Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт (государственный университет)»

Факультет радиотехники и кибернетики
Кафедра теоретической и прикладной информатики

Направление подготовки/специальность: 03.04.01 Прикладные математика и физика
Направленность (профиль) подготовки: Математические и информационные технологии
Форма обучения: очная

Допущен к ГИА
Руководитель учебного подразделения
_____/_____
(подпись) *расшифровка подписи*
«____» _____20____г.

Автоматический поиск аномалий во временных рядах

(магистерская диссертация)

Обучающийся: Соболев Константин Викторович

_____
(подпись обучающегося)

Научный руководитель: Тормасов Александр Геннадьевич , д.ф.-м.н.

_____
(подпись научного руководителя)

Научный консультант: Емельянов Павел Владимирович, к.ф.-м.н.

_____
(подпись научного консультанта)

Москва 2018

Аннотация

Данная работапосвящена проблеме обнаружения и классификации аномалий для данных временных рядов. Некоторые из важных применений обнаружения аномалий временных рядов - это здравоохранение, обнаружение мошенничества и распознание сбоев в работе системы.

Несмотря на обширную работу по обнаружению аномалий [2] во временных рядах, большинство методов ищут отдельные объекты, которые отличаются от обычных объектов, но не учитывают особенности последовательности данных.

В данной работе анализируются современные методы обнаружения аномалий и классификации временных рядов. Для обнаружения аномалий используются методы, основанные на прогнозировании: статистические и использующие глубокие нейронные сети. Для классификации использовались классические методы классификации, принимающие статистические параметры рядов. А так же был предложен метод, использующий сверточные нейронные сети, ранее не применявшийся в данной области.

Содержание

1 Введение

Аномалии - это образцы данных, которые не соответствуют четко определенному понятию нормального поведения. Важность выявления аномалий обусловлена тем фактом, что аномалии в данных приводят к значительной и действенной информации в самых разных областях применения [2]. Например, аномальная схема трафика в компьютерной сети может означать, что компьютер взломан и отправляет конфиденциальные данные неавторизованному адресату [3]. Аномалии на МРТ могут указывать на наличие злокачественных опухолей [4] или аномалии в данных транзакций по кредитной карте, которые могут указывать на кредитную карту или кражу личных данных [5]. Обнаружение аномалий было изучено несколькими исследовательскими сообществами для решения проблем в разных областях применения [2]. Во многих областях, таких как безопасность,обнаружение мошенничества, медицинское обслуживание и т.д.

Временной ряд - это собранный в разные моменты времени статистический материал о значении каких-либо параметров (в простейшем случае одного) исследуемого процесса. Другими словами, упорядоченная во времени последовательность значений какого-либо датчика. Такие данные легко собрать, и они имеют высокую значимость в определении состояния системы, так как могут быть получены в реальном времени. Таким образом остро встает задача анализа данного типа данных.

В данной работе рассматриваются различные алгоритмы определения аномалий во временных рядах и их классификации. Рассматривается конкретная постановка задачи - поиск и классификация аномалий в многомерных временных рядах. А рассмотренные алгоритмы и разработанный прототип характерны тем, что не привязаны к данным, с которыми работают. Алгоритм может быть полностью переобучен под другую задачу будь то анализ кардиограммы, обнаружение сбоев в работе промышленного завода или контроль работы вычислительной системы.

2 Обзор существующих алгоритмов поиска аномалий

В данном разделе исследуется проблема обнаружения аномалий для одномерных временных рядов. Существует множество работ по поиску аномалий [2], большинство из которых использует подход поиска объектов, выделяющихся из общего распределения. Данная стратегия хорошо работает при условии независимости объектов в обучающей и тестовой выборках. Временные ряды отличаются тем, что в них данное условие может нарушаться. Рассмотрим пример, приведенный на рисунке 1, который соответствует данным ЭКГ пациента, который обычно является периодическим. Выделенная область обозначает аномалию, потому что такое же низкое значение существует для аномально долгого времени (соответствующего сокращению предсердий). Обратите внимание, что низкое значение само по себе не является аномалией, поскольку оно происходит в разных соседних местах. Следовательно, если данные обрабатываются как совокупность значений амплитуды, игнорируя их временной аспект, аномалия не может быть обнаружена.

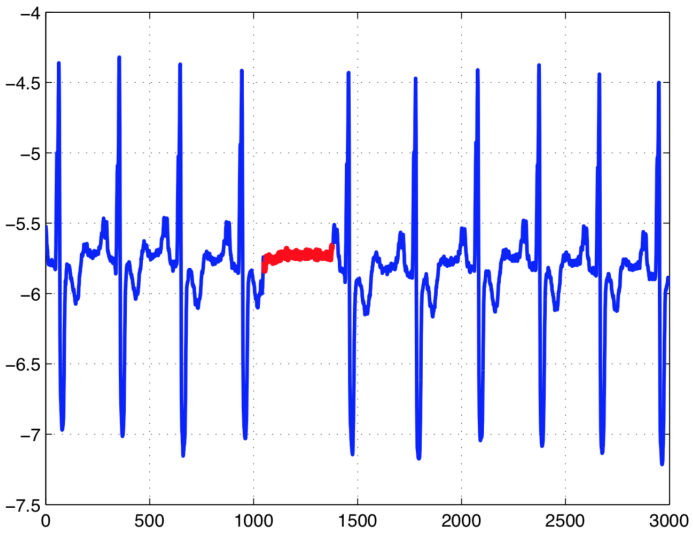


Рис. 1: Пример аномалии

Задача обнаружения аномалий во временных рядах не так хорошо изучена, как проблема обнаружения традиционных аномалий. Несколько исследований: Чандола и другие [2], Агиман и др. [6] и Ходж и др. [9] обсуждают проблему

обнаружения аномалий. Для символических последовательностей было предложено несколько методов обнаружения аномалий. Они обсуждаются в обзоре Чандола и др. [7]. Хотя для одномерных и многомерных временных рядов существует ограниченное число методов.

Существующие исследования по обнаружению аномалий для временных рядов были фрагментированы в разных областях применения, без хорошего понимания того, как разные методы связаны друг с другом и каковы их сильные и слабые стороны. Часть обзора данной работы представляет собой попытку обеспечить всестороннее понимание и структурированный обзор исследований методов обнаружения аномалий для временных рядов, охватывающих множество областей исследований и областей приложений. Мы попытаемся понять, как эффективность этих методов связана с различными аспектами проблемы, такими как характер данных, характер аномалий и т. д.

Данная глава организованна следующим образом. В разделе 2.1 обсуждаются некоторые важные применения поиска аномалий временных рядов. В разделе 2.2 описана формальная постановка задачи поиска аномалий временных рядов, а в разделе 2.3 рассматриваются проблемы, связанные с этой задачей. Различные типы временных рядов описываются в разделе 2.4. В разделе 2.5 дается краткий обзор существующих технологий.

2.1 Применения поиска аномалий во временных рядах

Некоторые из важных применений обнаружения аномалий временных рядов:

1. Обнаружение аномальных импульсов сердечного ритма с использованием данных ЭКГ [8]: Обычно данные ЭКГ можно рассматривать, как периодические временные ряды. Аномалия в этом случае была бы несоответствующей картиной, например, с точки зрения периодичности или амплитуды, что может указывать на проблему со здоровьем.

2. Обнаружение нападений в рекомендательных системах: шиллинговые атаки, в которых злоумышленники вводят предвзятые рейтинги, чтобы повлиять на будущие рекомендации.

3. Обнаружение аномалий в данных датчиков компьютерных систем. Типичное поведение системы полетов характеризуется информацией данных датчиков о разных параметрах, которые изменяются в ходе работы системы. Любое отклонение от типичного поведения системы аномально и может являться взломом или поломкой системы.

4. Нарушения экосистемы с использованием геологических данных, таких как растительность и температура.

2.2 Постановка задачи

Задача обнаружения аномалий во временных рядах может быть рассмотрена по-разному. В данном разделе приведены три возможных определения / постановки.

2.2.1 *Постановка задачи 1*: Обнаружение контекстных аномалий в временных рядах

В этой постановке задачи обнаружения аномалий во временном ряду аномалии представляют собой отдельные отрезки временного ряда, которые являются аномальными в определенном контексте, но не иначе. Это широко исследуемая проблема в сообществе статистики [12]. На рисунке 2 показан один такой пример для временного ряда температур, который показывает месячную температуру области за последние несколько лет. Температура 35F может быть нормальной в течение зимы (в момент времени t_1) в этом месте, но такое же значение в течение лета (в момент времени t_2) было бы аномалией.

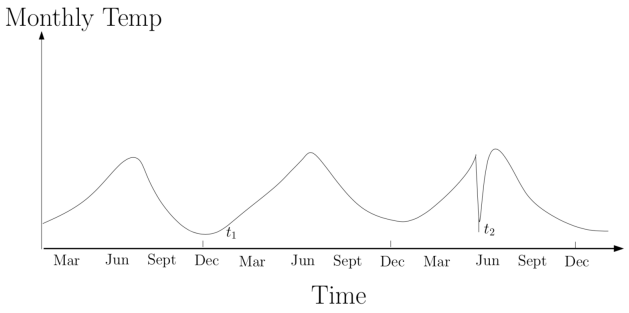


Рис. 2: Контекстная аномалия в момент времени t_2

2.2.2 *Постановка задачи 2*: Обнаружение аномальной подпоследовательности в заданном временном ряду

Другая постановка задачи обнаружения аномалий пытается найти аномальную последующую зависимость относительно заданной длинной последовательности (временного ряда). Рисунок 1 является примером временного ряда, содержащего аномальную подпоследовательность (выделенную область), где низкое значение ряда существует аномально долгое время, хотя низкое значение само по себе не является аномалией, как это происходит в нескольких других местах. Эта задача соответствует неконтролируемой обучающей среде из-за отсутствия помеченных данных для обучения, но большая часть длинной последовательности (временных рядов) считается нормальной. Если аномальная подпоследовательность имеет единичную длину, то эта задача эквивалентна нахождению контекстных аномалий во временном ряде, что является постановкой задачи 1.

2.2.3 *Постановка задачи 3*: Обнаружение аномальных временных рядов с учетом базы данных временных рядов

Третья постановка задачи обнаружения аномалий пытается определить, является ли временной ряд тестовой выборки аномальным по отношению к базе данных временных рядов обучения. Эта база данных может быть двух типов. Один тип состоит только из нормальных временных рядов (смешенное обучение [2]). В другом варианте он состоит из неразмеченных временных рядов (обнаружение без учителя) как нормальных, так и аномальных данных, но предполагается, что большинство является нормальным.

2.3 Проблемы поиска аномалий во временных рядах

Некоторые основные проблемы, связанные с обнаружением аномалий во временных рядах:

1. Существует множество способов, с помощью которых может быть определена аномалия, встречающаяся во временном ряде. Событие внутри временных рядов может быть аномальным; подпоследовательность внутри вре-

менного ряда может быть аномальной; или целый временной ряд может быть аномальным по отношению к набору нормальных временных рядов.

2. Для обнаружения аномальных подпоследовательностей точная длина подпоследовательности часто неизвестна.

3. Временные ряды для обучения и тестирования могут иметь разную длину.

4. Наилучшие меры сходства / различия, которые можно использовать для разных типов временных рядов, определить нелегко. Простые меры, такие как евклидово расстояние, не всегда хорошо работают, поскольку они очень чувствительны к выбросам, и они не действительны когда временные ряды имеют разную длину.

5. Точность многих алгоритмов обнаружения аномалий сильно подвержены шуму в данных, поскольку отделение аномалий от шума является сложной задачей.

6. Временные ряды в реальных приложениях обычно имеют большой размер (длину), а по мере увеличения длины вычислительная сложность также возрастает.

7. Точность многих алгоритмом обнаружения аномалий сильно зависит от соответствия масштабов временных рядов, что не верно для большинства данных.

2.4 Типы временных рядов

Алгоритмы поиска аномалий в данной работе используют данные для обучения, чтобы изучить модель для нормального поведения и оценить аномальность тестовых данных на основе модели. Таким образом, производительность любой техники зависит от характера нормальных временных рядов, а также от аномальных временных рядов. Различия между нормальным и аномальным временными рядами обсуждаются в разделе 2.2.

Рассмотрим две ключевые характеристики временных рядов: периодичность и синхронность. Сочетание этих свойств даст 4 разных типа времен-

ных рядов. Нам дан набор данных из n нормальных временных рядов $T = t_1, t_2, ., , , t_n$, которые можно рассматривать как:

- *Периодический и синхронный*: это самый простой пример, где каждый $t_i \in T$ имеет постоянный период времени (p), и каждый временной ряд выравнивается по времени (начинается с того же экземпляра времени). Набор данных мощности (24) на рис. 2.4 соответствует недельному использованию энергии исследовательской установкой.

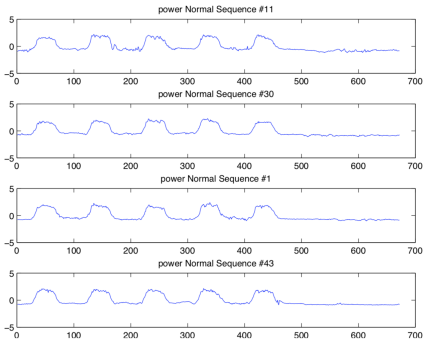


Рис. 3: Периодические и синхронные временные ряды

- *Апериодический и синхронный*: временные ряды не имеют периодичности, но они выровнены по времени. Набор данных клапана (24) на рисунке 2.5 соответствует измерениям тока через клапан на космическом челноке.

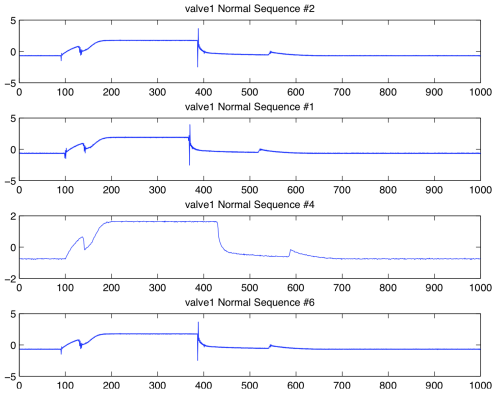


Рис. 4: Апериодические и синхронные временные ряды

- *Периодический и асинхронный*: каждый временной ряд имеет определенный период времени, но не временные. Набор данных двигателя (24) на рисунке 2.6 соответствует функционированию асинхронного двигателя.

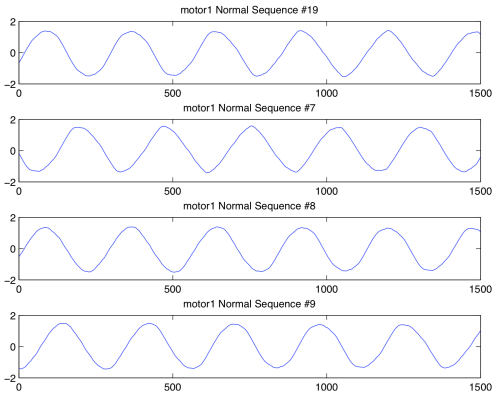


Рис. 5: Периодические и асинхронные временные ряды

- *Апериодические и асинхронные*: временные ряды не имеют периодичности и не выравниваются по времени. Рисунок 2.7 соответствует физиологическим сигналам, полученным из репозитория PhysioNet (25).

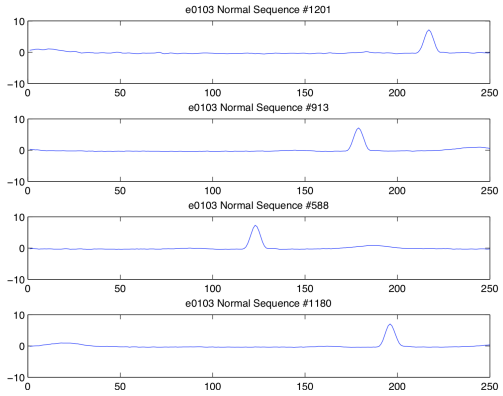


Рис. 6: Апериодические и асинхронные временные ряды

2.5 Подходы к поиску аномалий

Предыдущие разделы дали понимание того, как выглядят временные ряды и как их можно преобразовать для применения методов обнаружения аномалий. В этом разделе обсуждаются различные методы обнаружения аномалий,

некоторые из которых были предложены и оценены в литературе, многие другие являются новыми или небольшими вариациями существующих методов в соответствии с постановкой 3, упомянутой в разделе 2.2. В целом процесс обнаружения аномалий состоит из следующих этапов:

1. Вычислить оценки аномальности отдельных наблюдений или подпоследовательностей заданного временного ряда с использованием метода обнаружения.

2. Полученные оценки применить для вычисления оценок аномальности тестовых временных рядов. Это применение выполнено по-разному, например: (1) среднее значение всех показателей аномалии, (2) среднее значение верхней оценки аномалии, (3) среднее значение логарифма оценок аномалий, (4) оценки аномалий превышают порог и т. д.

Тестовый временной ряд с оценкой аномалии, превышающим порог, помечается, как аномальный.

2.5.1 Алгоритмы на основе скользящих окон

Методы этой категории делят заданные временные ряды на окна фиксированного размера (подпоследовательности), чтобы локализовать причину аномалии в одном или нескольких окнах. Мотивация этого метода заключается в том, что аномалия во временном ряду может быть вызвана наличием одной или нескольких аномальных подпоследовательностей. Оконные методы извлекают фиксированную длину (m) окна (подпоследовательности) из временных рядов для обучения и теста путем перемещения одного или нескольких символов за раз. Оценка аномалии тестового временного ряда рассчитывается путем агрегирования оценок аномалий его окон. Формальное описание общей схемы на основе окон выглядит следующим образом:

1. С учетом данных для обучения, $S_{training} = S_1, S_2, ..., S_n$, извлекают p окон каждого временного ряда $S_i, s_{i1}, s_{i2}, ...s_{ip}$, где p можно вычислить как $|S_i| + m - 1$при сдвиге окна размера m. Аналогично для тестовых данных $S_{test} = T_1, T_2, ..., T_n$, каждый тестовый временной ряд T_i делится на $|T_i| + m - 1$ окно, $t_{i1}, t_{i2}, ...t_{ip}$ '.

2. Оценка аномалий для каждого тестового окна ($A(t_{ij})$) рассчитывается с использованием его сходства с окнами обучения. Эта функция подобия может быть расстоянием, таким как евклидово, манхэттенское или корреляция и т. д.

Методы, основанные на скользящих окнах могут различаться по назначению оценки аномальности в тестовом окне. Например, оценка аномалий тестового окна может быть расстоянием до своего k-го ближайшего соседа между окнами обучения. Ма и др. [27] используют учебные окна для создания классов SVM класса для классификации. Оценка аномалии тестового окна равна 0 или 1 в зависимости от того, классифицируется ли она как нормальная или аномальная, с использованием подготовленного SVM.

После того как окна извлечены из учебных и тестовых временных рядов, можно применить любой традиционный метод обнаружения аномалий для многомерных данных, чтобы присвоить оценку аномалии каждому окну временного ряда теста.

Сильные и слабые стороны: Недостатком методов на основе окон является то, что размер окна должен выбираться тщательно, чтобы он мог явно зафиксировать аномалию. Оптимальный размер окна зависит от длины аномальной области в аномальном временном ряду. Например, в временных рядах, приведенных на рис. 7, аномальная область имеет ту же длину, что и периодичность временных рядов. Таким образом, если m выбрано меньше длины цикла, точность будет низкой, но она улучшится, если m будет больше, чем длина цикла. Еще один недостаток методов, основанных на использовании скользящих окон, заключается в том, что они являются дорогостоящими. Поскольку каждая пара тестовых и обучающих окон сравнивается, сложность $O((nl)^2)$, где l - средняя длина временного ряда, n - количество временных в наборе данных. Большинство методов, основанных на окнах, предлагаются для установки проблемы 2, проблемы обнаружения разлада.

Оконные методы могут захватывать все различные виды аномалий, упомянутых в разделе 2.3: единичные аномалии во временных рядах, аномальная подпоследовательность во временных рядах, аномальный временной ряд в целом. Поскольку весь временной ряд разбит на более мелкие подпоследовательности, мы можем легко определить, является ли наблюдение или подпоследо-

вательность аномальным. Если весь временной ряд является аномальным, то все подпоследовательности также аномальны, поэтому методы, основанные на окнах, хорошо его фиксируют.

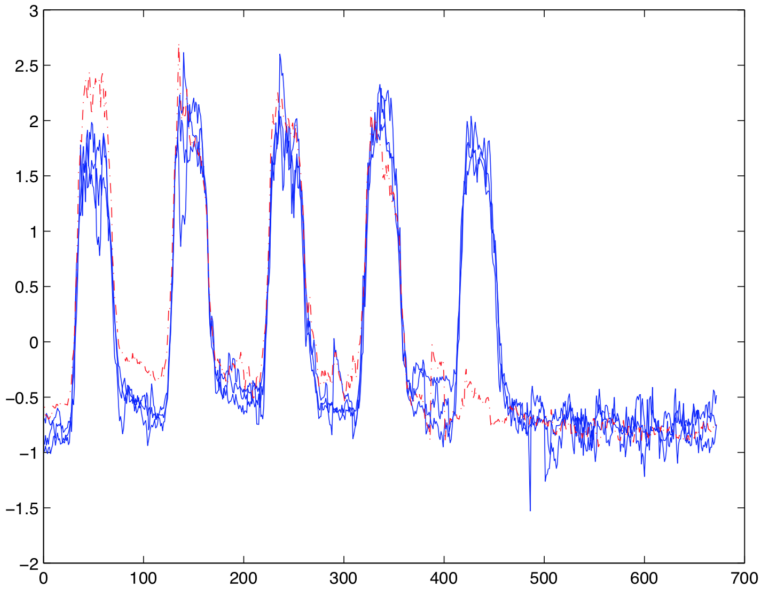


Рис. 7: Аномальный временной ряд (красный), нормальный временной ряд (синий). В аномальном временном ряду отсутствует последний цикл (аномальная область).

2.5.2 Метрические алгоритмы

Эти методы используют попарную близость между рядами для обучения и теста с использованием соответствующего ядра расстояния или подобия для вычисления оценки аномалии временных рядов испытаний. Предположение, лежащее в основе этих методов, заключается в том, что аномальные временные ряды «отличаются» от нормальных, и эта разница может быть зафиксирована с помощью меры близости.

Оценка аномалии тестового временного ряда по отношению рядам из обучения рассчитывается с использованием следующих подходов:

1. k-NN: расстояние от тестового временного ряда до его k-го ближайшего соседа в наборе данных тренировки - это показатель аномалии.

2. Кластеризация. Временные ряды обучения группируются в определенное количество кластеров и вычисляются центроиды кластера. Расстояние между временными рядами испытаний до его ближайшего кластерного центра - его показатель аномалии.

Методы сходства в основном отличаются выбором мер сходства для расчета показателя аномалии. Могут быть приняты различные меры сходства / расстояния, такие как корреляция, евклидово, косинусоидальная близость, DTW и т. Д. Простые меры, такие как евклидовы, хорошо изучены и быстро вычисляются, но не могут использоваться, когда временные ряды имеют разную длину. Кроме того, рассмотрим физиологические сигналы на рисунке 6, которые все нормальны, но у них есть всплески на разных отметках времени. Эти временные ряды нелинейно выровнены, что является одной из проблем при обработке данных временных рядов. Евклидова метрика обеспечила бы высокую оценку аномалии этим временным рядам в случае использования методов на основе расстояний.

Алгоритм динамической трансформации временной шкалы (DTW) - это мера расстояния, которая более подходит для сравнения временных рядов, которые нелинейно выровнены или имеют разную длину [28]. DTW выравнивает временные ряды, которые похожи, но с небольшими изменениями временной оси таким образом, что расстояние между ними минимально. Это минимальное

расстояние можно использовать для характеристики сходства между временными рядами. Недостатком DTW является то, что избыточный соответствий может сделать алгоритм сложным в плане вычислений и привести к искажению фактического расстояния между временными рядами [28].

Еще одна проблема временных рядов заключается в том, что различные ряды могут генерироваться в разных условиях и времени ,а следовательно, они могут быть несинхронными. Следовательно, такие меры, как евклидово и DTW, могут быть фазово несогласованными и, как следствие, неодинаковыми, хотя они почти схожи. Например, рассмотрим данные работы двигателя, показанный на рисунке 5, где все временные ряды являются нормальными, но они не находятся в фазе (с фазовым смещением). В то время, как DTW решает проблему нелинейного выравнивания временных рядов, кросс-корреляция, мера сходства, решают проблему фазовых искажений [28]. При двух временных рядах учитываются разные фазовые сдвиги пары и вычисляется корреляция между ними для каждого выравнивания. Максимум этих корреляций используется в качестве меры взаимной корреляции для этих двух временных рядов. Таким образом, два временных ряда с фазовыми смещениями будут иметь высокую корреляцию в одном из своих смещенных по фазе выравниваний, что показывает, что они имеют высокое сходство. Повторяя набор данных двигателя на рисунке 5, первые два временных ряда будут иметь максимальную корреляцию, когда один из них рассматривается с фазовым сдвигом $\frac{\pi}{2}$.

Методы обнаружения аномалий в этой категории включают метод обнаружения аномалий, основанный на k-ближайшем соседстве, который был предложен Propotopapas и др. [28] Они решают проблемы с фазосбалансированными данными с использованием меры взаимной корреляции, которая эффективно вычисляется в пространстве Фурье с использованием теоремы свертки.

Для той же проблемы сравнения периодических световых кривых, Rebba Pragada и др. (16) используют алгоритм кластеризации PCAD (периодическая кривая аномалии), который является вариантом k-средних. Близость временных рядов является функцией максимальной меры взаимной корреляции между временным рядом и центроидами кластеров, полученными из PCAD.

Сильные и слабые стороны: Недостатки метрических методов заключаются в том, что они могут определить, является ли весь временной ряд ано-

мальным или нет, но не могут точно определить аномальную подпоследовательность. Чтобы локализовать точную область (области) в пределах временного ряда, который вызывает аномалию, необходимо выполнить пост-обработку временного ряда. Также несогласованность фаз и нелинейные выравнивания различных временных рядов, которые являются некоторыми общими проблемами для данных временных рядов, ограничивают использование различных мер приближения для этих классов методов. Таким образом, эффективность этих методов сильно зависит от меры близости, которую часто бывает нелегко выбрать. При обнаружении различных типов аномалий методы, основанные на подобии, могут потерпеть неудачу, когда одно наблюдение является аномальным в временном ряде, поскольку его эффект может быть не заметным когда сразу рассматриваются все временные ряды.

Эти методы обнаруживали бы аномалии, такие как аномальные подпоследовательности во временном ряде или аномальные временные ряды в целом.

2.5.3 Алгоритмы на основе прогнозирования

Обнаружение аномалий с использованием предсказательных моделей в основном изучается в статистике, большинство подходов ищут отдельные наблюдения, как выбросы. Идея этих методов заключается в том, что нормальные временные ряды генерируются из статистического процесса, а аномальные временные ряды не соответствуют этому процессу. Таким образом, ключевой составляющей является изучение параметров этого процесса из набора нормальных временных рядов для обучения, а затем оценка вероятности того, что тестовый временной ряд обладает такими же характеристиками, что и ряды для обучения.

Алгоритмы на основе предсказаний состоит из следующих этапов:

1. Модель обучается на m идущих подряд временных отсчетов, чтобы предсказывать $m + 1$-й, следующий за ним.

2. Обученная модель из первого этапа применяется на данных для теста: для каждого временного отсчета начиная с $m+1$-ого прогнозируется его значение на основе предыдущих m наблюдений. Ошибка прогноза, соответствующая наблюдению, зависит от разницы между прогнозируемым значением,

фактическим наблюдением и некоторыми параметрами модели таких как дисперсия модели.

Методы отличаются используемыми моделями прогнозирования и могут быть классифицированы следующим образом:

1. Модели на основе временных рядов, такие как скользящее среднее (MA), авторегрессионная модель (AR), ARMA, ARIMA, фильтры Калмана и т. д. Входными данными этих моделей являются все временные ряды и длина истории (m), также обозначаемая как порядок модели. Эти модели отличаются типом фильтров, которые они используют для генерации прогнозов.

   (a) Скользящее среднее (MA): MA-модели представляют временные ряды, которые генерируются путем пропускания входного сигнала через линейный фильтр, который генерирует выход$y(t)$ в любое время t, используя только входные значения $x(t-\tau), 0 \leq \tau \leq m$, также называемый нерекурсивным фильтром.

   $$y(t) = \sum_{i=1}^{m} b(i)x(t-i) + \epsilon(t)$$

   где $b(1), b(2), ..., b(m)$ - коэффициенты нерекурсивного фильтра, $\epsilon(t)$ - шум в момент времени t. Если каждый экземпляр t временного ряда имеет значение, равное среднему значению его предыдущих значений m, то он может быть представлен моделью скользящего среднего, в этом случае $b(i) = \frac{1}{m}$ и $\epsilon(t) = 0$.

   (b) Авторегрессия (AR): AR-модели представляют временные ряды, которые генерируются путем пропускания входного сигнала через линейный фильтр, который производит выход y(t) в любое время t, используя предыдущие выходные значения $y(t-\tau), 1 \leq \tau \leq m$, также называемый рекурсивным фильтром.

   $$y(t) = \sum_{i=1}^{m} a(i)y(t-i) + \epsilon(t)$$

   где $a(1), a(2), ..., a(m)$ - коэффициенты авторегрессии рекурсивного фильтра (коэффициенты авторегрессии), $\epsilon(t)$ - шум в момент времени t.

(c) Модель авторегрессии — скользящего среднего (ARMA): модели ARMA представляют временные ряды, которые генерируются путем последовательного прохождения входного сигнала через рекурсивный и нерекурсивный линейный фильтр. Другими словами, модель ARMA представляет собой комбинацию модели авторегрессии (AR) и модели скользящего среднего (MA). Порядки AR-части модели и MA-части модели могут различаться.

$$y(t) = \sum_{i=1}^{n} a(i)y(t-i) + \sum_{i=1}^{n} b(i)x(t-i) + \epsilon(t)$$

где $a(1), a(2), ..., a(n)$ - коэффициенты авторегрессии рекурсивного фильтра (коэффициенты авторегрессии), $b(1), b(2), ..., b(m)$ - коэффициенты нерекурсивного фильтра, $\epsilon(t)$ - шум в момент времени t.

(d) Интегрированная модель авторегрессии — скользящего среднего (ARIMA): модели ARIMA, которые являются расширением модели ARMA, применяют модель ARMA не сразу к заданным временным рядам, а после ее предварительного дифференцирования, которое представляет собой временной ряд, полученный путем вычисления разницы между последовательными значениями исходного временного ряда.

(e) Фильтры Калмана: Основная идея фильтра Калмана описывается следующим образом: рассмотрим временной ряд с марковским свойством, описываемый следующим уравнением:

$$x(t+1) = Ax(t) + \epsilon(t)$$

где $x(t)$ представляет собой скрытое состояние системы, а A - матрица, описывающая причинную связь между текущим состоянием $x(t)$ и следующим состоянием $x(t+1)$. Фильтр Калмана для временных рядов $X = (x(1)x(2)...x(n))$ описывается следующим образом:

$$y(t+1) = Ax(t) + K(x(t) - y(t))$$

где K называется поправкой Калмана.

2. Регрессия общего вида (без учета временных рядов): линейная регрессия, регрессия Гауссовского процесса, метод опорных векторов и т. д. Подпосле-

довательности длины m (длина истории), извлеченные из исходного временного ряда, являются входными данныи для таких моделей. Тренировочные данные представляют собой набор подпоследовательностей, определяемый формулой $T = (X(t), y(t)), t = m, ...n-1$, где $X(t) = [x(t-m+1)...x(t)]$ и $y(t) = x(t+1)$. Функция линейной регрессии строится с использованием весового вектора W и функции отображения $\Phi(X(t)), y = W^T?(X(t)) + b$. Различные модели регрессии отличаются тем, как они соответствуют этой функции.

(a) Линейная регрессия: для простой линейной регрессии приведенное выше уравнение решается путем минимизации суммы квадратичной ошибки вычета $(y(i) - x(i+1))$. Функция отображения здесь тождественна, $\Phi(X(t)) = X(t)$.

(b) Поддержка векторной регрессии: эти модели используют функцию чувствительности, предложенную Вапником. Они решают вышеупомянутое уравнение, решая следующую целевую функцию.

$$minP = \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{l} \zeta_i - \zeta_i^*$$

$$\text{такой, что } y_i - (W^T\Phi(X(t)) + b) \leq \epsilon + \zeta_i$$

$$-y_i + (W^T\Phi(X(t)) + b) \leq \epsilon + \zeta_i^*$$

$$\zeta_i, \zeta_i^* \leq 0$$

Этот критерий оптимизации наказывает точки данных, значения у которых отличаются от $f(x)$ более чем на ϵ. Слабые переменные ζ_i, ζ_i^* представляют собой величину избыточного отклонения. Могут использоваться различные функции ядра, такие как полином, RBF и сигмоид.

Ма и др. [27] используют для обучения подпоследовательности длины m и применяют для них поддержку векторной регрессии для создания онлайн-модели обнаружения новизны, которая также использует статистические тесты для определения аномалий с некоторой фиксированной уверенностью.Чандола и другие [7] используют AR-модель в исходных временных рядах, в то время как Лотз и другие используют вейвлет-преобразования

временного ряда обучения, а затем использовать нейронные сети для прогнозирования. В [23] предлагается простую модель предсказания, основанную на наименьшем разрешении вейвлета (называемом трендом) подпоследовательностей. Пусть исходная подпоследовательность $X_i = x(1)x(2)...x(i-1)$, а ее тренд $Yi = y(1)y(2)...y(i-1)$. Построено распределение разности X_i и Y_i, называемое остаточным. Если разность $x(i)$ и тренд $y(i-1)$ статистически незначительна в соответствии с остаточным распределением, то наблюдение $x(i)$ называется аномальным.

Сильные и слабые стороны: Подобно методам, основанным на скользящем окне, длина, выбранной здесь истории имеет важное значение, при определении местонахождения аномалии. Обращаясь к рис. 7, аномальная область имеет ту же длину, что и периодичность временных рядов. Таким образом, если длина истории m выбрана меньше длины цикла, производительность будет неудовлетворительной, а производительность улучшится, если m будет больше, чем длина цикла. Но если значение m слишком велико, то мы имеем очень большие размерные данные, что может увеличить вычислительную сложность. Кроме того, из-за редкой природы высокоразмерных данных вступает в действие проклятие размерности, т. е. наблюдения, которые ближе в меньшее размерных пространствах, будут казаться очень далеко в пространстве большей размерности из-за его разреженности.

Поскольку эти методы предполагают, что данные генерируются из статистического процесса, если это предположение верно (пример: двигатель (рисунок 5), мощность (рис. 3)), эти методы работают хорошо, хотя задача состоит в том, чтобы найти правильный процесс и оценить его параметры, иначе, если данные не генерируются процессом (Ex: физиологические сигналы (рис. 7)), они могут не зафиксировать аномалии.

Все методы, основанные на прогнозировании, используют истории фиксированной длины. Иногда для прогнозирования достаточно маленькой истории, но в других случаях может потребоваться более длинная история. Таким образом, можно использовать историю динамической длины, где если наблюдение не может быть предсказано с высокой достоверностью, учитывая историю длины m, то увеличивайте или уменьшайте длину истории, чтобы предсказать наблюдение с большей достоверностью.

Методы, основанные на прогнозах, вычисляют показатель аномалии для каждого из наблюдений в временных рядах. Следовательно, они могут захватывать все виды аномалий: аномальное наблюдение , аномальную подпоследовательность, аномальный временной ряд в целом.

2.5.4 Алгоритмы на основе скрытых Марковских моделей

Скрытая марковская модель (СММ) — статистическая модель, имитирующая работу процесса, похожего на марковский процесс с неизвестными параметрами, и задачей ставится разгадывание неизвестных параметров на основе наблюдаемых. СММ широко используются для моделирования последовательности [29] и обнаружения аномалии в последовательности [30]. Они также были применены к обнаружению аномалий во временных рядах [31]. Предположение о методах, основанных на СММ, заключается в том, что наблюдаемый временной ряд $O = O_1, ..., O_n$ - косвенное наблюдение лежащего в основе (скрытого) временного ряда $Q = Q_1, ..., Q_n$, где процесс, создающий скрытые временные ряды, является марковским, хотя наблюдаемый процесс, создающий исходный временной ряд, может быть не таким. Таким образом, нормальный временной ряд может быть смоделирован с использованием СММ, а аномального временного ряда быть не может. С набора учетом данных для обучения, мы можем построить единую модель СММ (λ), которая состоит из параметров, описывающих нормальные данные, таких как распределение начального состояния, вероятности перехода состояния и т. д. [33]. Каждая временной ряд для обучения может иметь конкретную модель или может быть единая модель для всех временных рядов обучения. Основной метод обнаружения аномалий на основе СММ работает следующим образом:

1. Данные для обучения, $O_{train} = O_1, ..., O_n$ рассматриваются как последовательность косвенных наблюдений модели СММ. Существует хорошо установленная процедура, которая позволяет определять параметры СММ, максимизируя вероятность $P(O_{train}|\lambda)$, используя метод, называемый процедурой переоценки Баума-Уэлша [34].

2. На этапе тестирования, учитывая неизвестный временной ряд $O_{test} = O1, ..., O_n$, $P(O_{test}|\lambda)$ вероятность вычисляется с использованием обученной модели.

Среди временных рядов испытаний можно сказать, что аномальными являются те, которые имеют минимальное значение $P(O_{test}|\lambda)$.

Существующие методы немного отличаются по используемым параметрам CMM. Не и др. [32] используют стандартные параметры, такие как распределение начального состояния, вероятности перехода состояния и т.д. Лю и др. используют сегментный CMM, где дополнительный признак - это вероятностное распределение по продолжительности каждого скрытого состояния.

Сильные и слабые стороны: Идея методов, основанных на CMM, заключается в том, что существует скрытый процесс, который является марковским и генерирует нормальные временные ряды. При отсутствии основного марковского процесса эти методы могут не учитывать аномалии.

Методы на основе CMM создают марковскую модель для временных рядов. Следовательно, они оценивают аномальное поведение для каждого из наблюдений в временном ряду, что помогает определить все типы аномалий (пока действительно предположение).

3 Описание работы алгоритмов классификации временных рядов

3.1 Метрические методы

Алгоритмы классификации, такие как k ближайший соседей (kNN), зависят от расстояний между данными. В обычных алгоритмах классификации для работы с последовательными данными необходимо найти новые метрики для определения расстояния между двумя последовательностями. В работе [14] утверждается, что выбор критериев расстояния (сходства) играет значительную роль в качестве алгоритма классификации.

Хотя евклидово расстояние является общепринятой метрикой, требуется, чтобы временные ряды были одинаковой длины [14]. В дополнение к этому ограничению в [15] подчеркивается чувствительность к искажению во времени. В работе [16] говорится, что искажение по времени является общим для задач распознавания речи, где скорость речи не является постоянной. Искажение также нелинейно, поэтому линейного преобразования будет недостаточно. Таким образом, для решения этой проблемы необходимы алгоритмы позволяющие найти оптимальное соответствие между временными последовательностями, такие как алгоритм динамической трансформации временной шкалы (DTW).

DTW - это нелинейное отображение между двумя последовательностями, где расстояние между ними минимизировано. В нем строится матрица $n \times m$, и каждый элемент в ней представляет собой попарное расстояние между точками в двух последовательностях. Затем ищется путь в матрице, где общая сумма расстояний минимальна, который затем возвращаются как расстояние между двумя строками. Хотя DTW решает многие проблемы евклидова расстояния, его вычислительная неэффективность ограничивает его принятие. DTW рассчитывается с использованием динамического программирования, следовательно, имеет квадратичную временную сложность ($O(n * m)$ или $O(n^2)$).

3.2 Метод основанный на признаках

Классические алгоритмы классификации, такие как ANN и деревья принятия решений, выполняют свою классификацию на основе набора признаков, поэтому методы классификации основанные на признаках преобразуют временные ряды в набор признаков перед передачей его алгоритмам классификации [14]. Выбор соответствующих признаков - самая сложная часть этого процесса, всегда существует компромисс между выполнением этого процесса вручную или его автоматически, но менее точно. Существует множество подходов к извлечению признаков. Некоторые из них перечисленны ниже.

Самым простым методом извлечения признаков является статистический метод. Так как временной ряд - это просто упорядоченная последовательность чисел, то его можно характеризовать такими статистическими параметрами, как среднее, стандартное отклонение, квантили различного порядка и др. Недостатком такого подхода является отсутвие возможности извлечь локальные признаки, так как метод извлекает статистики из всего ряда.

В работе [35] введен алгоритм извлечения признаков под названием Minimal Distinguishing Subsequence (MDS). Однако MDS допускает зазоры в подпоследовательности, что делает его более подходящим для классификации биологических последовательностей, как упоминалось ранее.

Другой метод извлечение признаков - преобразование данных временного ряда в частотную область, где размерность данных может быть уменьшена. Например, DFT (дискретное преобразование Фурье), DWT (дискретное преобразование вейвлета) и SVD (сингулярное декомпозиция). Однако отмечается, что DWT более распространен в классификации, поскольку он сохраняет как временные, так и частотные характеристики, тогда как DFT предоставляет только частотные характеристики. Такая трансформация также решает проблему, рассмотренную ранее, в 2.1, где нам необходимо изучить как локальные, так и широкие аномалии в последовательных данных. DWT преобразует данные в разные частотные компоненты. Компоненты с коэффициентами более высокого порядка отражают глобальные тенденции данных, а коэффициенты с более низким коэффициентом отражают локальные тенденции в ней.

Ядерные методы (KM) также хороши при извлечении признаков, кроме

того, они могут иметь дело с символьными последовательностями разной длины [34]. Хотя рассматривались текстовые данные. такие как мешок слов, а не последовательные данные, подчеркивалась способность методов ядра обрабатывать текстовые данные независимо от их огромного количества признаков (более 10 тысяч). В частности, использовался Support Vector Vector Machine, который является одним из ядерных методов. Ядерные методы вычисляют скалярное произведение входных векторов в высокоразмерном пространстве [21].

3.3 Классификация основанная на модели

Согласно [21], модельные методы строят модель каждого класса и сопоставляет новым данным метку класса, которая наилучшим образом подходит для нее. Он разделил модели, используемые в классификации, на статистические и нейронные сети. Статистические модели, такие как: гауссовские, пуассоновские, марковские и скрытые марковские модели, построены так, что они моделируют распределение вероятностей данных [29]. В [23], с другой стороны, разделяли модели на предсказательные модели, которые пытаются предсказать недоступные значения данных с использованием существующей и описательные модели, которые пытаются найти шаблоны и отношения в данных.

Скрытая Марковская модель (HMM) определена, как "набор состояний S, алфавит из m символов, матрица вероятностного перехода $T = (t_{ij})$ и матрица вероятностных излучений $E = (e_{ia})$. Когда система находится в состоянии i, она имеет вероятность t_{ij} перехода к состоянию j и вероятности e_{ia} исходного символа a" [24]. В [23] объяснено использование HMM в классификации следующим образом: для каждого класса HMM строится с использованием данных обучения из этого класса, затем новые экземпляры сравниваются со встроенными моделями, чтобы определить, какая модель (класс) соответствует новым данным лучше. HMM более успешна в классификации биологических последовательностей по сравнению с Neural Networks, поскольку она может иметь дело с последовательностями переменной длины, в то время как другой метод требует ввода фиксированной длины [?]. В [29] с другой стороны, определил некоторые общие ограничения HMM, такие как предположение, что вероятность того, что они находятся в определенных состояниях, опирается только в

предыдущем состоянии, а также предположение о том, что вероятности наблюдений независимый.

Как правило, искусственные нейронные сети (ANN) очень близки к статистическим моделям. Рекуррентные нейронные сети являются (RNN) как особым типом ANN, где есть связь обратной связи в сети, чтобы отслеживать ее внутреннее состояние при работе с новыми входами [25]. RNN подходит для последовательных данных, поскольку RNN способен моделировать временную природу последовательности [25]. Кроме того, в отличие от HMM, RNN не требует знания особенностей данных [26]. RNN невосприимчивы к временному шуму. Тем не менее, как было замечено ранее, они требуют ввода фиксированной длины.

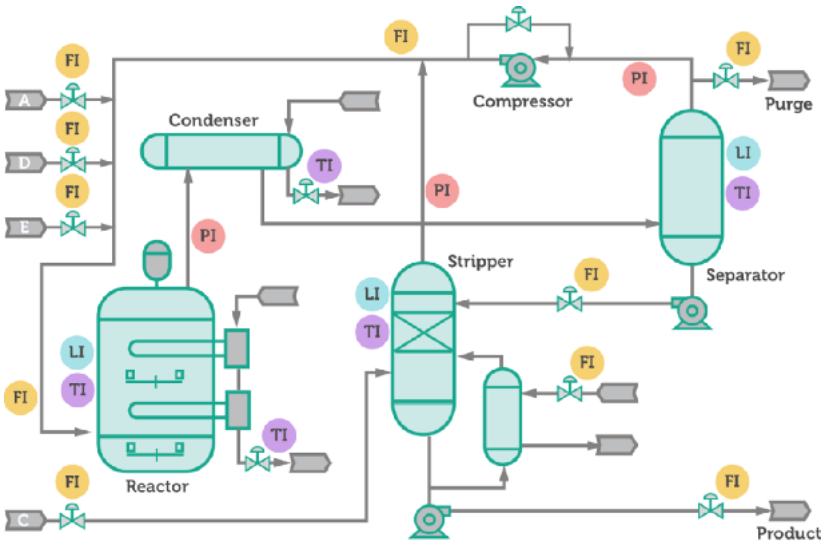
4 Описание эксперимента

4.1 Описание данных

Существует два основных подхода к сбору данных для обучения моделей. Первый вариант - это сбор данных с реального устройства. Для этого производится запись показаний устройств в различных режимах его нормальной работы и во время сбоев или атак. Основной проблемой данного подхода является время и стоимость сбора данных, так как устройство работает в реальном времени и не может быть ускоренно, а так же может быть повреждено в ходе атак. Второй вариант - это симуляция работы устройства и атётак на него. Данный вариант является менее приближен к реальности, но решает проблемы предыдущего.

В данном проекте использовался второй тип данных сгенерированной для модели Tennessee Eastman Process (TEP), описанная в [10]. Кибер-атаки для модели TEP были предложенны в [11], а реализация была позаимствована у [13].

На рис. 1 изображена модель TEP.



Описание данных приведено в таблице ниже:

59	**Размерность временных рядов**
41	Датчика (MEAS)
12	Управляемых переменных (MV)
1	Индикатор атаки на MEAS

1	Индикатор атаки на MV
1	Индикатор атаки SP (set point)
3	Особые переменные (состояние, скорость выпуска, стоимость выпуска за час)
7	Нормальных состояний
28	Переходных состояний
	Типы аттак
	DoS (значение переменной не меняется)
	Integrity (значение переменной меняется))
	Noise (значение датчиков + шум)
	Временные ряды с атаками (№, Тип, MV/MEAS/SP, продолжительнось)
	№21: Integrity: MEAS "температура реактора", 0.012-0.027 ч
	№22: DoS: MV "Сливной поток", MEAS "Уровень слива", MEAS "Нижний уровень слива", 5.663-25.019 ч
	№23: DoS: MV "D скорость потока", 10 ч
	№24: Noise: MV "C скорость потока", MV "Поток зачистки", MEAS "Нижний уровень слива потока зачистки", MV "Выпуск пара", 7.727 - 71.291 ч
1000	**отсчетов за час**
	Данные для обучения (длительность)
201	Ряд в нормальном состоянии
336	Рядов с переходными процессами
	Данные для тестирования (длительность)
142	образцов с атаками ($\leq$ 120 ч до поломки)

В наборе данных имеется 201 ряд с нормальным состоянием, 336 с переходными и 142 с MEAS/MV/SP атаками. Каждый образец является многомерным временным рядом порядка 59. Атаки настроены так, что работа завода возвращается к нормальному режиму сразу после атаки. Было проведено 4 типа атак, которые описаны в таблице выше.

4.2 Поиск аномалий

Входные данные могут быть описаны как многомерные временные ряды $X = x(1), x(2), ..., x(n)$, где $x(t)$ принадлежит $m-$мерному пространству R^m, $n-$число точек времени. Предлагаемый алгоритм обнаружения неисправностей состоит из двух частей: прогнозирования и обнаружения. Сначала мы разбиваем все временные ряды на равные последовательности длины w, обозначенные как $X(i) = x(j), x(j+1), ..., x(j+w-1)$. Здесь i - номер последовательности, $_j = w(i-1)+1-$номер первой точки времени в последовательности. В части прогнозирования мы прогнозируем значения для следующей последовательности $X(i+1)$, используя уже наблюдаемые измерения $X(1), X(2), ..., X(i)$. Часть обнаружения основана на поиске временных точек, где среднеквадратическая ошибка (MSE) между измеренными значениями $X(i+1)$ и предсказанными значениями $X(i+1)$ становится выше, чем предварительно вычисленный порог.

В рамках данной работы использовались 2 алгоритма: ARIMA и LSTM. Описание работы которых приведено ниже.

4.2.1 Предобработка данных

Все точки данных в представленном наборе данных имеют одну и ту же временную сетку и имеют существенно различающиеся абсолютные значения. Чтобы уменьшить эти вариации и объединить разные размеры, мы применили преобразование нормализации по каждому измерению отдельно:

$$x_i^{*(j)} = \frac{x_i^{(j)} - \bar{x_j}}{\sigma_i}, i = \overline{1, m}.$$

$\bar{x_i}$ и σ_i- это среднее и стандартное отклонение для каждого измерения.

4.2.2 Поиск аномалий на основе ARIMA

Модели ARIMA строятся на основании предыстории исследуемых временных рядов. Данная модель впервые была предложена Дж. Боксом и Г. Дженкинсом, и поэтому в некоторых источниках модель авторегрессии проинтегрированного скользящего среднего также называют «моделью Бокса-Дженкинса» или в англоязычной литературе Auto Regressive Integrated Moving Average model

(ARIMA-model). Модель ARIMA представляет собой обобщение модели авторегрессионного скользящего среднего и предназначена для описания нестационарных временных рядов $y_t, t = \overline{1,T}$. В общем случае модель обозначается следующим образом: $ARIMA(p,d,q)$, где $p-$порядок авторегрессии, $d-$порядок интегрирования, $q-$порядок скользящего среднего.

Построение модели AR и модели MA предполагает стационарность временных рядов. Стационарность во временных рядах означает то, что взаимное распределение вероятностей m наблюдений ряда y_t не зависит от времени t. В качестве приема преодоления проблемы нестационарности временного ряда можно использовать переход к ряду приращений:

$$\Delta^1 y_t = y_t - y_{t-1}$$

$$\Delta^2 y_t = \Delta y_t - \Delta y_{t-1}$$

$$...$$

$$\Delta^d y_t = \Delta^{d-1} y_t - \Delta^y_{t-1} \backsim ARMA(p,q) \Rightarrow y_t \backsim ARIMA(p,d,q)$$

Если ряд приращений порядка d является стационарным $\Delta^d y_t$, то исходный временной ряд y_t называется интегрируемым порядка d.

Количество разностей, которые берутся для достижения стационарности, определяется порядком разности d. Необходимый порядок разности определяется путем исследования графика ряда. Сильные изменения уровня (сильные скачки вверх или вниз) обычно требуют взятия несезонной разности первого порядка. Сильные изменения наклона требуют взятия разности второго порядка.

Не смотря на широкое применение алгоритма ARIMA в эконометрике и статистике, в данной задаче он имеет ряд недостатков:

1. Так как поиск аномалий осуществляется в многомерных рядах, то многие из рядов могут быть скоррелированны между собой и хранить информацию о совместной работе разных компонент системы. Модель ARIMA никак не учитывает этот фактор.

2. Исследуемая система может работать в различных режимах, статистические характеристики которых могут существенно отличаться. Таким образом обучая модель на разнородных типах данных будет возрастать ошибка

предсказаний. В силу большой ошибки работы модели качество обнаружения аномалии может существенно упасть.

Все выше указанные проблемы решены следующей моделью.

4.2.3 Поиск аномалий на основе глубоких нейронных сетей

Выбор оптимальной архитектуры нейронной сети основан на нескольких факторах. Во-первых, датчики процессов компьютерных систем генерируют сильно коррелированные многомерные временные ряды. Кроме того, мы часто имеем дело с многомасштабными процессами (см. Рис. 2), имеющими быстрые (долгосрочные) и медленные (краткосрочные) подпроцессы. В этих условиях обычные нейронные сети прямого распространения обычно демонстрируют плохие результаты. Точная управляемая данными прогнозирующая модель может быть разработана с использованием нейронной сети с ячейками LSTM [18]. Предлагаемая сетевая архитектура включает в себя два слоя LSTM с линейным выходным уровнем (рисунок 4). Кроме того, для модели прогнозирования используется последовательную архитектуру сети LSTM (рисунок 4).

Обнаружение основано на MSE между фактическими данными и прогнозируемыми значениями.

$$MSE(X^{*(i)}, \tilde{X}^{(i)}) = \frac{1}{m}\sum_{i=2}^{m}(x^{*(i)} - \tilde{x}^{(i)})^2$$

Чтобы сгладить большие ошибки в одиночных точках, мы применили экспоненциальное скользящее среднее MSE, где экспоненциальный параметр «период полураспада» был выбран как удвоенная длина ряда.

Долгая краткосрочная память (Long short-term memory; LSTM) – особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям. LSTM разработаны специально, чтобы избежать проблемы долговременной зависимости (запоминание информации на долгие периоды времени). Структура LSTM также напоминает цепочку, но модули выглядят иначе. Вместо одного слоя нейронной сети они содержат четыре (см. Рис).

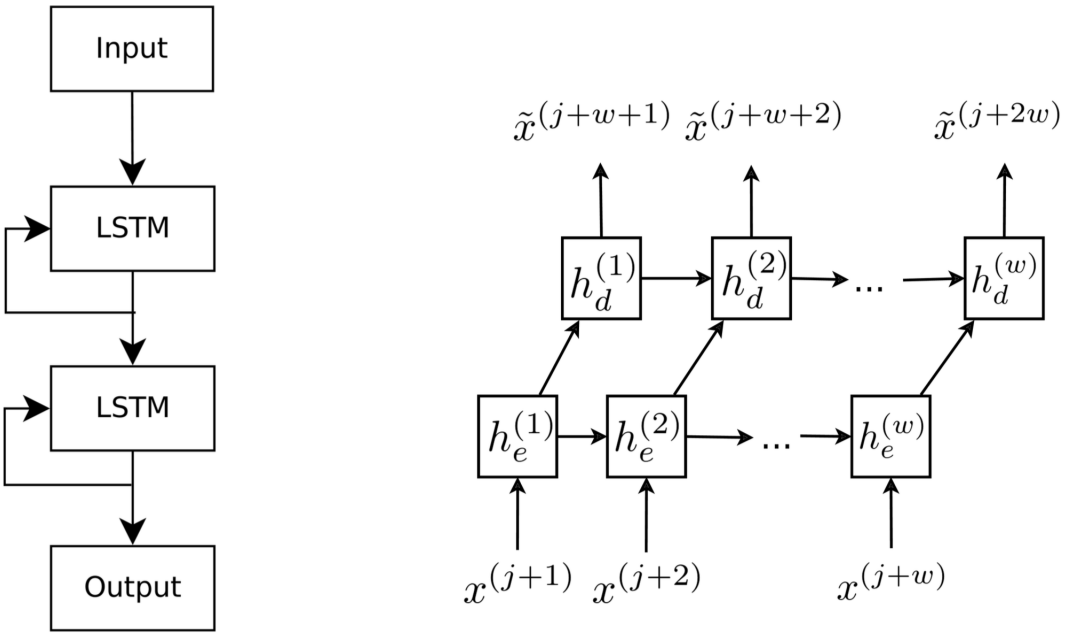


Рис. 8: Архитектура нейронной сети и
схема прогнозирования

4.2.3.1 Принцип работы ячейки LSTM

Рассмотрим структуру ячейки LSTM. Ключевой компонент LSTM – это состояние ячейки (cell state) – горизонтальная линия, проходящая по верхней части схемы. LSTM может удалять информацию из состояния ячейки; этот процесс регулируется структурами, называемыми фильтрами (gates). Фильтры позволяют пропускать информацию на основании некоторых условий. Они состоят из слоя сигмоидальной нейронной сети и операции поточечного умножения. Сигмоидальный слой возвращает числа от нуля до единицы, которые обозначают, какую долю каждого блока информации следует пропустить дальше по сети. Ноль в данном случае означает “не пропускать ничего”, единица – “пропустить все”.

Рассмотрим поэтапно алгоритм работы ячейки LSTM:

1. Определяется информация, которую можно удалить из состояния ячейки. Это решение принимает сигмоидальный слой, называемый “слоем фильтра забывания” (forget gate layer). Он учитывает h_{t-1} и x_t и возвращает число от 0 до 1 для каждого числа из состояния ячейки C_{t-1}. 1 означает “полностью сохранить”, а 0 – “полностью выбросить”. Рассчет производится по

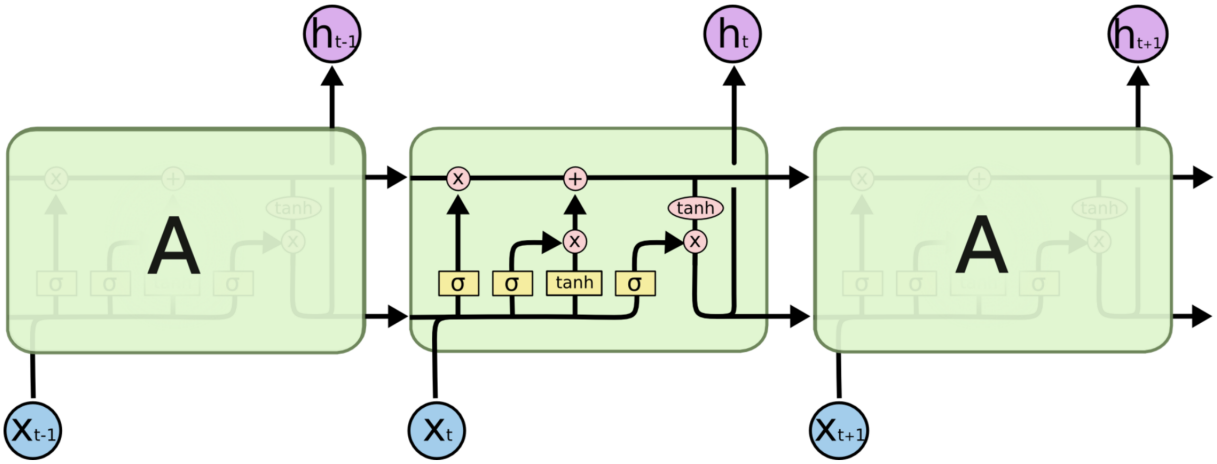


Рис. 9: Схема яцейки LSTM

Рис. 10: Обозначение

следующей формуле:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

2. Принимается решение о том, какая новая информация будет храниться в состоянии ячейки. Этот этап состоит из двух частей.

   (a) Сначала сигмоидальный слой под названием “слой входного фильтра” (input layer gate) определяет, какие значения следует обновить.

   $$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

   (b) Затем tanh-слой строит вектор новых значений-кандидатов $\tilde{C}_t$, которые можно добавить в состояние ячейки.

   $$\tilde{C}_t = \sigma(W_C \cdot [h_{t-1}, x_t] + \tilde{b_C})$$

3. Обновление старого значения состояния C_{t-1} ячейки на новое C_t.

   $$C_t = f_t * C_{t_1} + i_t * \tilde{C}_t$$

4. Генерация выходных данных:

   (a) С помощью сигмоидального слоя определяется, какая информацию из состояния ячейки будет выводиться.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

   (b) Значения состояния ячейки проходят через tanh-слой, чтобы получить на выходе значения из диапазона от -1 до 1, и перемножаются с выходными значениями сигмоидального слоя, что позволяет выводить только требуемую информацию.

$$h_t = o_t * tanh(C_t)$$

4.3 Классификация аномалии

Обозначим размеченный набор временных рядов как $D = (T_i, y_i)N_i = 1$, который содержит N временных рядов и их ассоциированные метки. Для каждого $i = 1, ..., N, T_i$ представляет собой i-й временной ряд и его метку - y_i. y_i является категориальным значением в $C = 1, ..., C$, где $C \in Z+$ - количество меток. Задачей классификации временных рядов является построение предсказательной модели для прогнозирования метки класса $y \in C$ с учетом входного временного ряда T. В отличие от некоторых предыдущих работ, мы не требуем, чтобы все временные ряды обучения и тестирования имели одинаковое количество временных меток в наших рамках.

4.3.1 Классический подход классификации

В рамаках классического подхода классификации временных рядов из них были выделены статистические признаки такие, как: среднее, стандартное отклонение, минимум, максимум, квантили различного порядка и так далее. Такие признаки считались в по отдельности для каждого из рядов многомерного временного ряда, а затем конкатинировались. Таким образом каждый многомерный временной ряд переводился в пространство параметров размерности $8 \times 53 = 432$. В рамках данного пространства можно применить классические алгоритмы классификации.

4.3.1.1 Наивный Байесовский классификатор

Байесовский подход к классификации основан на теореме, утверждающей, что если плотности распределения каждого из классов известны, то искомый алгоритм можно выписать в явном аналитическом виде. Более того, этот алгоритм оптимален, то есть обладает минимальной вероятностью ошибок [1].

Пусть для каждого класса $y \in Y$ известна априорная вероятность P_y того, что появится объект класса y, и плотности распределения $p_y(x)$ каждого из классов, называемые также функциями правдоподобия классов. Требуется построить алгоритм классификации $a(x)$, доставляющий минимальное значение функционалу среднего риска.

Средний риск опредеяется как математическое ожидание ошибки:

$$R(a) = \sum_{y \in Y} \sum_{s \in Y} \lambda_y P_y \mathsf{P}_{(x,y)} \big\{ a(x) = s | y \big\},$$

где λ_y — цена ошибки или штраф за отнесение объекта класса y к какому-либо другому классу.

Решением этой задачи является алгоритм

$$a(x) = \arg\max_{y \in Y} \lambda_y P_y p_y(x)$$

Значение $P\{y|x\} = P_y p_y(x)$ интерпретируется как апостериорная вероятность того, что объект x принадлежит классу y.

Если классы равнозначимы, $\lambda_y P_y = \text{const}(y)$, то объект x просто относится к классу с наибольшим значением плотности распределения в точке x.

4.3.1.2 Классификация с помощью Логистической Регрессии

Положим $Y = \{-1, +1\}$. В логистической регрессии строится линейный алгоритм классификации $a: X \to Y$ вида

$$a(x, w) = \text{sign}\left(\sum_{j=1}^{n} w_j f_j(x) - w_0 \right) = \text{sign}\langle x, w \rangle$$

, где w_j - весj-го признака, w_0 - порог принятия решения, $w = (w_0, w_1, \ldots, w_n)$ - вектор весов, $\langle x, w \rangle$ - скалярное произведение признакового описания объек-

та на вектор весов. Предполагается, что искусственно введён «константный» нулевой признак: $f_0(x) = -1$.

Задача обучения линейного классификатора заключается в том, чтобы по выборке X^m настроить вектор весов w. В логистической регрессии для этого решается задача минимизации эмпирического риска с функцией потерь специального вида:

$$Q(w) = \sum_{i=1}^{m} \ln\left(1 + \exp(-y_i\langle x_i, w\rangle)\right) \to \min_w$$

После того, как решение w найдено, становится возможным не только вычислять классификацию $a(x) = \text{sign}\langle x, w\rangle$ для произвольного объекта x, но и оценивать апостериорные вероятности его принадлежности классам:

$$\mathbb{P}\{y|x\} = \sigma\left(y\langle x, w\rangle\right), \quad y \in Y$$

где $\sigma(z) = \frac{1}{1+e^{-z}}$ - сигмоидная функция. Во многих приложениях апостериорные вероятности необходимы для оценивания рисков, связанных с возможными ошибками классификации.

4.3.1.3 Классификация с помощью Случайного Леса

Случайный лес - композиция глубоких деревьев, которые строятся независимо друг от друга.

Чтобы построить случайный лес из N решающих деревьев, необходимо:

1. Построить с помощью бутстрапа N случайных подвыборок

$$\tilde{X}_n, n = 1, ..., N$$

2. Каждая получившаяся подвыборка $\tilde{X}_n$ используется как обучающая выборка для построения соответ- ствующего решающего дерева $b_n(x)$. Причем:

   - Дерево строится, пока в каждом листе окажется не более n_{min} объектов. Очень часто деревья строят до конца ($n_{min} = 1$), чтобы получить сложные и переобученные решающие деревья с низким смещением.

- Процесс построения дерева рандомизирован: на этапе выбора оптимального признака, по которому будет происходить разбиение, он ищется не среди всего множества признаков, а среди случайного подмножества размера q.
- Следует обратить особое внимание, что случайное подмножество размера q выбирается заново каж- дый раз, когда необходимо разбить очередную вершину. В этом состоит основное отличие такого подхода от метода случайных подпространств, где случайное подмножество признаков выбиралось один раз перед построением базового алгоритма.

3. Построенные деревья объединяются в композицию:

- В задачах регрессии $a(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x)$
- В задачах классификации $a(x) = sign(\frac{1}{N}\sum_{n=1}^{N} b_n(x))$.

4.3.1.4 Классификация с помощью Градиентного бустинга над деревьями

Классификация с помощью случайного леса деревьев имеет следующуие проблемы. Обучение глубоких деревьев требует очень много вычислительных ресурсов, особенно в случае большой выборки или большого числа признаков. Если ограничить глубину решающих деревьев в случайном лесе, то они уже не смогут улавливать сложные закономерности в данных. Это приведет к тому, что сдвиг будет слишком большим. Вторая проблема со случайным лесом состоит в том, что процесс построения деревьев является ненаправленным: каждое следующее дерево в композиции никак не зависит от предыдущих. Из-за этого для решения сложных задач необходимо огромное количество деревьев

В задаче бинарной классификации ($Y = -1, +1$) популярным выбором для функции потерь является логистическая функция потерь:

$$\sum_{i=1}^{n} log(1 + exp(-y_i a(x_i)))$$

где $a(x) \in R$ — оценка принадлежности положительному классу. Если $a(x) > 0$, классификатор относит объект x к классу $+1$, а при $a(x) \leq 0$ - к классу -1.

Причем, чем больше $|a(x)|$, тем больше классификатор уверен в своем выборе. Функция потерь в этом случае записывается следующим образом:

$$L(y, z) = log(1 + exp(-yz)),$$

$$L_z^{`}(y, z) = -\frac{y}{1 + exp(yz)}.$$

Вектор сдвигов s в этом случае будет иметь вид:

$$\mathbf{x} = \begin{bmatrix} \frac{y_1}{1+exp(y_1 a_{N-1}(x_1))} \\ ... \\ \frac{y_l}{1+exp(y_l a_{N-1}(x_l))} \end{bmatrix}$$

Новый базовый алгоритм будет настраиваться таким образом, чтобы вектор его ответов на объектах обучающей выборки был как можно ближе к s. После того, как вычислен алгоритм $a_N(x)$, можно оценить вероятности принадлежности объекта x к каждому из классов:

$$P(y = 1|x) = \frac{1}{1 + exp(-a_N(x))}$$

$$P(y = -1|x) = \frac{1}{1 + exp(a_N(x))}$$

4.3.2 Классификация с помощью сверточных нейронных сетей

Чтобы решить эти проблемы классификации временных рядов, была использована многомасштабную структуру сверточной нейронной сети (MCNN), в которой ввод представляет собой временной ряд, который должен быть предсказан, а выход - его меткой. Общая архитектура MCNN показана на рисунке 11. Данная архитектура была представлена в статье Zhicheng Cui, Wenlin Chen и Yixin Chen [19] Структура MCNN имеет три последовательных этапа: трансформация, локальную свертку и полномасштабную свертку.

1. На этапе трансформации применяются различные преобразования во входных временных рядах. Производится преобразование прямого отображения, преобразование с уменьшением выборки во временной области и спектральные преобразования в частотной области. Каждая часть называется ветвью, так как они является ветвями, входящим в сверточную нейронную сеть.

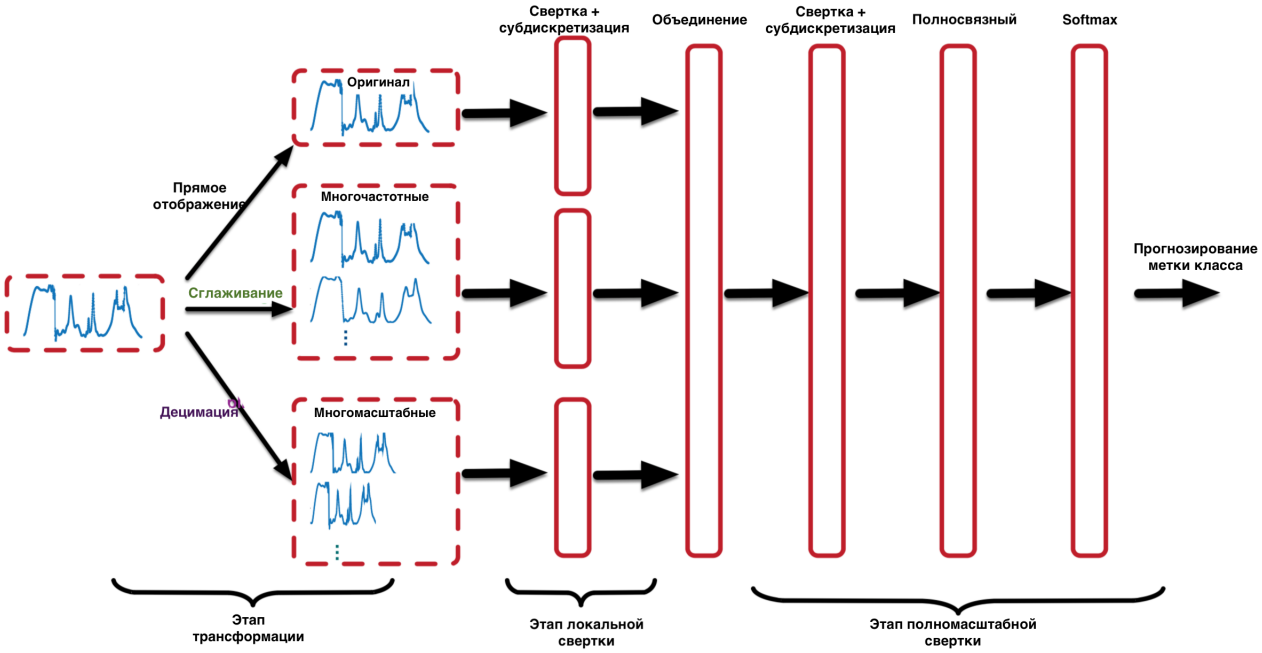


Рис. 11: Схема классификатора CNN

2. На этапе локальной свертки используется несколько слоев для извлечения компонентов для каждой ветви. На этом этапе свертки для разных ветвей независимы друг от друга. Все выходы проходят через процедуру субдескритизацией (max pooling) с несколькими размерами.

3. На стадии полномасштабной свертки объединяются все извлеченные признаки и применяется еще несколько сверточных слоев (каждый из которых следует за субдескритизацией (max pooling)), полносвязные слои и слой softmax используются для генерации конечного результата. Все параметры обучаются совместно посредством обратного распространения ошибки.

4.3.2.1 Этап трансформации

Ветвь с различными длиннами. Устойчивая модель классификации временных рядов должна иметь возможность фиксировать закономерности поведения временного ряда в разных временных масштабах. Долгосрочные характеристики отражают общие тенденции, а краткосрочные признаки указывают на незначительные локальные изменения, которые могут быть потенциально важными для качественной классификации.

В данной ветви MCNN мы используется с выводом для создания эскизов

временного ряда в разных временных масштабах. Предположим, что у нас есть временные ряды $T = t_1, t_2, ..., t_n$, а частота дискретизации равна k, тогда мы будем хранить только каждую k-ый точку нового временного ряда.

$$T^k = t_{1+k*i}, i = 0, ..., \lfloor \frac{n-1}{k} \rfloor$$

Этот метод используется для того, чтобы сгенерировать множество новых входных временных рядов (используя $k = 2, 3, ...$).

Ветви различных частот. Во временных рядах часто встречаются высокочастотные возмущения и случайные шумы, что создает еще одну проблему для достижения высокой точности прогнозирования. Часто бывает трудно извлечь полезную информацию о необработанных временных рядах данных с присутствием этих шумов. В MCNN применяются низкочастотные фильтры с несколькими степенями гладкости для решения этой проблемы. Низкочастотный фильтр может уменьшить дисперсию временных интервалов. В частности, используется скользящее среднее для достижения этой цели. С учетом входных данных, генерируется несколько новых временных рядов с разной степенью гладкости с использованием скользящей средней с разными размерами окон. Таким образом, вновь создаваемые временные ряды представляют собой общую низкочастотную информацию, которая делает тенденцию временных рядов более ясной. Предположим, что исходный временной ряд $T = t_1, t_2, ..., t_n$, скользящее среднее преобразует этот исходный временной ряд в новый временной ряд

$$T_l = \frac{x_i + x_{i+1} + ... + x_{i+l-1}}{l}$$

где l - размер окна и $i = 0, 1, ..., n - l + 1$. При разных l MCNN генерирует несколько временных рядов разных частот, все из которых будут подаваться в локальный сверточный слой этой ветви. В отличие от ветви с различными длиннами каждый временной ряд в многочастотной ветви имеет одинаковую длину, что позволяет нам собирать их в несколько каналов для следующего сверточного слоя.

4.3.2.2 Этап локальной свертки

Локальная свертка. После преобразования рядов мы получаем несколько временных рядов с разной длиной от одного входного временного ряда.

Применяются независимые одномерные локальные свертки на каждом из этих вновь созданных временных рядов. В частности, размер фильтра локальной свертки будет одинаковым во всех этих временных рядах. Обратите внимание, что при одном и том же размере фильтра более короткие временные ряды будут иметь разный выход. Таким образом, каждый выход с этапа локальной свертки захватывает другой масштаб исходного временного ряда. Преимущество этого метода состоит в том, что, уменьшая выборку временного ряда вместо увеличения размера фильтра, мы можем значительно уменьшить количество параметров в локальном сверточном слое.

Субдискретизация с несколькими размерами. Субдискретизация также выполняется между последовательными сверточными слоями в MCNN. Это может уменьшить размер пространства признаков, а также количество параметров следующих слоев, чтобы избежать переобучения и повышения эффективности вычислений. Что более важно, субдискретизация приводит к инвариантности пространственного сдвига, что делает MCNN более надежным.

Вместо использования небольших размеров параметра субдескретизации, в MCNN мы вводим переменную, называемую коэффициентом объединения p, которая является длиной после субдискретизации. Предположим, что время ряд после свертки имеет длину n, тогда и размер объединения, и шаг субдискретизации - $\frac{n}{p}$. Размер объединения p довольно большой, так как p часто выбирается из $2, 3, 5$. Таким образом мы можем создать большее количество фильтров и применять каждый фильтр для изучения только локальной функции, так как на этапе обратного распространения ошибки фильтры будут обновляться на основе тех немногих активированных частей свертки.

4.3.2.3 Этап полномасштабной свертки

После извлечения признаков из нескольких ветвей данные объединяются и передаются другие уровни свертки, а также в полносвязный уровень, за которым следует преобразование softmax. Следуя [20], мы применяем метод глубокой конкатенации, чтобы объединить все признаки.

На выходе MCNN будет предсказанное распределение каждой возможной метки для входных временных рядов. Для обучения нейронной сети MCNN

использует кросс-энтропийную потерю, определяемую как:

$$\max_{W,\mathbf{b}} \sum_{i=1}^{N} log(o_{y_i}^{(i)}),$$

где $o_{y^i}^{(i)}$ является y_i-ым выходом экземпляра i через нейтральную сеть, что является вероятностью того, что экземпляр принадлежит этому классу. Параметры W и смещение $\mathbf{b}$ в MCNN - это параметры в локальных и полных сверточных слоях, а также в полносвязных слоях, все из которых изучаются совместно с помощью обратного распространения ошибки.

4.4 Результаты

4.4.1 Метрики качества

Точность и полнота. Точность (precision) и полнота (recall) являются метриками которые используются при оценке большей части алгоритмов извлечения информации.

Точность системы в пределах класса – это доля объектов действительно принадлежащих данному классу относительно всех объектов которые система отнесла к этому классу. Полнота системы – это доля найденных классификатором объектов принадлежащих классу относительно всех объектов этого класса в тестовой выборке. Эти значения легко рассчитать на основании таблицы контингентности, которая составляется для каждого класса отдельно.

	Положительная разметка	Отрицательная разметка
Положительная оценка	TP	FP
Отрицательная оценка	TN	FN

В таблице содержится информация сколько раз система приняла верное и сколько раз неверное решение по документам заданного класса. А именно:

- TP — истино-положительное решение;
- TN — истино-отрицательное решение;

- FP — ложно-положительное решение;
- FN — ложно-отрицательное решение.

Тогда, точность и полнота определяются следующим образом:

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$

В случае многоклассовой классификации точность и полнота рассчитываются с использованием матрицы неточностей (confusion matrix).Матрица неточностей – это матрица размера N на N, где N — это количество классов. Столбцы этой матрицы резервируются за разметкой, а строки за решениями классификатора. Когда мы классифицируем объект из тестовой выборки мы инкрементируем число стоящее на пересечении строки класса который вернул классификатор и столбца класса к которому действительно объект документ. Тогда точность и полнота для класса c вычисляются, как:

$$Precision_c = \frac{A_{c,c}}{\sum_{i=1}^{n} A_{c,i}}$$

$$Recall_c = \frac{A_{c,c}}{\sum_{i=1}^{n} A_{i,c}}$$

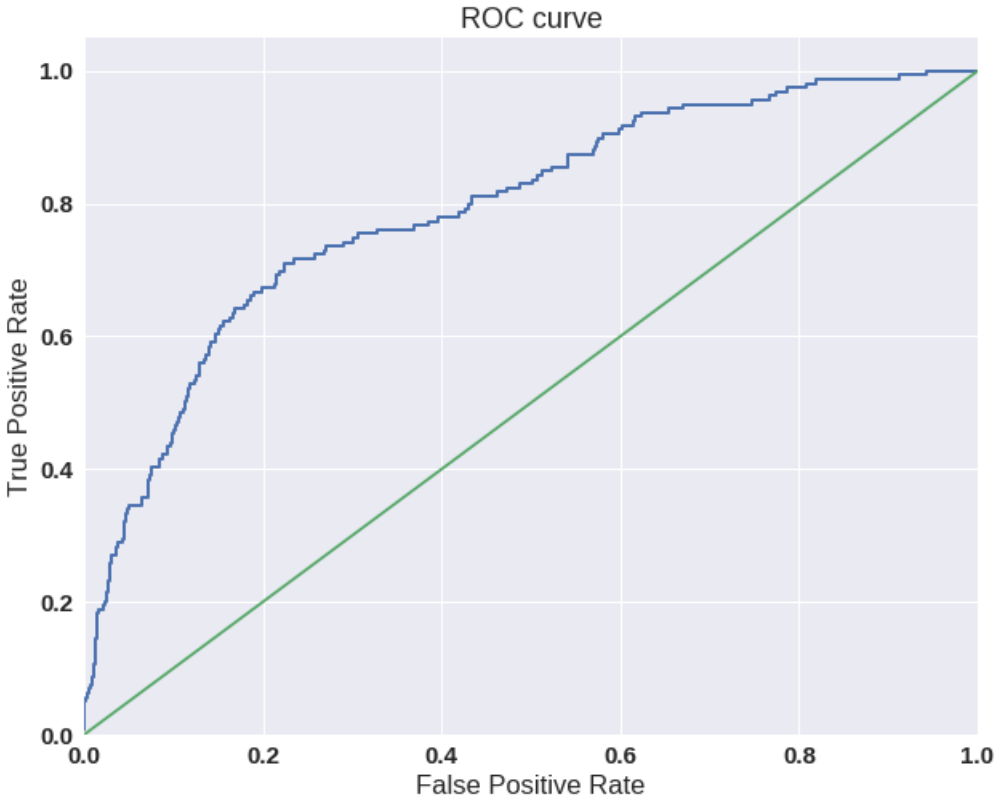
F-мера. F-мера представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю. Вычисляется по формуле:

$$F = 2\frac{Precision \times Recall}{Precision + Recall}$$

Площадь под ROC - кривой. Кривая ошибок или ROC-кривая – графичекая характеристика качества бинарного классификатора, зависимость доли верных положительных классификаций от доли ложных положительных классификаций при варьировании порога решающего правила. Преимуществом

ROC-кривой является её инвариантность относительно отношения цены ошибки I и II рода.

ROC-кривая показывает зависимость TPR от FPR при варьировании порога w_0. Она проходит из точки $(0, 0)$, соответствующей максимальному значению w_0, в точку $(1, 1)$, соответствующую минимальному значению w_0.



Большинство метрик описанных выше работают только для бинарной классификации. Для того, чтобы обобщить их на многоклассовый слущай задача сводится к бинарной классификации отделяющий каждый класс от всех остальных. Далее считаются метрики качества и усредняются. Существует 2 метода усреднения:

1. Микро-усреднение (micro-averaging):

   - Находятся метрики качества из кажой задачи;
   - Метрики усредняется их по всем задачам;
   - Вычисляется итоговую метрику.

Вклад каждого класса не завичит от его размера.

2. Макро-усреднение (macro-averaging):

   - Вычисляется итоговую метрику для каждой из задач;
   - Производится усреднение по всем классам

В качестве метрик для многоклассовой классификации в данной работе используются метрики усредненные методом микро-усреднения.

4.4.2 Обнаружение аномалии

В качестве метрик качества для задачи обнаружения аномалий использовались точность, полнота и F-мера. Чтобы вычислить их, каждый тестовый временной ряд разбивался на равные интервалы и проверялось, превышает ли среднеквадратичная ошибка пороговый уровень. Такой эксперимент проводился для различных длин интервалов. Регулируя значение порога можно решить проблему ложных срабатываний.

Таблица 3: Результаты ARIMA

ξ	**Precision**	**Recall**	**F-мера**
25	0,371	0,328	0,348
50	0,572	0,462	0,511
75	0,684	0,701	0,692
100	0,714	0,752	**0,733**
125	0,599	0,845	0,701

Таблица 4: Результаты LSTM

ξ	**Precision**	**Recall**	**F-мера**
25	0,295	0,302	0,299
50	0,481	0,412	0,444
75	0,571	0,592	0,581
100	0,642	0,621	**0,631**
125	0,479	0,773	0,592

Таблица 5: Сравнение лучших результатов

Метод	**Precision**	**Recall**	**F-мера**
ARIMA	0,642	0,621	**0,631**
LSTM	0,714	0,752	**0,733**

4.4.3 Классификация аномалии

Для тестирования алгоритмов классификации на вход классификатора подавались временные ряды разной длинны (нормальная работа в различных режимах и различные типы аномалий). Ниже приведена таблица, описывающие результаты работы различных алгоритмов.

Таблица 6: Результаты работы различных алгоритмов классификации

Алгоритм классификации	**значение ROC-AUC**
Наивный Байесовский классификатор	0.621
Логистическая Регрессия	0.683
Случайный Лес	0.792
Градиентный бустинг над деревьями	**0.838**
Сверточная нейронная сеть	0.787

4.5 Заключение и дальнейшая работа

В рамках работы было произведено исследование и сравнительный анализ существующих алгоритмов, решающих задачу обнаружения и классификации аномалий, а так же был предложен метод классификации, основанной на сверточных нейронных сетях, который ранее не применялся в данной области. Как видно из результатов исследования, метод обнаружения аномалий, основанный на грубоких нейронных сетях показал гораздо большую эффективность нежели статистический метод. В задаче классификации модель, использующая глубокие нейронные сети, уступает в производительности различным композициям решающих деревьев. Но данный метод показал себя, как жизнеспособный и может быть улучшен.

Результатом работы является не только всестороннее исследование и теоретическое обоснование теории, связанной с анализом временных рядов, но также и прототип, написанный на языке Python. Данная программа может быть использована для запуска обнаружения аномального поведения в системе. Ее работа состоит из двух стадий: обучения и обнаружения. На этапе обучения программа собирает данные о нормальном поведении системы и учится максимально точно прогнозировать ее следующие шаги. В режиме обнаружения программа прогнозирует поведение системы основываясь на ее текущей истории работы, если ошибка прогноза выше порогового значения, то данный временной отрезок считается аномальным. Далее аномальный отрезок времени подается на вход обученному классификатору, где определяется тип аномалии.

На данный момент данная модель работает только с набором данных Tennessee Eastman Process. В дальнейшем планируется протестировать ее работу на других искуственных данных и провести испытания на реальной вычислительной системе. Так же планируется улучшить работу существующих алгоритмов обнаружения и классификации аномалий, а так же расширить их список.

Список используемой литературы

[1] Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: классификация и снижение размерности. — М.: Финансы и статистика, 1989.

[2] Arindam Banerjee Varun Chandola and Vipin Kumar, *Anomaly detection : A survey, ACM Computing Surveys, 2009.*

[3] Eilertson E. Lazarevic A. Tan P.N. Kumar V. Srivastava J. Ertoz, L. and P. Dokas. Minds - minnesota intrusion detection system. In Data Mining - Next Generation Challenges and Future Directions. MIT Press., 2004.

[4] Parra L. Spence, C. and P. Sajda. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In In Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis. IEEE Computer Society, Washington, DC, USA, 3, 2001.

[5] Freisleben B. Aleskerov, E. and Rao. Cardwatch: A neural network based database mining system for credit card fraud detection. In In Proceedings of IEEE Compu- tational Intelligence for Financial Engineering. 220-226, 1997.

[6] Malik Agyemang, Ken Barker, and Rada Alhajj. A comprehensive survey of numeric and symbolic outlier mining techniques. Intell. Data Anal., 10(6):521–538, 2006.

[7] Arindam Banerjee Varun Chandola and Vipin Kumar. Anomaly detection for discrete sequences : A survey. Unpublished Work.

[8] S. Akhavan and G. Calva. Automatic anomaly detection in ecg signal by fuzzy decision making. In In Proceedings of 6th International Conference on Fuzzy Theory and Technology: Association for Intelligent Machinery, 23-28, 1998.

[9] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. Artif. Intell. Rev., 22(2):85–126, 2004.

[10] Downs, J and Vogel, E. A plant-wide industrial process control problem. Computers and chemical engineering, 17(3):245–255, 1993.

[11] Krotofil, M. Damn vulnerable chemical process, 2014. URL http://github.com/satejnik/ DVCP-TE.

[12] A. J. Fox. Outliers in time series. Journal of the Royal Statistical Society. Series B(Methodological), 34(3):350–363, 1972

[13] Pavel Filonov, Fedor Kitashov, Andrey Lavrentyev, RNN-based Early Cyber-Attack Detection for the Tennessee Eastman Process

[14] Xing, Z., Pei, J., and Keogh, E. (2010). A brief survey on sequence classification. ACM SIGKDD Explorations Newsletter, 12(1):40–48.

[15] Keogh, E. J. and Kasetty, S. (2003). On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Min. Knowl. Discov., 7(4):349–371.

[16] Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. Acoustics, Speech and Signal Processing, IEEE Transactions on, 26(1):43–49.

[17] Filonov, P, Lavrentyev, A, and Vorontsov, A. Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model. NIPS 2016 Time Series Workshop papers, 2016. URL http://arxiv.org/abs/1612.06676.

[18] Sepp Hochreiter and Ju ?rgen Schmidhuber. Long short-term memory. Neural Comput., 9(8):1735– 1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi. org/10.1162/neco.1997.9.8.1735.

[19] Zhicheng Cui, Wenlin Chen, Yixin Chen, Multi-Scale Convolutional Neural Networks for Time Series Classification

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–9, 2015.

[21] Liao, T. W. (2005). Clustering of time series data - a survey. Pattern Recognition, 38(11):1857–1874.

[22] Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286.

[23] Laxman, S. and Sastry, P. (2006). A survey of temporal data mining. Sadhana, 31(2):173–198.

[24] Baldi, P., Chauvin, Y., Hunkapiller, T., and McClure, M. (1994). Hid- den markov models of biological primary sequence information. Proceedings of the National Academy of Sciences, 91(3):1059–1063.

[25] Giles, C., Lawrence, S., and Tsoi, A. (2001). Noisy time series prediction using recurrent neural networks and grammatical inference. Machine Learning, 44(1):161–183.

[26] Graves, A., Fern Sandez, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd international conference on Machine learning, pages 369–376. ACM.

[27] J. Ma and S. Perkins. Time-series novelty detection using one-class support vector machines. volume 3, pages 1741–1745 vol.3, 2003.

[28] P. Protopapas, J. M. Giammarco, L. Faccioli, M. F. Struble, R. Dave, and C. Alcock. Finding outlier light-curves in catalogs of periodic variable stars. MON.NOT.ROY.ASTRON.SOC., 369:677, 2006.

[29] L. Rabiner and B. Juang. An introduction to hidden markov models. ASSP Mag- azine, IEEE, 3(1):4–16, Jan 1986.

[30] Veselina Jecheva. About some applications of hidden markov model in intrusion detection systems. In International Conference on Computer Systems and Technologies - CompSysTech, 2006.

[31] Shrijit S. Joshi and Vir V. Phoha. Investigating hidden markov models capabilities in anomaly detection. In ACM-SE 43: Proceedings of the 43rd annual Southeast regional conference, pages 98–103, New York, NY, USA, 2005. ACM.

[32] Hai-Tao He and Xiao-Nan Luo. A novel hmm-based approach to anomaly detection. 2004.

[33] Ninad Thakoor and Jean Gao. Hidden markov model based 2d shape classification, 2005.

[34] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The Annals of Mathematical Statistics, 41(1):164–171, 1970.

[35] Ji, X., Bailey, J., and Dong, G. (2005). Mining minimal distinguishing subsequence patterns with gap constraints. In Data Mining, Fifth IEEE International Conference on, pages 8–pp. IEEE.