Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра математических методов прогнозирования

Романов Никита Алексеевич

Детектирование аномалий во временных рядах при помощи глубоких нейронных сетей

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Научный руководитель:

Д. А. Кропотов

Москва, 2018

Оглавление

Глава 1

Введение

В современном мире существует запрос на «цифровизацию» процессов. Крупные компании в России и в мире стремятся автоматизировать процессы на всех уровнях, в том числе активно применяя методы машинного обучения. Очевидно, что промышленные предприятия не могли обойти этот тренд стороной и сейчас активно внедряют новые технологии в свои производства. Спектр задач, интересных для заводов, крайне широк и варьируется от бизнес-аналитики и анализа процессов до компьютерного моделирования производственной цепочки. Достаточно интересной задачей, для которой на данный момент не существует гарантированно приводящего к успеху решения, является предиктивная аналитика. В рамках такого подхода предлагается попытаться смоделировать события в производственном процессе на некоторое время вперед и, используя эти знания, сделать выводы, которые помогут оптимизировать процесс.

Важной подзадачей предиктивной аналитики является детектирование аномалий. Аномалия - это некоторые отклонения производственного процесса от обычного для него течения, такие как увеличение продолжительности какой-то стадии процесса или изменение его параметров и т.д. В заводских условиях, где все подчинено физике, типичное поведение многократно смоделированно при помощи математики, аномалии как минимум вызывают интерес. Очень часто по ним можно судить о том, что оборудование вышло из строя или работает в нестандартном режиме.

К примеру на нефтеперерабатывающих заводах используется такой аппарат как ректификационная колонна. Это высокая колонна, в которой при массо- и теплообмене происходит разделение на фракции нефти. Для поддержания нужной температуры в тепловом контуре циркулирует водяной пар. Его оптимальное давление

рассчитывается при помощи математической модели. Если построить график изменения давления пара в долгосрочной перспективе, то можно обнаружить 3 последовательных шаблона поведения, которые отличаются «резкостью» смены значений. Первый режим работы - стандартный, для него подходят разработанные модели управления. Во втором режиме колонна может работать нормально, но при частой корректировке давления, что увеличивает стоимость эксплуатации. В третьем же колонна практически сразу выходит из строя. Следует отметить, что ремонт колонны - очень дорогой и долгий процесс. Подробный анализ данных помог установить, что вышеуказанные режимы связаны с уровнем внутреннего загрязнения колонны. Таким образом, если суметь предугадать время перехода из одного состояния в другое, анализируя давление пара, можно также предугадать и уровень загрязнения колонны. Отправляя колонну на плановую чистку в конце второго режима можно максимально продлить срок эксплуатации колонны, сэкономив существенные ресурсы.

Тем не менее, неполадки на заводах - явление достаточно редкое. Производство должно быть максимально прогнозируемым и безаварийным, поэтому существенные аномалии на заводах встречаются редко. Так как в настоящее время стоимость накопителей информации низкая, а на современных заводах установлено огромное количество датчиков, то накопление большой выборки информации не является проблемой. Однако практически вся информация будет соответствовать нормальному течению процессов. Данный факт резко ограничивает пользу методов обучения с учителем, так как для их нормального использования обычно необходимо следить за сбалансированностью выборки. В худшем случае (разумеется для исследователя, а не завода) примеры аномалий могут вообще отсутствовать.

Вторая сложность в детекции аномалий заключается в сложности разметки выборки. В примере с заводом выше можно было бы построить регрессию уровня засора от давления пара. Но сложность в том, что загрязнение колонны не оценивалось датчиками. Колонна редко останавливается, ее фотографируют, после чего технологи решают, в каком состоянии сейчас агрегат. Вполне возможно, что колонну остановят для осмотра раз за весь срок ее службы. При большом числе информации с измерительной аппаратуры данных об аномалиях может быть крайне мало.

Третья сложность - это локализация аномалии. Проблема связана с двумя вышеуказанными. Не всегда, даже когда известно, что процесс пошел не так, легко указать когда именно это случилось (и, возможно, закончилось). В примере выше локализовать аномалии на обучающей выборке вообще невозможно, так как плановые проверки проходят редко, а увеличить частоту проверок даже для сбора данных невозможно из-за затратности таких мероприятий.

Задачей данной работы является создание метода, способного детектировать аномалии даже при условии вышеуказанных ограничений. Основной упор сделан на использование рекуррентных нейронных сетей. Данная модель хорошо зарекомендовала себя при обработке длинных последовательностей данных, в том числе и для задач предиктивной аналитике на производстве. Кроме метода детекции аномалий будет рассмотрен способ исключения несущественных данных из модели.

Дальнейшее изложение будет построено по следующему плану:

- в главе 2 подробно разбирается теоретическая часть задачи. В разделе 2.1 ставится формальная постановка задачи. В разделе 2.2 рассматриваются существующие методы детекции аномалий. Разделы 2.3 - 2.4 описывают предложенный в работе подход детектирования аномалий и разреживания данных;
- в главе 3 проводятся эксперименты с построенными моделями;
- в главе 4 делаются подводятся итоги и анализируются результаты предложенного метода.

Глава 2

Методы детектирования аномалий

В данном разделе вводится теоретический базис настоящей работы. Дается краткий обзор существующих методов, предлагается новый метод, а также способ, позволяющий отделять неинформативные данные от важных. Описанный подход предполагает использование рекурретных нейронных сетей, но возможно использование и других видов нейросетей.

2.1 Постановка задачи

Пусть дан многомерный временной ряд $X = \{x_1, x_2, \ldots, x_L\}$, где каждый элемент $x_t \in \mathbf{R}^m$ - m-мерный вектор, представляющих собой показания m датчиков в момент времени t. Рассматривается только тот случай, когда такие показания возможно получить. При этом часть сигналов $x_t^{(c)} \in \mathbf{R}^c$ может поступать с датчиков управления, а часть $x_t^{(r)} \in \mathbf{R}^r$ - с откликов, $m = r + c$. Первые рассчитываются некоторой моделью и управляют агрегатами завода, вторые считывают фактические показания с агрегатов. Необходимо каждому моменту времени t поставить в соответствие некоторое число $a_t \in \mathbf{R}$, называемое рейтингом аномальности, показывающее насколько нетипичны в данный момент значения временного ряда. При этом важно отметить, что в обучающей выборке нет или практически нет аномальных значений. Сами аномалии, в общем случае, могут иметь произвольную природу и не иметь никаких общих признаков, то есть не могут быть сгруппированы в один класс на основании некоторой общности. Разные временные ряды X_1 и X_2 могут иметь разные длины $L_1 \neq L_2$, но число датчиков всегда одинаково $m_1 = m_2 = m$. Если временной ряд очень длинный (например, для случаев непрерывного производства), то он может разбиваться на подряды меньшей длины, исходя из возможных вычислительных мощностей и специфики задачи.

2.2 Обзор существующих методов

Существует множество разных методов детекции аномалий, однако все они используют не очень большое количество идей. Самым простым и очевидным решением является существенно использовать свойство интерполяторов сильно ошибаться на тех данных, которые не использовались при обучении модели [1], [2], [3]. Однако такой подход вынуждает учить прогнозную модель, что может быть по каким-то причинам нежелательно. Второй проблемой является трактование прогноза модели. Разумеется, что модель будет расходиться с реальными данными, но как понять, какое расхождение является аномальным, а какое - допустимым, является тем, что отличает эти методы.

Модели, опирающиеся на расстояние (distance-based) в качестве рейтинга аномальности a_t используют некоторую метрику [4], [5], [6]. Предполагается, что нормальные данные расположены ближе друг к другу, чем аномальные. Например, расстояние основанное на методе ближайших соседей:

$$a_t = \sum_{x_i \in kNN(x_t)} D(x_i, x_t), \tag{2.1}$$

где D - некоторая метрика, $kNN(x_t)$ - k ближайших соседей x_t. Стоит отметить, что формула (2.1) была изначально разработана для независимых наблюдений, а точки временного ряда таковыми не являются. Поэтому в лоб использовать такой подход не совсем разумно, необходима предобработка данных, например, [7]. Недостатком такого семейства методов является неочевидная процедура подбора порога τ, при превышении которого можно считать $a_t > \tau$ данные аномальными, так как $a_t \in \mathbf{R}_+$ - неограниченная величина. К особенным вариантам применения напрямую для временных рядов среди таких методов можно выделить SAX [8] и HOT SAX [9].

Если нормальные данные группируются в несколько плотных кластеров, то предыдущий подход обычно показывает плохое качество. Модели, опирающиеся на плотности (density-based) считают, что нормальные данные собираются в более плотные группы, чем аномальные. Метод фактора локального выброса (LOF) [10] в качестве рейтинга аномальности использует:

$$a_t = \frac{1}{k} \sum_{x_i \in kNN(x_t)} \frac{LD(x_i)}{LD(x_t)}, \tag{2.2}$$

где плотность

$$LD(x_t) = \left(\frac{1}{k} \sum_{x_i \in kNN(x_t)} RD(x_i, x_t) \right)^{-1},$$

$$RD(x_i, x_t) = \max\left(D(x_i, x_t), D(x_t, kNN_i(x_t))\right).$$

Здесь $kNN_i(x_t)$ - i-ближайший сосед для x_t. Считается, что у нормальных данных при таком подходе $a_t \approx 1$, а у аномальных $a_t \gg 1$. Выбор корректного значения τ и тут затруднителен, поэтому существуют модификации вроде LoOP [11], которые позволяют трактовать a_t как вероятность. Более продвинутыми методами, использующими механизм, близкий к LOF, непосредственно к временным рядам, являются KNN-ICAD и LOF-ICAD [12].

К идее собрать все нормальные данные в ряд плотных кластеров можно отнести лес изолирующих деревьев [13] и одноклассовую машину опорных векторов [14].

Возвращаясь к прогнозным моделям, можно также считать, что ошибка алгоритма или само предсказание подчиняются некоторому распределению вероятности. Тогда можно оценивать правдоподобие модели, а по нему выносить решение об аномальности. Это используется в [2], [1] и будет рассмотрено подробнее далее.

2.3 Детектирование аномалий с помощью правдоподобия

Для того, чтобы детектировать аномалии, необходимо уметь восстанавливать нормальное течение процесса. Если модель получила достаточно богатую выборку, в которой содержалось большинство сценариев нормальной работы, то можно надеяться, что после обучения она сможет успешно решить такую задачу регрессии. При этом, если модель является слабым экстраполятором, то она не сможет давать адекватные оценки на тех данных, которые «не видела» ранее.

Такими свойствами обладают практически все алгоритмы машинного обучения. Однако мы подробнее остановимся на использовании рекуррентных нейронных сетей. Это обусловлено двумя причинами. Во-первых, рекуррентные сети естественным образом поддерживают временные ряды. Для того, чтобы учесть зависимость от времени в данных, для такой модели нужно лишь подавать векторы x_t в нужном порядке без применения сложного конструирования признаков (которое обычно само по себе является сложной задачей, специфичной для каждого отдельного случая). Во-вторых, на данный момент нейронные сети дают самое высокое качество регрессии в подобных задачах, когда данные на тестовой выборке похоже на данные в обучении, но при этом сильно ошибаются в противоположном случае. Такой сильный контраст, являющийся во многих случаях недостатком, для детектирования редких событий является ключевым фактором успеха.

На первом шаге работы алгоритма временной ряд X разбивается на окна длины l. Модель принимает часть ряда $X_{t,\dots,t+l} = \{x_t, x_{t+1}, \dots, x_{t+l}\}$, и по ней предсказывает следующую часть ряда такой же длины $\hat{X}_{t+l,\dots,t+2l} = \{\hat{x}_{t+l+1}, \hat{x}_{t+l+2}, \dots, \hat{x}_{t+2l}\}$. Далее спрогнозированные значения сравниваются с фактическими и вычисляются векторы ошибок $e_t = x_t - \hat{x}_t$ для каждого момента времени t. Норма величины ошибки $||e_t||_2 = \sqrt{\sum_{i=1}^{m} e_{ti}^2}$ или $||e_t||_1 = \sum_{i=1}^{m} |e_{ti}|$, где e_{ti} - i-компонента вектора e_t, уже может являться хорошим рейтингом аномальности a_t, однако далее будут рассмотрены лучшие способы построить отображение $f(e_t) = a_t$.

Важно отметить, что необязательно восстанавливать все значения сигналов, если того не требует задача. Часто на практике об аномалии говорит отклонение именно контрольных детекторов $x_t^{(r)}$, поэтому возможно восстанавливать только эти значения, вместо x_t. Впрочем, этот вопрос сильно зависит от специфики задачи.

Величина l, вообще говоря, является гиперпараметром метода, так как от нее может зависеть качество регрессии. Тем не менее, не всегда выбор этой величины должен руководствоваться только соображениями качества. Слишком большая величина окна требует времени на сбор информации датчиками, что может замедлить систему. В итоге это может привести к случаю, когда детектировать аномалии уже поздно и оборудование уже сломалось, а сенсоры все еще собирают информацию для модели. С другой стороны маленькое окно, во-первых, обычно менее точно восстанавливает регрессию, а с другой увеличивает число проходов по модели, что может быть вычислительно затратно для сложной нейронной сети.

Итак, итоговая схема алгоритма:

- разбить итоговый сигнал X на окна длины l;
- по данным в окне $X_{t,\dots,t+l}$ предсказать следующее окно $\hat{X}_{t+l+1,\dots,t+2l}$;
- вычислить векторы ошибок $e_t = x_t - \hat{x}_t$;
- по некоторому закону построить рейтинги аномальности $a_t = f(e_t)$.

Далее подробнее будут рассмотрены вопросы восстановления регрессии и построения отображения $f(e_t) = a_t$ на основе правдоподобия.

2.3.1 Обычная рекуррентная модель и модель Seq-to-Seq

В глубинном обучении наиболее хорошо себя зарекомендовали не обычные рекуррентные нейронные сети, а их модификации, такие как LSTM [16] или GRU [17].

Это более сложные системы, которые могут «запоминать» очень длинные последовательности, что недоступно для обычных сетей. Кроме того, скорость сходимости таких архитектур обычно выше. Практически все работы, связанные с анализом последовательностей, используют их в качестве основных вычислительных блоков.

Обучать рекуррентные сети можно по-разному. Существует обычный способ, когда множество слоев обрабатывают вход последовательно слой за слоем также как в других видах нейронных сетей (полносвязных или сверточных). Такой вариант изображен на рис. 2.1. Каждый вектор входных данных x_t видоизменяется в последовательность скрытых состояний $h_t^{(1)}, h_t^{(2)}, \ldots, h_t^{(n)}$ по числу слоев в сети и из последнего состояния восстанавливается $\hat{x}_t$.

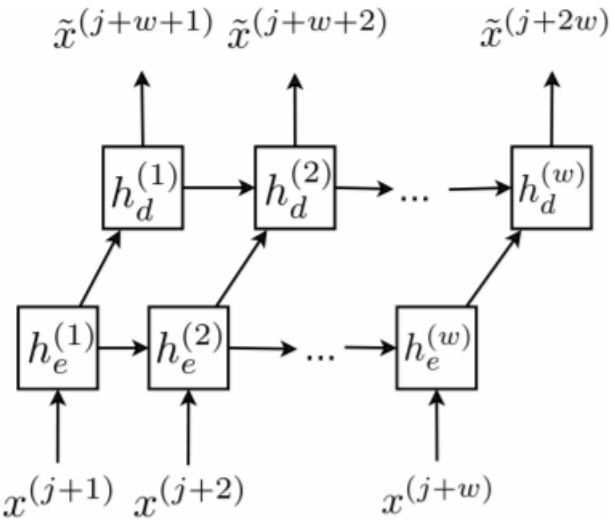


Рис. 2.1: Стандартная схема многослойной рекуррентной сети из [1]. На изображении вместо нижних индексов используются верхние.

Нетрудно заметить, что первый вариант не может обрабатывать последовательности произвольной длины (отобразить вход произвольной длины в выход произвольной длины). Несмотря на то, что в данной задаче такое не требуется, имеет смысл упомянуть про второй вариант обучения, модель seq-to-seq, который хорошо зарекомендовал себя в задачах машинного перевода. Для него естественным образом можно использовать механизмы внимания [18], а также иногда он дает лучшее качество и в моделях, где не требуется выравнивание входов или выходов.

Опишем его подробнее. Модель состоит из двух частей - энкодера и декодера. Каждый из них может содержать несколько рекуррентных блоков, причем необязательно энкодер и декодер должны совпадать друг с другом. Архитектура seq-to-seq сети изображена на рис. 2.2. Энкодер принимает на вход последовательность $X_{t,\ldots,t+l}$ и строит последовательность состояний $h_t^e, h_{t+1}^e, \ldots, h_{t+l}^e$. Последнее состояние энкодера h_{t+l}^e используется как стартовое для декодера. На основе него он предсказывает

первый элемент выхода $\hat{x}_{t+l+1}$. Далее декодер использует $\hat{x}_{t+l+1}$ и первое состояние декодера h^d_{t+l+l}, чтобы предсказать $\hat{x}_{t+l+2}$ и h^d_{t+l+2} и т.д., пока не восстановит всю последовательность $\hat{X}_{t+l,\ldots,t+2l}$. Энкодер и декодер учатся совместно, чтобы минимизировать функцию потерь. Для задач регрессии это обычно среднее квадрата ошибки $F = \sum\limits_{X \in s_N} \sum\limits_{i=1}^{l} (x_{t+i+1} - \hat{x}_{t+i+1})^2$, где s_N - множество всех доступных X для обучения.

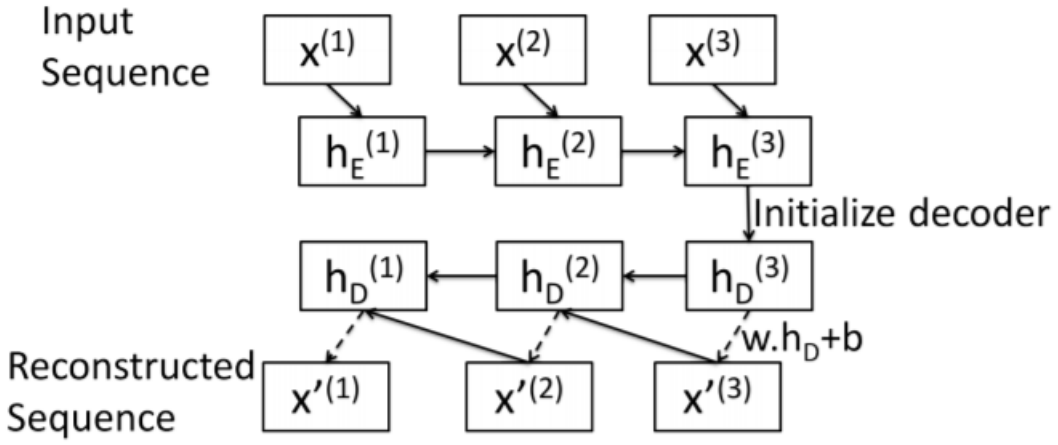


Рис. 2.2: Seq-to-seq схема из [24]. На изображении вместо нижних индексов используются верхние.

Для упрощения обучения seq-to-seq моделей существует несколько подходов, которые коротко описаны ниже:

- teacher forcing. Данный способ подразумевает использование на обучении для предсказания $\hat{x}_{t+1}$ не предыдущее предсказанное значение $\hat{x}_t$, а фактическое x_t. Таким образом увеличивается скорость сходимости модели, однако вместе с этим и растет риск переобучения. Для того, чтобы регуляризовать модель и уменьшить уровень ошибки на тесте, уровень teacher forcing постепенно уменьшается в процессе обучения: постепенно все больше последовательностей начинает обрабатываться на основе предыдущего предсказания, а не истинного значения.

- реверсирование последовательности. Этот трюк подразумевает использование на входе обратной последовательности временного ряда, например для данной задачи $X^{inv}_{t,\ldots,t+l} = \{x_{t+l}, x_{t+l-1}, \ldots, x_t\}$. Последовательность, заданная в обратном порядке, в самом начале содержит последние элементы, которые обычно сложнее предсказать, так как в общем случае нужно знать всю информацию о предыдущих состояниях. Так как проблема затухающих градиентов все еще

актуальна для рекуррентных сетей, обрабатывающих длинные последовательности, такой способ может ее решить в какой-то мере. Следует отметить, что как этот, так и предыдущий трюки носят «инженерный» характер, а не обосновываются строгой математикой.

- механизмы внимания. Это очень популярный подход, суть которого в том, что на выходе энкодера используется не последнее состояние, а взвешенная сумма всех состояний. Веса обычно генерируются другой нейронной сетью. Конкретных реализаций внимания очень много, подходящую нужно выбирать исходя из условий задачи. Также сгенерированные веса можно интерпретировать как «важность» состояний, что позволяет лучше понимать, на что опирается сеть при принятии решений и лучше визуализировать результаты.

2.3.2 Построение правдоподобия

Как было отмечено выше, нейронная сеть является интерполятором, поэтому не может хорошо восстанавливать те данные, которые ранее не видела. Именно благодаря этому можно детектировать аномалии: ошибка на новых данных будет выше, чем на обычных. Имея процедуру, которая позволяет переводить вектор ошибок e_t в рейтинг аномальности a_t, можно будет установить некоторый порог τ, при превышении которого $a_t > \tau$ принимается решение, что в x_t содержатся аномальные значения.

Самый простой рейтинг аномальности - l^1- или l^2-норма. Чтобы понять недостатки использования такого подхода, необходимо подробнее рассмотреть, что значит использование l^2-нормы. Пусть необходимо оценить вероятность принадлежности точки некоторому множеству точек, которое распределено нормально $\mathcal{N}(\mu, \Sigma)$, где $\mu \in \mathbf{R}^m$ и $\Sigma \in \mathbf{R}^{m \times m}$. Интуитивно понятно, что вероятность будет тем выше, чем ближе точка к центру масс. Зная и то, как сильно точки могут рассеиваться вокруг центра масс, можно сделать итоговое предположение о принадлежности. Логично предположить, что на нормальных данных ошибка модели должна находиться вблизи от нуля. Пусть матрица ковариаций будет единичной $\Sigma = \mathbf{I}$. Тогда расстояние Махаланобиса до данного множества в точности совпадет с l^2-нормой.

Однако такая оценка будет верной только при условии, что средние расположены вокруг нуля, а точки рассеяны так, что образуют сферу с диаметром, совершенно не учитывающим особенности данных. Чтобы исправить это, необходимо воспользоваться вычисленным средним μ и полной матрицей ковариации Σ. Информация в матрице ковариации содержит не только информацию о диаметре сферы рассеяния точек. При полной матрице данные могут быть расположены внутри эллипсоида, а

значения матрицы будут учитывать направления, а не только расстояние, принимая решение о принадлежности точки.

Итак, расстояние Махаланобиса может использоваться эффективнее, чем просто l^2-норма, если считать, что ошибки распределены нормально, тогда $a_t = (e_t - \mu)^T \Sigma^{-1} (e_t - \mu)$. Параметры распределения μ и Σ могут быть определены при помощи метода максимального правдоподобия [1].

Важно отметить, что тот факт, что распределение ошибки хорошо обученного классификатора будет иметь колоколообразную форму, кажется интуитивно понятным. Если регрессор хорошо восстанавливает процесс, то ошибка будет колебаться вокруг какого-то значения, скорее всего близкого к нулю (это может быть не точный ноль в силу смещенности обучающей выборки), а вероятность ошибиться и величина этой погрешности будут в обратной пропорциональности. Однако в таком определении нет никаких прямых указаний на то, что такое распределение будет нормальным. Распределение Лапласа и Стьюдента вполне подходят под описание. Конкретный выбор из трех распределений можно сделать лишь после анализа тяжести хвостов эмпирического распределения.

Однако для распределений выше придется отказаться от расстояния Махаланобиса. Для того, чтобы заменить его, необходимо вспомнить, что это расстояние порождается из функции распределения нормального распределения. Взяв логарифм правдоподобия и отбросив нормирующие множители

$$\log \mathcal{N}(x|\mu, \Sigma) = \log \left(\frac{1}{(2\pi)^{\frac{m}{2}} |\Sigma|^{1/2}} \exp \left(-\frac{1}{2} (x-\mu)^T \Sigma^{-1} (x-\mu) \right) \right) =$$

$$= \log \frac{1}{(2\pi)^{\frac{m}{2}} |\Sigma|^{1/2}} - \frac{1}{2} (x-\mu)^T \Sigma^{-1} (x-\mu) \sim$$

$$\sim -(x-\mu)^T \Sigma^{-1} (x-\mu),$$

можно получить метрику Махаланобиса. Следовательно, логарифм правдоподобия и рейтинг аномальности находились в обратной зависимости.

Из вышесказанного следует, что в качестве a_t можно использовать минус логарифм правдоподобия $a_t = -\log p(x_t|\theta)$, где θ - параметры распределения.

Очевидно, что параметры распределения необходимо оценивать на некоторой отложенной выборке. Пусть все данные на обучении разбиты на три множества: s_N, v_{N_1}, v_{N_2}, данные для тестирования находятся в множестве v_A. Кроме того, согласно постановке задачи, обучающий набор содержит малое число аномалий или не содержит их вообще. Нейронная сеть для задачи регрессии будет обучаться на множестве s_N,

минимизируя некоторый функционал качества F. Качество обучения и время ранней остановки подбирается на множестве v_{N_1}. Так как данная выборка в обучении напрямую не задействована, то на ней же и подбираются параметры распределения θ. Итоговые пороги τ выставляются по выборкам v_{N_2} и v_A. Следует отметить, что, вообще говоря, может быть v_A не содержит аномалий и в таком случае исследователю придется опираться на внутренние особенности задачи и здравый смысл. Кроме того, в идеале необходимо иметь выборки t_N и t_A. Это тестовые выборки, которые совсем не были бы задействованы в обучении. По первой из них возможно оценить финальное качество регрессии, по второй - качество детектирования аномалий. Разумеется, что t_A должно содержать аномалии для оценки качества.

Итак, чтобы выявлять аномалии на основе правдоподобий, необходимо:

- разбить исходную выборку на множества $s_N, v_{N_1}, v_{N_2}, t_N$ и v_A, t_A;
- разбить каждый сигнал X на окна длины l;
- обучить нейронную сеть по окну $X_{t,\dots,t+l}$ восстанавливать следующее окно $\hat{X}_{t+l+1,\dots,t+2l}$, используя все сигналы множества s_N. В качестве валидационной выборки для ранней остановки использовать v_{N_1};
- оценить финальное качество регрессии по множеству t_N;
- вычислить векторы ошибок $e_t = x_t - \hat{x}_t$ для множества v_{N_1};
- считая, что $e_t \sim \mathcal{N}(\mu, \Sigma)$ или из иного распределения, вычислить параметры распределения по v_{N_1};
- пользуясь в качестве рейтинга $a_t = (e_t - \mu)^T \Sigma^{-1} (e_t - \mu)$ для нормального распределения или $a_t = -\log p(e_t|\theta)$, где p - плотность распределения, выбрать порог τ при превышении которого принимается решение, что x_t содержит аномальные данные. Подбирать τ следует по выборкам v_{N_2}, v_A;
- оценить качество детекции по выборке t_A.

Разумеется, что алгоритм можно модифицировать при необходимости, например, исключить t_N, если число данных невелико и т.д.

2.4 Выделение информативных данных

Глубокие нейронные сети склонны к переобучению. Для того, чтобы избежать этого нежелательного явления, успешно используются разные методы регуляризации,

например, дропаут [20] или батч-нормализация [19]. Существует и иной способ избежать такой проблемы. Он заключается в сокращении числа параметров обучения. Для этого можно использовать тензорные разложения или использовать в качестве параметров разреженные матрицы.

Использование разреженных матриц позволяет добиться исключения всех неинформативных весов. Для первого слоя сети (который преобразует входные данные) разреживание данных эквивалентно исключению неинформативных признаков. Соответственно, накладывая некоторую структуру на матрицу весов, можно наложить и некоторую структуру на входные данные. Если разреживать целые блоки в матрице, то это будет соответствовать исключению целых групп входных данных. Таким образом решается сразу несколько задач: происходит регуляризация модели, ускоряется время работы на тестировании, а также остаются только наборы ценных признаков.

Для того, чтобы разреживать матрицы весов, существует множество подходов. Одним из них является использование байесовских методов разреживания. Метод ARD был описан в [26], где использовался для обучения малой нейронной сети. В дальнейшем он был детально исследован для линейных моделей таких как RVM [29] и других методов обучения с ядром.

Плюсом использования байесовских методов в глубинном обучении является то, что фактически учится не одна сеть, а целый ансамбль сетей в рамках одной модели. Ансамблирование может увеличить точность модели, увеличить устойчивость модели относительно малых изменений в данных. Байесовские методы можно использовать для регуляризации [28], а также оценки «неуверенности» модели. Ценой всех этих положительных качеств является увеличение числа параметров обучения и более трудоемкий процесс оптимизации модели. Вместо весов сети необходимо обучать распределения, и вместо каждого единичного нейрона вводится сразу несколько параметров вероятностного распределения. Например, для двухпараметрического нормального распределения каждому нейрону будет соответствовать два веса вместо одного.

Далее детально будет рассмотрен процесс группового разреживания в байесовских нейронных сетях.

2.4.1 Вариационный вывод

Пусть Y - тот сигнал, который необходимо восстановить. Тогда цель байесовкого обучения заключается в том, чтобы подобрать такие параметры w, чтобы максимизировать $p(Y|X,w)$ по данному X. В байесовском машинном обучении обычно делается какое-либо предположение об априорном распределении весов $p(w)$. После того, как в модели учитываются данные для обучения s_N, то априорное распределение преобразуется в апостериорное

$$p(w|s_N) = \frac{p(s_N|w)p(w)}{p(s_N)} = \frac{p(s_N|w)p(w)}{\int\limits_{w\in\mathbf{R}^d} p(s_N|w)p(w)dw}.$$

Такая трансформация называется байесовским выводом. Основной проблемой такой техники является необходимость вычисления многомерных интегралов в знаменателе. Обычно аналитически это можно сделать только для ограниченного числа распределений (семейство сопряженных распределений), которые не всегда соответствуют реально наблюдаемым распределениям. Например, в тематическом моделировании активно используется распределение Дирихле, которое позволяет выводить формулы, но не имеет никакого трактования в лингвистике.

Тем не менее, в случае, когда невозможно использовать аналитический вывод, можно попытаться построить некоторое приближенное решение. Одним из вариантов является использование вариационного вывода. В данном подходе апостериорное распределение $p(w|s_N)$ приближается некоторым параметрическим распределением $q_\phi(w)$. Такое распределение может быть достаточно сложным и генерироваться, например, нейронной сетью. Качество такого приближения чаще всего оценивается при помощи дивергенции Кульбака-Лейблера

$$D_{KL}(q_\phi(w) \,\|\, p(w|s_N)) = \mathbf{E}_{q_\phi(w)} \log \frac{q_\phi(w)}{p(w|s_N)}.$$

Оптимизировать параметры ϕ можно максимизируя вариационную нижнюю оценку или ELBO:

$$\mathcal{L}(\phi) = L(\phi) - D_{KL}(q_\phi(w) \,\|\, p(w)) \to \max_\phi \tag{2.3}$$

$$L(\phi) = \sum_{i=1}^{N} \mathbf{E}_{q_\phi(w)} \left[\log p(y_i|x_i,w)\right] \tag{2.4}$$

Правая часть (2.3) состоит из двух частей: логарифма правдоподобия (2.4) и KL-дивергенции, которая выступает в роли регуляризатора.

Из недостатков такого приближения можно назвать априорный зазор (prior gap в англоязычной литературе). Не вдаваясь в детали он заключается в том, что не всегда возможно одним распределением приблизить иное. Если нормальное распределение вырождается в равномерное при бесконечной дисперсии, то равномерное распределение можно приблизить нормальным без такого зазора. Однако показательным распределением не получится сколь угодно приблизить логнормальное.

2.4.2 Стохастический вариационный вывод

Однако вариационный вывод для сложных моделей тоже не всегда осуществим. Для сложных моделей математическое ожидание (2.4) нельзя вывести аналитически, что приводит к тому, что в (2.3) невозможно вычислить градиенты. Решить эту проблемы можно используя методы Монте-Карло и стохастические методы оптимизации. Для несмещенной оценки математического ожидания можно использовать среднее по мини-батчу данных $\mathcal{B}$:

$$\mathcal{L}(\phi) \simeq \mathcal{L}^{SVI}(\phi) = L_{\mathcal{B}}^{SVI}(\phi) - D_{KL}(q_\phi(w) \| p(w)) \tag{2.5}$$

$$L_{\mathcal{B}}^{SVI}(\phi) = \frac{N}{M} \sum_{i=1}^{M} \log p(\tilde{y}_i | \tilde{x}_i, w) \tag{2.6}$$

Градиенты, вычисленные напрямую в (2.6), сильно зашумлены. Чтобы снизить дисперсию в непрерывных распределениях используется трюк репепараметризации [25]. Его идея в том, чтобы заменить параметры w на некоторую дифференцируемую функцию $w = f(\phi, \epsilon)$, где $\epsilon \sim p_\epsilon(\epsilon)$ - случайный шум. Тогда

$$L_{\mathcal{B}}^{SVI}(\phi) = \frac{N}{M} \sum_{i=1}^{M} \log p_\epsilon(\tilde{y}_i | \tilde{x}_i, f(\phi, \epsilon_i)) \tag{2.7}$$

$$\nabla_\phi L_{\mathcal{B}}(\phi) \simeq \frac{N}{M} \sum_{i=1}^{M} \nabla_\phi \log p_\epsilon(\tilde{y}_i | \tilde{x}_i, f(\phi, \epsilon_i)) \tag{2.8}$$

Заменив (2.6) на (2.7) в (2.5) можно вычислять несмещенные градиенты по мини-батчу, существенно уменьшив уровень дисперсии. Для нормального распределения трюк выглядит так: если $w \sim \mathcal{N}(\mu, \sigma^2)$, то репараметризованная версия будет выглядеть $w \sim \mu + \sigma\epsilon$, где $\epsilon \sim \mathcal{N}(0, 1)$. Трюк можно выполнить не для всех распределений, к таким относится распределение Дирихле.

2.4.3 Разреживание с априорным нормальным распределением

Перед тем, как рассматривать групповое разреживание, проще рассмотреть независимое исключение весов. Переход к групповому случаю не представляет существенных сложностей. Следуя стандартной схеме ARD [26], пусть априорное распределение на веса будет нормальным с центром в нуле $p(w) = \mathcal{N}(0, \alpha^2)$, апостериорное - нормальным с произвольными параметрами $q_\phi(w) = \mathcal{N}(\mu, \sigma^2)$, где $\mu \in \mathbf{R}^m$, $\alpha, \sigma \in \mathbf{R}^{m\times m}$ и α, σ - диагональные матрицы (чтобы подчеркнуть это используются не прописные буквы). Далее для упрощения выкладок считается, что $\sigma_i = \sigma_{ii}$. KL-дивергенция в (2.5) для отдельного веса w_i будет выглядеть:

$$\begin{aligned} D_{KL}(q_\phi(w_i)\|p(w_i)) &= \mathbf{E}_{q_\phi(w)} \log \frac{q_\phi(w_i)}{p(w_i)} = \mathbf{E}_{q_\phi(w)} \left(\log q_\phi(w_i) - \log p(w_i)\right) = \\ &= \mathbf{E}_{q_\phi(w)} \left(\log \frac{1}{\sqrt{2\pi}\sigma_i} - \frac{(w_i - \mu_i)^2}{2\sigma_i^2} - \log \frac{1}{\sqrt{2\pi}\alpha_i} + \frac{w_i^2}{2\alpha_i^2} \right) = \\ &= \log \frac{1}{\sqrt{2\pi}\sigma_i} - \log \frac{1}{\sqrt{2\pi}\alpha_i} - \mathbf{E}_{q_\phi(w)} \left(\frac{(w_i - \mu_i)^2}{2\sigma_i^2} - \frac{w_i^2}{2\alpha_i^2} \right) \to \max_{\alpha,\mu,\sigma}. \end{aligned} \tag{2.9}$$

Можно исключить веса α, выразив их максимальные значения через μ и σ:

$$\frac{\partial D_{KL}}{\partial \alpha_i} = \frac{1}{\alpha_i} - \frac{1}{\alpha_i^3} \mathbf{E}_{q_\phi(w)} w_i^2 = 0 \Rightarrow \alpha_i^2 = \mathbf{E}_{q_\phi(w)} w_i^2 = \mu_i^2 + \sigma_i^2. \tag{2.10}$$

Тогда (2.9) можно упростить

$$D_{KL}(q_\phi(w_i)\|p(w_i)) = \log \frac{1}{\sqrt{2\pi}\sigma_i} - \log \frac{1}{\sqrt{2\pi(\mu_i^2 + \sigma_i^2)}} + \frac{1}{2} - \frac{1}{2} = \frac{1}{2} \log \frac{\mu_i^2 + \sigma_i^2}{\sigma_i^2}$$

и итоговая формула примет вид

$$D_{KL}(q_\phi(w)\|p(w)) = \frac{1}{2} \sum_{i=1}^{d} \log \frac{\mu_i^2 + \sigma_i^2}{\sigma_i^2} \tag{2.11}$$

Для исключения весов можно пользоваться разными способами. Можно отсекать по порогу соотношения сигнала к шуму $\frac{\mathbf{E}\theta}{\sqrt{\mathrm{Var}\theta}}$ или использовать значение α^2.

Чтобы исключать целые группы весов необходимо назначить единственное значение α для одной группы G. Группа объединяет веса по какому-то признаку, например, это целые строки матрицы или целые каналы в сверточных сетях и т.д. Тогда вид апостериорного распределения в ней сохранится, а априорное примет вид $p(w) =$

$\mathcal{N}(0, \alpha^2\mathbf{I}), \alpha \in \mathbf{R}$. KL-дивергенция сразу для всех весов в группе G:

$$\sum_{i=1}^{m} D_{KL}^{G}(q_\phi(w_i)\|p(w_i)) = \sum_{i=1}^{m}\left(\log\frac{1}{\sigma_i} - \log\frac{1}{\alpha} - \mathbf{E}_{q_\phi(w)}\left(\frac{(w_i-\mu_i)^2}{2\sigma_i^2} - \frac{w_i^2}{2\alpha^2}\right)\right) =$$
$$= -\sum_{i=1}^{m}\log\sigma_i - \frac{m}{2} + m\log\alpha + \sum_{i=1}^{m}\mathbf{E}_{q_\phi(w)}\frac{w_i^2}{2\alpha^2} =$$
$$= -\frac{1}{2}\sum_{i=1}^{m}\log\sigma_i^2 - \frac{m}{2} + \frac{m}{2}\log\frac{\sum_{i=1}^{m}(\mu_i^2+\sigma_i^2)}{m} + \frac{m}{2} = \frac{1}{2}\log\frac{\left(\sum_{i=1}^{m}(\mu_i^2+\sigma_i^2)\right)^m}{m^m\prod_{i=1}^{m}\sigma_i^2} \tag{2.12}$$

Параметры α были исключены по аналогии с (2.10) с поправкой на размер группы:

$$\frac{\partial D_{KL}^{G}}{\partial\alpha} = \frac{m}{\alpha} - \sum_{i=1}^{m}\frac{1}{\alpha^3}\mathbf{E}_{q_\phi(w)}w_i^2 = 0 \Rightarrow \alpha^2 = \frac{\mathbf{E}_{q_\phi(w)}\sum_{i=1}^{m}(\mu_i^2+\sigma_i^2)}{m},$$

где $m = |G|$ - мощность множества G. Можно заметить, что все формулы сходятся при $m = 1$.

Итоговая дивергенция равна сумме всех (2.12) для каждой группы. В отличие от (2.11) здесь не получится использовать соотношение сигнал-шум. Однако все также возможно отсекать неинформативные веса по порогу α или же абсолютным значениям KL-дивергенции в группе. KL-дивергенция будет стремиться к нулю для таких групп, так как на них не оказывает никакого влияния правдоподобие модели (2.7).

2.4.4 Разреживание с априорным распределением Лапласа

Распределение Лапласа имеет более тяжелые хвосты, чем нормальное. Наличие острого пика в нуле в теории должно собирать большее число весов вокруг моды распределения и усиливать разреживающие свойства.

Пусть априорное распределение имеет вид $p(w) = \frac{\alpha}{2}\exp(-\alpha||w||_1), \alpha \in \mathbf{R}$. Как и в предыдущем примере α - вещественное число, единственный параметр априорного распределения для группы G. Апостериорное распределение остается нормальным с произвольными параметрами $q_\phi(w) = \mathcal{N}(\mu, \sigma^2)$, где $\mu \in \mathbf{R}^m$, $\sigma \in \mathbf{R}^{m\times m}$ и σ - диагональная матрица. Далее для упрощения выкладок $\sigma_i = \sigma_{ii}$. Тогда для одной

группы:

$$\sum_{i=1}^{m} D_{KL}^{G}(q_\phi(w_i)\|p(w_i)) = \sum_{i=1}^{m} \mathbf{E}_{q_\phi(w)} \left(\log \frac{1}{\sqrt{2\pi}\sigma_i} - \frac{(w_i-\mu_i)^2}{2\sigma_i^2} - \log\frac{\alpha}{2} + \alpha|w_i| \right) =$$
$$= \sum_{i=1}^{m} \left(\log \frac{1}{\sqrt{2\pi}\sigma_i} - \log\frac{\alpha}{2} - \frac{1}{2} + \alpha \mathbf{E}_{q_\phi(w)}|w_i| \right) =$$
$$= \sum_{i=1}^{m} \left(\log \frac{1}{\sqrt{2\pi}\sigma_i} - \frac{1}{2} - \log \frac{m}{2\sum\limits_{i=1}^{m} \mathbf{E}_{q_\phi(w)}|w_i|} \right) \to \max_{\alpha,\mu,\sigma}. \tag{2.13}$$

Как и ранее в (2.10) значения α были заменены на их наибольшие значения. Для уравнения (2.13):

$$\frac{\partial D_{KL}^{G}}{\partial \alpha} = \sum_{i=1}^{m} \left(-\frac{1}{\alpha} + \mathbf{E}_{q_\phi(w)}|w_i| \right) = 0 \Rightarrow \alpha = \frac{m}{\sum\limits_{i=1}^{m} \mathbf{E}_{q_\phi(w)}|w_i|}.$$

Осталось только вычислить математическое ожидание. Его теоретически можно сэмплировать вместе с математическим ожиданием из правдоподобия, но такой подход может сильно повлиять на точность.

$$\mathbf{E}_{q_\phi(w)}|w_i| = \int\limits_{-\infty}^{\infty} |w_i| \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(\frac{(w_i-\mu_i)^2}{2\sigma_i^2} \right) dw_i = \frac{2}{\sqrt{2\pi}\sigma_i} \int\limits_{0}^{\infty} w_i \exp\left(\frac{(w_i-\mu_i)^2}{2\sigma_i^2} \right) dw_i.$$

После замены $z_i = \frac{w_i-\mu_i}{\sigma_i}$:

$$\frac{2}{\sqrt{2\pi}\sigma_i}\sigma_i \int\limits_{-\mu_i/\sigma_i}^{\infty} (\sigma_i z_i + \mu_i) \exp\frac{-z_i^2}{2} dz_i = \frac{2\sigma_i}{\sqrt{2\pi}} \int\limits_{-\mu_i/\sigma_i}^{\infty} z_i \exp\frac{-z_i^2}{2} dz_i + \frac{2\mu_i}{\sqrt{2\pi}} \int\limits_{-\mu_i/\sigma_i}^{\infty} \exp\frac{-z_i^2}{2} dz_i =$$

$$= -\frac{2\sigma_i}{\sqrt{2\pi}} \exp\frac{-z_i^2}{2}\Big|_{-\mu_i/\sigma_i}^{\infty} + \frac{2\mu_i}{\sqrt{2\pi}} \int\limits_{-\mu_i/\sigma_i}^{0} \exp\frac{-z_i^2}{2} dz_i + \frac{2\mu_i}{\sqrt{2\pi}} \int\limits_{0}^{\infty} \exp\frac{-z_i^2}{2} dz_i =$$

$$= \frac{2\sigma_i}{\sqrt{2\pi}} \exp\frac{-\mu_i^2}{2\sigma_i^2} - \frac{2\mu_i}{\sqrt{2\pi}} \int\limits_{0}^{-\mu_i/\sigma_i} \exp\frac{-z_i^2}{2} dz_i + \frac{\mu_i}{\sqrt{2\pi}} \int\limits_{-\infty}^{\infty} \exp\frac{-z_i^2}{2} dz_i =$$

$$= \frac{2\sigma_i}{\sqrt{2\pi}} \exp\frac{-\mu_i^2}{2\sigma_i^2} - 2\mu\Phi(-\mu_i/\sigma_i) + \mu_i,$$

где $\Phi(x)$ - функция распределения нормального распределения $\mathcal{N}(0,1)$. Итак,

$$\mathbf{E}_{q_\phi(w)}|w_i| = \frac{2\sigma_i}{\sqrt{2\pi}} \exp \frac{-\mu_i^2}{2\sigma_i^2} - 2\mu\Phi\left(-\frac{\mu_i}{\sigma_i}\right) + \mu_i. \tag{2.14}$$

Как и в предыдущем разделе, отсекать неинформативные группы можно по порогу. Малая KL-дивергенция говорит, что веса практически не задействованы в логарифме правдоподобия и могут быть исключены.

Глава 3

Экспериментальное исследование

3.1 Разреживание данных

Для отработки техники разреживания данных была взята стандартная задача классификации рукописных цифр на основе набора данных MNIST. Для того, чтобы проверить эффект разреживания бралась трехслойная полносвязная байесовская сеть с числом нейронов 784-300-100-10 на каждом слое. Нелинейность $ReLU(x) = \max(0, x)$, кроме последнего слоя, где использовался $softmax$. Далее нейронная сеть училась с регуляризатором (2.11). Эффект для первого слоя показан на рис. 3.1.

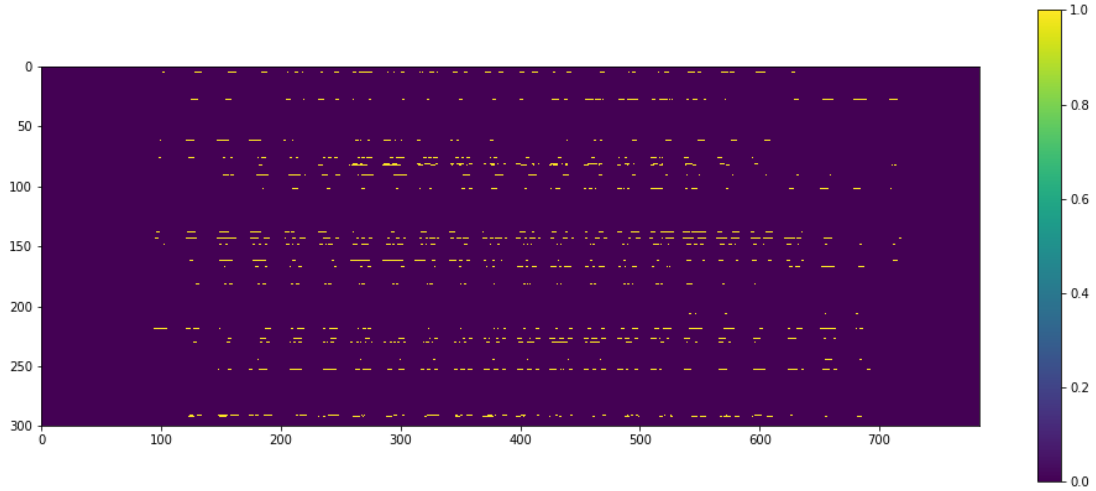


Рис. 3.1: Разреживание первого слоя нейронной сети с априорным нормальным распределением. Желтым показаны сохранившееся веса, фиолетовым - обнуленные.

Сначала обучалась сеть с регуляризатором с коэффициентом перед ним 10^{-4} с использованием метода [22] и темпом обучения 10^{-3}. Далее постепенно понижался темп обучения при повышении коэффициента до 1.

Для того, чтобы качество не проседало на тесте, веса нейронной сети - не средние μ, а сэмплируются из апостериорного распределения. Далее предсказание по ним усредняется.

Результат применения разреживания представлен в табл. 3.1.

Метод	Ошибка %	Разрежеживание по слоям %	$\frac{\lvert W\rvert}{\lvert W_{\neq 0}\rvert}$
Без разреживания	1.64		1
Pruning	1.59	92.0 – 91.0 – 74.0	12
DNS	1.99	98.2 – 98.2 – 94.5	56
SWS	1.94		23
Sparse VD	1.92	98.9 – 97.2 – 62.0	68
Предложенный ARD	2.25	99 – 99.1 – 82.9	88

Таблица 3.1: Сравнение методов разреживания.

Метод показал самое сильное разреживание, однако ценой этому стало падение точности.

Групповое разреживание в первом слое рекуррентной сети показано на рис. 3.2. Модельная задача представляла собой восстановление одного сигнала из двадцати других, где только первый и двадцатый были информативными. Длина окна $l = 5$. Использовалась модель GRU с 3 слоями и размером скрытого слоя 100 нейронов. Соответственно на входе использовалось $20 \times l = 100$ нейронов, а на выходе - l.

Модель смогла отдать приоритет информативным сигналам, однако это не так заметно на рис. 3.2. Необходимо вычислить KL-дивергенцию регуляризатора (2.13) в каждой группе. Эти значения представлены ниже:

$$0.5416, 0.0560, 0.0753, 0.0707, 0.0797, 0.0608, 0.0775, 0.0599, 0.0719, 0.0574,$$

$$0.0641, 0.0554, 0.0680, 0.0532, 0.0691, 0.0718, 0.0795, 0.0545, 0.0583, 0.5642$$

Видно, что значения крайних групп выше, что говорит об их большей важности для модели. Качество регрессии достигло $\approx 10^{-5}$ по метрике MSE для обоих подходов: с разреживанием и без.

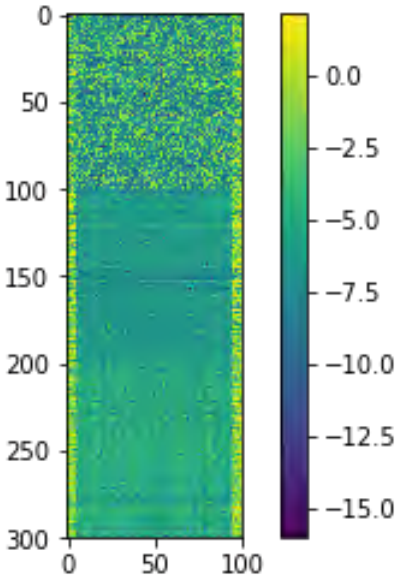


Рис. 3.2: Разреживание первого слоя рекуррентной нейронной сети с априорным распределением Лапласа. Цветом показан логарифм отношения уровня сигнала к шуму $\log \frac{|\mathbf{E}w|}{\sqrt{\mathrm{Var}w}}$ весов.

3.2 Детектирование аномалий

3.2.1 Одномерный случай

В качестве модельного примера для одномерных данных рассматривалась задача восстановления сигнала, представляющего собой сумму синусов и косинусов в разных фазах:

$$f_k(t) = \cos(\psi_{k_1} t) + \sin(\psi_{k_2} t) + \sin(\psi_{k_3} t) + \sin(\psi_{k_4} t), \tag{3.1}$$

где $\psi, t \in \mathbf{R}$, k - номер сигнала. При этом каждое ψ_{k_i} выбиралась из множества Ψ_i, содержащего три разных значения, например, $\psi_{k_1} = 2$, $\psi_{k_1} = 4$ или $\psi_{k_1} = 6$ для любого k. Полный набор значений представлен в табл. 3.2.

	ψ_{k_1}	ψ_{k_2}	ψ_{k_3}	ψ_{k_4}
Вариант 1	2	3	8	58
Вариант 2	4	5	10	57
Вариант 3	6	7	12	56

Таблица 3.2: Множество Ф.

Аномалиями считались изменения фазы в сигнале. Известно, что такой тип аномалий хуже детектируется обычными методами. Моделировалось такое изменение двумя путями:

- аномальный сигнал f_a содержит $\psi_{a_i} \notin \Phi_i$;
- в аномальном сигнале синус или косинус содержит сдвиг по фазе, например, $\sin(\psi_{k_4} t + \varepsilon)$ вместо $\sin(\psi_{k_4} t)$ в (3.1).

Для рисунков использовался сигнал с вариантом коэффициентов из табл. 3.2. В него умышленно вносились поправки коэффициентов на определенных участках. На рис. 3.3 в первой красной зоне $\phi_{k_2} = 4$ вместо 2, во второй $\phi_{k_4} = 59$, в третьей в f_k вместо члена $\sin(\psi_{k_3} t)$ используется $\sin(\psi_{k_3} t + 1)$.

Для решения использовалась трехслойная GRU-модель, соединенная обычным образом как на рис. 2.1. В качестве функции активации использовался $softplus(x) = \log(1 + e^x)$. Длина окна предсказания $l = 5$, число нейронов скрытого слоя - 25. Для обучения использовался метод Adam [22] с темпом обучения 10^{-3} в течение 300 эпох. Каждые 100 эпох темп уменьшался в 10 раз. Распределения ошибок модели показаны на рис. 3.4.

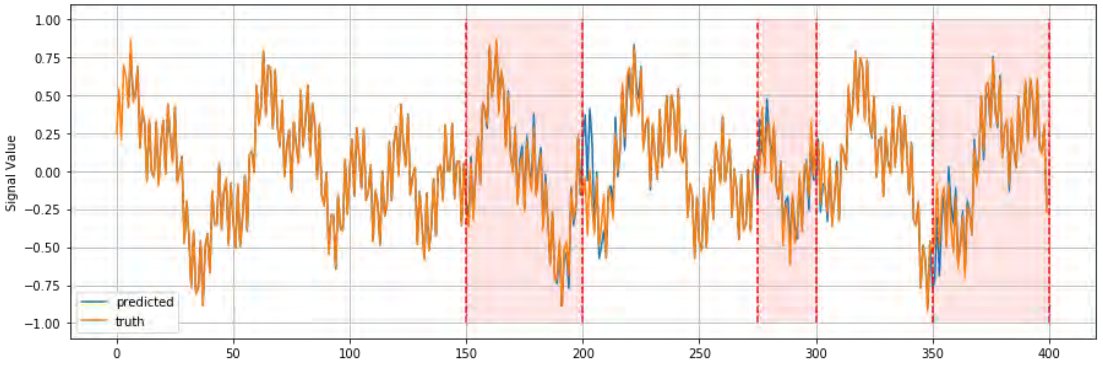


Рис. 3.3: Истинный сигнал с аномалиями (оранжевый) и восстановленный (синий). Красным отмечены зоны аномальности.

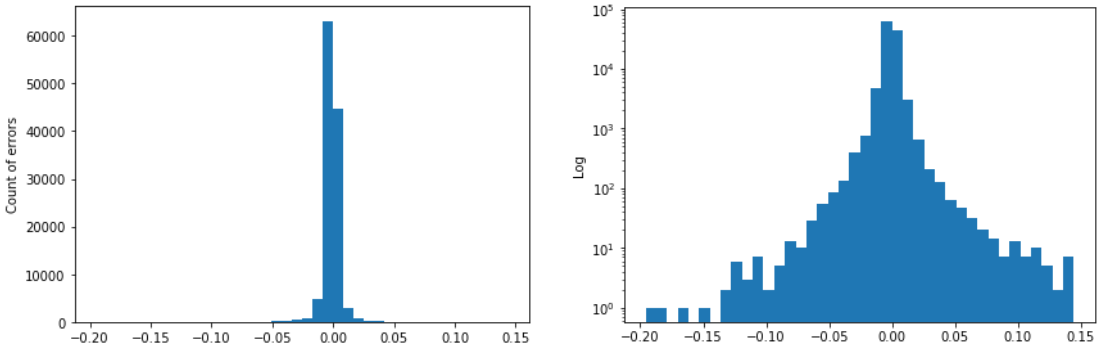


Рис. 3.4: Гистограммы ошибок. Слева - обычная гистограмма, справа - в логарифмической шкале.

На рис. 3.5 видно, что метод задетектировал все три зоны аномалий. Видно, что для повышения качества можно считать соседние зоны одной и объединять их между

собой. Кроме того, метод после аномального поведения какое-то время может относить к аномалиям нормальное поведение.

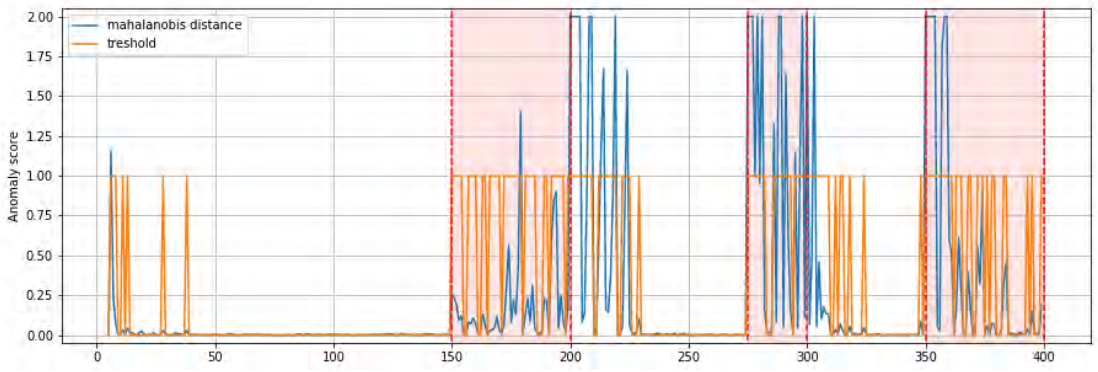


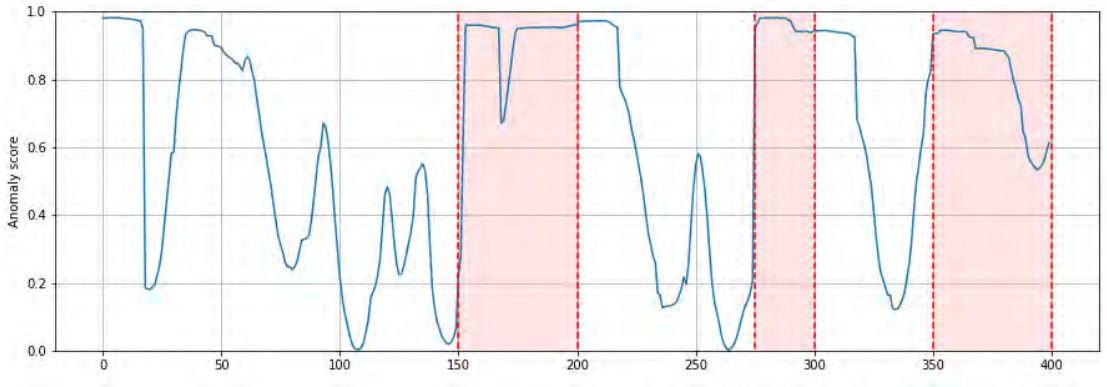
Рис. 3.5: Значения метрики Махаланобиса для данного сигнала (синим) и метка аномальности (оранжевый). Когда оранжевая линий равная единице, то метод считает, в сигнале аномалия.

Для сравнения с предложенным методом использовались методы из репозитория Numenta [23]: KNN-ICAD [12] и оконный метод Гаусса. В самом простом случае второй метод можно описать так: необходимо проходить окном по временному ряду, считая, что данные в окне распределены нормально. По статистикам из предыдущего окна рассчитывать правдоподобие следующего окна. Оба способа относятся к обучению без учителя и применимы напрямую к одномерным временным рядам. Кроме того, они больше нацелены на стриминговые данные: один длинный временной ряд, в котором ищутся аномальные шаблоны, а не множество маленьких обучающих данных. Поэтому обучающая выборка сливалась в один длинный временной ряд, в конце к которой добавлялись данные для тестирования. Предсказания для аномального временного ряда представлены на рис. 3.6.

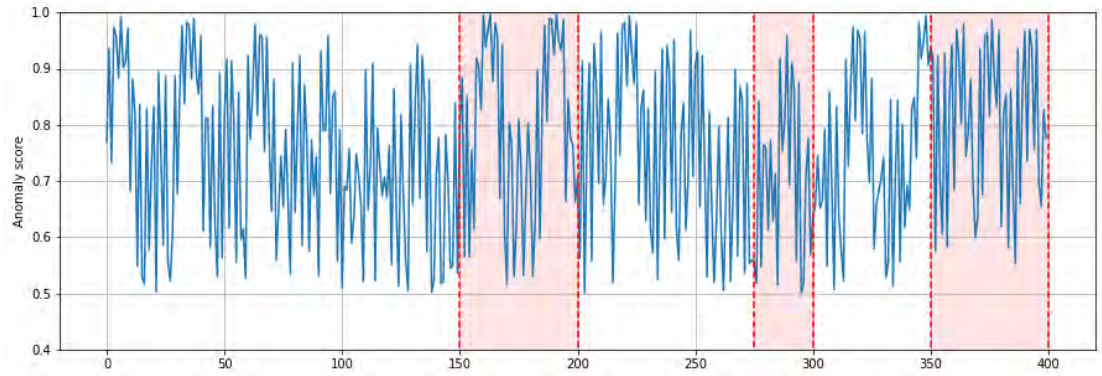
Интересно отметить, что метод Гаусса совсем не справился с прогнозированием аномалий. Это может объясняться тем, что для таких методов (Гаусса и подобных ему) нужно проводить более сложную обработку данных и извлечение признаков, чтобы детектировать подобный тип аномалий.

KNN-ICAD хорошо справился с поиском аномалий. Однако, как и предложенный метод, он имеет свойство «запаздывать»: аномалия уже прошла, но метод все еще отмечает считает, что сигнал нестандартный. Сигнал в самом начале так же некорректно отнесен к аномальному. Это может объясняться, что методу нужно некоторое время для «разогрева», чтобы уловить нормальные шаблоны в данных.

Сравнение методов представлено в табл. 3.3. Пороги подбирались для максимизации f1-меры. К предложенному в работе методу предлагается следующая эвристика: так как известно, что в данном примере аномалии группируются в окна, то имеет



а) KNN-ICAD



б) оконный метод Гаусса

Рис. 3.6: Рейтинги аномальности для описанного сигнала методами KNN-ICAD и оконным Гаусса.

смысл размечать как аномалии все $x_{t+1}, x_{t+2}, \ldots$, находящиеся между аномальными x_t и x_{t+k}, если $k < l$, где l - длина окна. Для первых значений x_t (в данном случае пятидесяти) такая эвристика не использовалась, так как первое время рекуррентная нейронная сеть чаще ошибается; ей нужно время, чтобы захватить шаблон поведения.

Метод	f1	precision	recall
Оконный Гаусса	47,6	31,2	1
KNN-ICAD	65,8	54,1	84
Предложенный	72,3	64,9	81,6
Предложенный с эвристикой	80,1	66,8	1

Таблица 3.3: Сравнение методов детекции.

3.2.2 Данные Yahoo

Вторым набором данных для валидации результатов стал набор данных Yahoo [34]. Это закрытый набор данных, который можно запросить для академических целей.

Он содержит размеченные данные об аномалиях во временных рядах. Все ряды разбиты на четыре категории: один набор реальных данных и три синтетических. В синтетических наборах есть два более сложных, в которых временной ряд декомпозируется на составляющие: линия тренда, разные сезональности и шумовая компонента. Для проверки метода использовались реальные данные и простые синтетические.

Предложенный метод использовал seq-to-seq модель с двумя LSTM слоями - один для энкодера, другой для декодера. Кроме того, декодер содержал еще и линейный слой. Число нейронов в каждом LSTM-слое - 50. Так как ряды короткие, использовался метод LBFGS [35]. Длина окна и число эпох варьировались для лучшей сходимости для каждого из рядов.

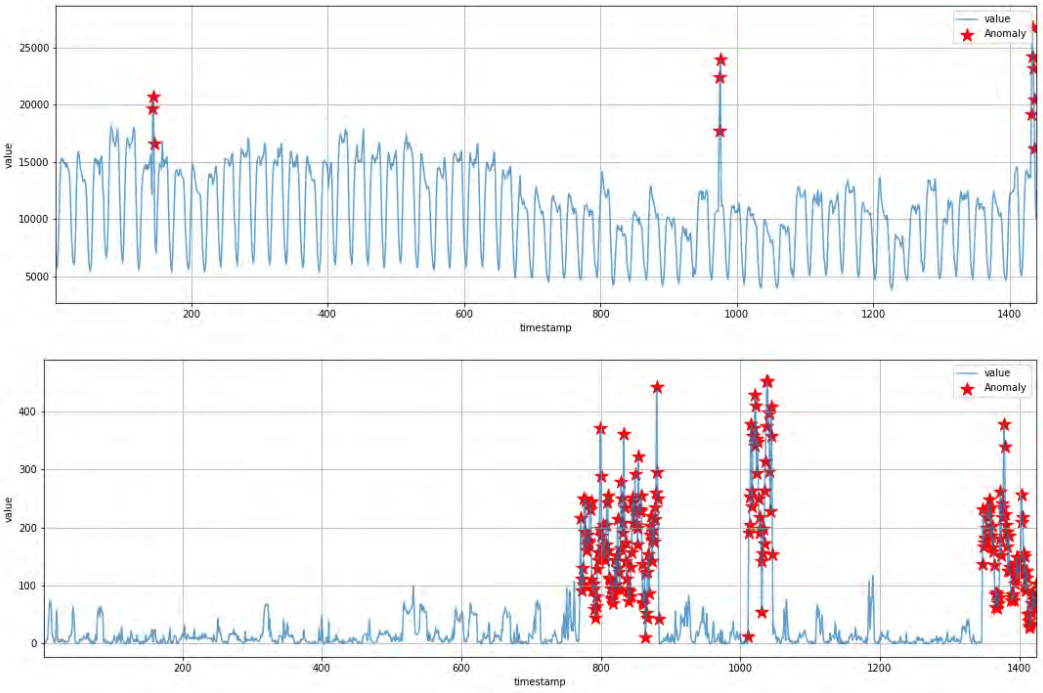
Как и для одномерного случая сравнение проходило с оконным методом Гаусса и KNN-ICAD. Удивительно, что в данных Yahoo аномалии в реальных данных оказались гораздо более простыми для обнаружения. Все они могут быть выявлены просто выставлением порогов на пиковые значения. В файле, описывающем данные, на это тоже обращают внимание и предлагают максимизировать полноту, а не точность, так как часть аномалий могла быть не отражена в разметке.

В синтетических данных сначала генерировался сигнал, а потом в случайных точках помещались аномалии. Аномалии могли встречаться даже в самом начале ряда, что может быть затруднением для методов, которые нацелены на стриминг. В таком случае аномалия не может быть выявлена, так как методу для старта нужно какое-то время, чтобы уловить шаблоны поведения. Если аномалия встречалась ранее, чем в 15% от начала временного ряда, то она не учитывалась при расчете. Результаты работы представлены в табл. 3.4-3.5. Метод Гаусса сработал здесь лучше, чем KNN-ICAD. Стоит отметить, что этот метод оказался очень чувствительным к порогам, по которым принимается решение. Для многих случаев $\tau = 1 - 10^{-6}$ и $\tau = 1 - 10^{-7}$ приводили к совсем разным значениям f1. Предложенный метод был лишен такого недостатка и обладал более контрастными рейтингами аномальности.

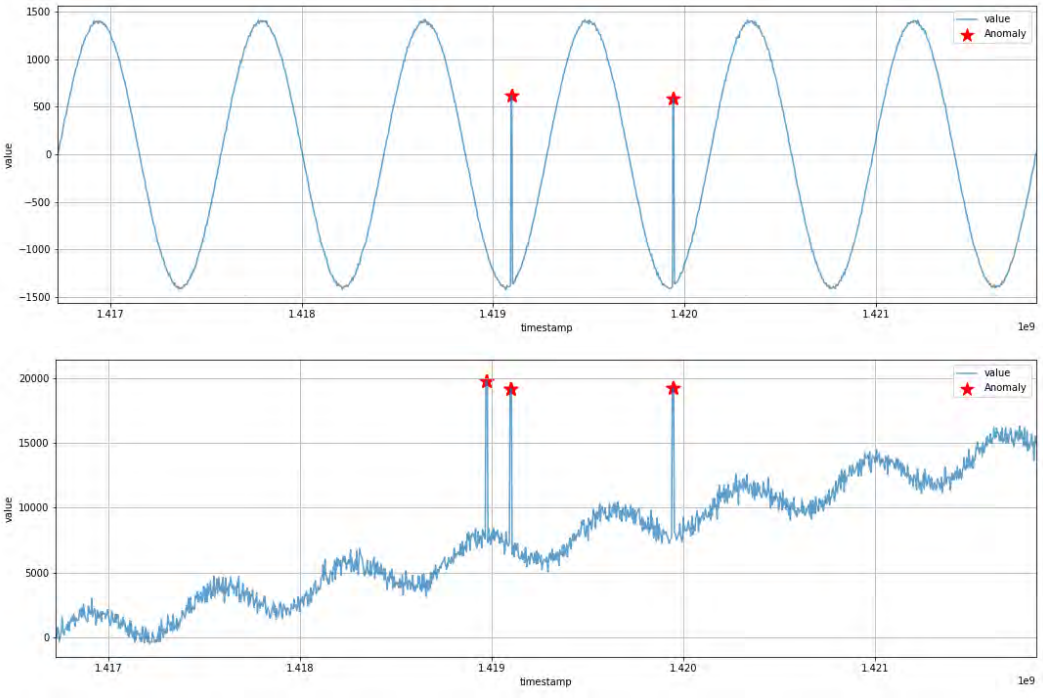
Метод	f1	precision	recall
Оконный Гаусса	72,4	64,2	80
KNN-ICAD	44,8	42,5	46,7
Предложенный	86,9	84	90.1

Таблица 3.4: Ошибки на реальных данных.

Учитывая сильную сезональность в синтетических данных, логично предположить, что методы, подобные ARIMA [36], смогут хорошо детектировать аномалии. Подобные подходы специально не рассматривались в работе, так как целью является



а) реальные данные



б) синтетические данные

Рис. 3.7: Примеры рядов из набора данных Yahoo.

построение максимально общего алгоритма, учитывающего как можно меньше локальных особенностей.

Успех примененного подхода можно объяснить тем, что он базируется на обучении с учителем. Не уменьшая число сценариев, где он может использоваться, такой

Метод	f1	precision	recall
Оконный Гаусса	33,9	30,6	60
KNN-ICAD	30	28	48,7
Предложенный	94,95	92,1	98

Таблица 3.5: Ошибки на синтетических данных.

метод может лучше учитывать особенности в данных для конкретных выборок. В данном случае метод может эффективно восстанавливать последовательности с сезональностями и линией тренда, что объясняет то, почему он легко находит отклонения.

Глава 4

Заключение

В данной работе был рассмотрен один из методов детектирования аномалий в многомерных временных рядах. Предложен метод группового разреживания для байесовских нейронных сетей на основе распределения Лапласа или нормального распределения. Использование байесовских сетей позволяет в рамках одной модели строить целый ансамбль. Для непосредственного детектирования аномалий используется логарифм правдоподобия ошибок. Экспериментальная часть подтвердила, что таким образом можно находить отклонения от нормального процесса эффективнее, чем классическими способами.

На защиту выносятся следующие положения:

- метод детектирования аномалий в многомерных временных рядах на основе правдоподобия распределения ошибки.
- методы разреживания (индивидуальный и групповой) в байесовских нейронных сетях для разных видов априорного распределения: Лапласа или нормального.
- экспериментальный анализ методов разреживания на примере полносвязной и рекуррентной сети.
- экспериментальный анализ предложенного метода детектирования аномалий на синтетических данных и данных Yahoo.

Литература

[1] *Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff.* Lstm-based encoder-decoder for multi-sensor anomaly detection. CoRR, abs/1607.00148, 2016. URL http://arxiv.org/abs/1607.00148.

[2] *Subutai Ahmad, Alexander Lavin, Scott Purdy, Zuha Aghaa* Unsupervised real-time anomaly detection for streaming data. 2017, Neurocomputing Volume 262, Pages 134-147. URL: https://doi.org/10.1016/j.neucom.2017.04.070

[3] *Pavel Filonov, Andrey Lavrentyev, Artem Vorontsov.* Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model. 2016. URL: https://arxiv.org/abs/1612.06676.

[4] *Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim.* Efficient algorithms for mining outliers from large data sets. In ACM SIGMOD Record, volume 29, pages 427–438. ACM, 2000.

[5] *Fabrizio Angiulli and Clara Pizzuti.* Fast outlier detection in high dimensional spaces. In European Conference on Principles of Data Mining and Knowledge Discovery, pages 15–27. Springer, 2002

[6] *Stephen D Bay and Mark Schwabacher.* Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 29–38. ACM, 2003.

[7] *D Danilov and A Zhigljavsky.* Principal components of time series: the «caterpillar» method. St.Petersburg: University of St. Petersburg, pages 1–307, 1997.

[8] *Lin J, Keogh E, Lonardi S, et al.* A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD workshop on research issues in data mining and knowledge discovery, pp 2–11, 2002.

[9] *E. Keogh, J. Lin and A. Fu.* HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. In Proc. of the 5th IEEE International Conference on Data Mining (ICDM 2005), pp. 226 - 233., Houston, Texas, Nov 27-30, 2005.

[10] *Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jorg Sander.* Lof: identifying densitybased local outliers. In ACM sigmod record, volume 29, pages 93–104. ACM, 2000.

[11] *Hans-Peter Kriegel, Peer Kroger, Erich Schubert, and Arthur Zimek.* Loop: local outlier probabilities. In Proceedings of the 18th ACM conference on Information and knowledge management, pages 1649–1652. ACM, 2009.

[12] *Evgeny Burnaev, Vladislav Ishimtsev.* Conformalized density- and distance-based anomaly detection in time-series data. 2016. URL https://arxiv.org/abs/1608.04585

[13] *Liu, Fei Tony, Ting, Kai Ming and Zhou, Zhi-Hua.* Isolation forest. Data Mining, 2008. ICDM 08. Eighth IEEE International Conference on.

[14] *B. Scholkopf, J.C. Platt, J.Shawe-Taylor, A.J. Smola, and R.C. Williamson.* Estimating the support of a high-dimensional distribution. Technical report, Microsoft Research, MSR-TR-99-87, 1999.

[15] *Malhotra, Pankaj, Vig, Lovekesh, Shroff, Gautam, and Agarwal, Puneet.* Long short term memory networks for anomaly detection in time series. In ESANN, 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2015.

[16] *Sepp Hochreiter and Jurgen 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.

[17] *Cho, Kyunghyun; van Merrienboer, Bart; Gulcehre, Caglar; Bahdanau, Dzmitry; Bougares, Fethi; Schwenk, Holger; Bengio, Yoshua.* Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078, 2014.

[18] *Minh-Thang Luong, Hieu Pham, Christopher D. Manning* Effective Approaches to Attention-based Neural Machine Translation. 2015. URL http://arxiv.org/abs/arXiv:1508.04025

[19] *Sergey Ioffe, Christian Szegedy* Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015. URL: https://arxiv.org/abs/1502.03167

[20] *Nitish Srivastava and Geoffrey Hinton and Alex Krizhevsky and Ilya Sutskever and Ruslan Salakhutdinov* Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 2014, volume 15, 1929-1958.

[21] *L H Chiang, E L Russell, and R D Braatz.* Fault detection and diagnosis in industrial systems. Measurement Science and Technology, 12(10):1745, 2001. URL http://stacks.iop.org/0957-0233/ 12/i=10/a=706.

[22] *Kingma, Diederik P. and Ba, Jimmy.* Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

[23] *Lavin, A and Ahmad, Subutai.* Evaluating real-time anomaly detection algorithms - the Numenta Anomaly Benchmark. CoRR, abs/1510.03336, 2015. URL http://arxiv.org/abs/1510.03336.

[24] *Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V.* Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), Advances in Neural Information Processing Systems 27, pp. 3104–3112. Curran Associates, Inc., 2014.

[25] *Kingma, Diederik P, Salimans, Tim, and Welling, Max.* Variational dropout and the local reparameterization trick. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 28, pp. 2575–2583. Curran Associates, Inc., 2015.

[26] *Neal, Radford M.* Bayesian learning for neural networks, volume 118. Springer Science & Business Media, 1996.

[27] *MacKay, David JC et al.* Bayesian nonlinear modeling for the prediction competition. ASHRAE transactions, 100 (2):1053–1062, 1994.

[28] *Gal, Yarin and Ghahramani, Zoubin.* Dropout as a bayesian approximation: Insights and applications. In Deep Learning Workshop, ICML, 2015.

[29] *Tipping, Michael E.* Sparse bayesian learning and the relevance vector machine. Journal of machine learning research, 1(Jun):211–244, 2001.

[30] *Marti, Luis, Sanchez-Pi, Nayat, Molina, Jose Manuel, and Garcia, Ana Cristina Bicharra.* Anomaly detection based on sensor data in petroleum industry applications. Sensors, 15(2):2774, 2015.

[31] *Matteson, David S and James, Nicholas A.* A nonparametric approach for multiple change point analysis of multivariate data. Journal of the American Statistical Association, 109(505): 0:334–345, 2013. URL https: //arxiv.org/abs/1306.4933v2.

[32] *Nanduri, Anvardh, Candidate, M S, and Sherry, Lance.* Anomaly detection in aircraft data using recurrent neural networks (rnn). 2016.

[33] *Ricker, N Lawrence. Tennessee Eastman Challenge Archive*, May 2013. URL http://depts.washington.edu/control/LARRY/TE/download.html.

[34] URL: http://webscope.sandbox.yahoo.com/myrequests.php

[35] *Liu, D. C.; Nocedal, J.* On the Limited Memory Method for Large Scale Optimization. Mathematical Programming B. 45 (3): 503–528. 1989. doi:10.1007/BF01589116.

[36] *Asteriou, Dimitros; Hall, Stephen G.*. ARIMA Models and the Box–Jenkins Methodology. Applied Econometrics (Second ed.). 2011. Palgrave MacMillan. pp. 265–286. ISBN 978-0-230-27182-1.