминать долгосрочные зависимости, применяется в анализе временных рядов; Иерархическая временная память (HTM) - основана на исследованиях в нейронауке, способная изучать и запоминать долгосрочные зависимости. Качество подхода определяется точностью полученного результата. Поэтому мы также ставим перед собой задачу разобраться, с какими аномалиями нам предстоит работать, определить для себя подход к их обнаружению и выбрать оптимальный алгоритм или комбинацию нескольких. Стоит отметить, что многие источники доказывают то, что самым эффективным подходом является комбинация нескольких алгоритмов. Вообще говоря, помимо методов машинного обучения в детектировании аномалий также применяются графические и статистические методы, но в своей курсовой работе мы решили ограничиться машинным обучением, так как уже сталкивались с ним и заинтересованы в его применении в промышленной разработке. Для облегчения аналитической части работы мы постараемся реализовать еще и визуализацию логов работы системы, чтобы сразу оценить характер аномалий.

В качестве результатов работы планируется выбрать теоретический подход для детектирования аномалий и прогнозирования нагрузки и реализовать на его основе интегрируемую систему, которая бы позволяла на основании логов серверов микросервисной архитектуры на Kubernetes выполнять свои задачи. Также планируется интегрировать эту систему в серверную инфраструктуру отдела мобильных приложений дирекции по порталу и мобильным приложениям Национального исследовательского университета «Высшая школа экономики», которая в данный момент занимается обработкой запросов от 5 мобильных приложений в суммарном количестве около 500 000 запросов в сутки. Помимо прочего ставим для себя цель разобраться в более продвинутом использовании машинного обучения, изучить существующие статьи и математическое обоснование эффективности предложенных алгоритмов. Хотелось бы, чтобы наша система имела дальнейшее развитие и применения и оказалась полезна не только внутри университета, но и за его пределами.

В итоге мы планируем построить рабочий процесс таким образом:

1. НОЯБРЬ - ЯНВАРЬ: постановка задачи, поиск источников литературы и сравнение существующих аналогов;
2. ЯНВАРЬ - ФЕВРАЛЬ: анализ алгоритмов и выбор оптимального для решения поставленной задачи, теоретическое обоснование эффективности выбранного метода;
3. МАРТ: реализация механизма визуализации и детектирования аномалий, прогнозирования нагрузки в распределенных системах с помощью методов машинного обучения;
4. МАРТ - АПРЕЛЬ: интеграция реализованной системы в серверную инфраструктуру отдела мобильных приложений Национального исследовательского университета «Высшая школа экономики»;
5. АПРЕЛЬ: тестирование реализованной системы, анализ полученных результатов, написание текста курсовой работы;
6. МАЙ: успешная защита.

Источники:

1. Линдигрин А.Н. Сравнительный анализ методов машинного обучения в задачах обнаружения сетевых аномалий [Текст]
2. Примечание о поиске аномалий - Машинное обучение - 2021 (<https://ru.sciencewal.com/68149-a-note-about-finding-anomalies-f9cedee38f0b-54>) [Интернет-ресурс]
3. Романов Н.А. Детектирование аномалий во временных рядах при помощи глубоких нейронных сетей [Текст]

References:

* <https://itnan.ru/post.php?c=1&p=532152>
* <https://ics-cert.kaspersky.ru/reports/2018/01/16/mlad-machine-learning-for-anomaly-detection/>
* <https://www.elastic.co/what-is/elasticsearch-machine-learning>
* <https://sematext.com>
* <https://research.google.com/pubs/archive/46283.pdf>
* <https://towardsdatascience.com/time-series-analysis-for-machine-learning-with-python-626bee0d0205>