

Содержание

Аннотация	4
1 Введение	5
1.1 Описание предметной области	5
1.2 Постановка задачи	5
2 Обзор существующих методов детектирования аномалий в ICS/CPS	6
2.1 OC-SVM, Local Outlier Factor, IsolationForest	6
2.2 TranAD [6]	6
2.3 AnomalyTransformer [10]	6
2.4 RNN: LSTM models	6
2.5 Автокодировщики (AE)	7
2.6 Вариационные автокодировщики (VAE)	7
2.7 Диффузионные модели	7
2.8 Свёрточные нейронные сети (CNN)	7
2.9 Графовые нейронные сети (GNN)	7
3 Описание предлагаемого метода	8
3.1 Автокодировщик (AE)	8
3.2 Вариационный автокодировщик (VAE)	8
4 Экспериментальное исследование	9
4.1 Используемые данные	9
4.2 Функции потерь	10
4.3 Метрики	11
4.4 Обнаружение аномалий	11
4.5 Процесс и параметры обучения	12
4.6 Линейные слои в модели	13
4.7 Размеры входных последовательностей	13
4.8 Функция активации	14
4.9 Отбор признаков	14
5 Выводы и результаты	15
5.1 Восстановленные ряды	15

5.2	Анализ скрытого пространства	16
5.3	Заключение	17
	Список литературы	19

Аннотация

Задача обнаружения кибератак и различного вида дефектов является критически важной при использовании киберфизических систем (CPS) и промышленных систем управления (ICS). Такие атаки и поломки можно предсказать с помощью обнаружения аномалий в многомерных последовательностях данных о системе. В данной работе сравниваются два подхода к предсказанию промышленных кибератак: методы, построенные на автокодировщиках и вариационных автокодировщиках.

Ключевые слова

Многомерные временные ряды, обнаружение аномалий, автокодировщики (AE), вариационные автокодировщики (VAE), последовательность к последовательности (seq2seq)

1 Введение

1.1 Описание предметной области

В сложных промышленных системах, таких как: заводы, системы поддержания жизнеобеспечения, любые другие киберфизические механизмы, зачастую встаёт вопрос об опциональном обнаружении дефектов и ошибок. Например, последовательное изменение результатов работы промышленного оборудования может привести к поломке, а изменение показателей сердцебиения может указать на приближающиеся проблемы с сердцем.

Данные, описывающие различные упорядоченные последовательности показателей действий или состояний систем, представляют из себя временные ряды. Зачастую в сложных системах существует несколько показателей состояний, а следовательно данные о них описываются многомерными временными рядами.

Таким образом, задача обнаружения отклонений во временных рядах является необходимой для контроля и поддержания киберфизических систем. Это делает задачу получения оптимальной структуры модели для обнаружения аномалий важной и популярной для исследований.

1.2 Постановка задачи

В данной работе была поставлена задача обнаружения аномалий, используя трансформенные seq2seq модели кодировщик-декодировщик. Модели должны принимать на вход многомерный временной ряд и возвращать ряд таких же размеров, как и исходный. Задача состоит в том, чтобы при нормальной работе исследуемой киберфизической системы данные на входе и выходе не отличались или отличались незначительно, однако при аномальном поведении системы выход модели должен отличаться от аномального входа. Таким образом и будет обнаружена аномалия.

В качестве базовой модели была выбрана модель автокодировщика предложенная в статье о диффузионных моделях [4]. После, модель была заменена на вариационный автокодировщик. В данной работе приведено сравнение результатов предсказания аномалий по метрикам ROC_K -AUC и $F1_K$ -AUC [14]. Все эксперименты проведены на наборе данных "Secure Water Treatment (SWaT)" [5], описывающем работу системы очистки воды. Так же в работе проведён визуальный анализ реконструкции временных рядов и результатов обнаружения аномалий на них, а так же визуализация скрытых представлений, полученных с помощью t-SNE метода, для выбранных моделей.

2 Обзор существующих методов детектирования аномалий в ICS/CPS

2.1 OC-SVM, Local Outlier Factor, IsolationForest

OC-SVM [2] метод, тренируемый на нормальных данных и предсказывающий аномалии для точек, не принадлежащих обученным границам.

Local Outlier Factor [7] - метод, работающий с плотностью точек. Аномалией считаются точки с меньшей плотностью, чем у соседей.

IsolationForest [9] - модель пытается оптимально выбрать точки изоляции для разбиения объектов. Этот процесс повторяется много раз. Чем меньше ветвлений нужно для изоляции вершины, тем вероятнее, что соответствующий объект аномальный.

Данные классические методы плохо применимы для работы с многомерными временными рядами, однако для определённых задач в одномерном случае могут соревноваться с глубокими методами.

2.2 TranAD [6]

Наиболее современный метод обнаружения аномалий, восстанавливающий входящую последовательность в две фазы. А именно, входные данные сначала проходят, через один слой трансформерного кодировщика и декодировщика, после этот результат проходит через вторую фазу трансформера. В текщее время, *TranAD* показывает наилучшие результаты среди всех известных моделей.

2.3 AnomalyTransformer [10]

Модель использует блоки трансформерной модели одновременно с гауссовым ядром для построения связи между точками в входных данных. Таким образом, если связь точки с соседними маленькая, то точка считается аномальной. Данный метод показывает схожие результаты с моделью TranAD на таких видах аномалий, как: сезонные, глобальные и контекстные.

2.4 RNN: LSTM models

Используются рекуррентные нейронные сети для предсказания следующего значения временного ряда. При большом значении разницы предсказания и действительного значения

в точке определяется аномалия. [13]

2.5 Автокодировщики (AE)

Автокодировщики [12] - модели, состоящие из двух частей: кодировщик, отображающий данные в латентное пространство и декодировщик, соответственно преобразующий обратно в изначальное пространство. Разница значений на выходе и входе модели отвечает за показатель аномальности. Такие модели показывают хороший результат в обнаружении аномалий на различных временных рядах, в том числе на несинтетических данных.

2.6 Вариационные автокодировщики (VAE)

Генеративные модели, во многом схожие по структуре с автокодировщиками, но предсказывающие не точки в латентном пространстве, а их распределение. Результат предсказания аномалий вычисляется аналогично автокодировщику. [8]

2.7 Диффузионные модели

Диффузионные модели часто используются для обнаружения аномалий в изображениях, измеряя разницу между исходным изображением и изображением, прошедшим через слой зашумления и деазащумления. [3] Также могут быть использованы для предсказания аномалий в многомерных временных рядах [4]. Хорошо работают для синтетических данных, однако на данных из реального мира показывают низкие результаты.

2.8 Свёрточные нейронные сети (CNN)

Свёрточные нейронные сети обычно используются для обработки изображений. Однако многомерный временной ряд $\mathbb{R}^{T \times d}$ тоже можно обрабатывать свёртками с размером каналов d , находя зависимости между каналами. Сама сеть обычно схожа с U-net [11].

2.9 Графовые нейронные сети (GNN)

Графовые нейронные сети (GNN) [1] преобразуют входные данные в скрытое пространство, а после чего обучают структуру графа, находя зависимости между различными сенсорами (вершинами). Аномалии определяются по отклонениям графа состояния от обычной структуры графа.

3 Описание предлагаемого метода

В текущий момент, самые передовые модели, такие как TranAD [6] и AnomalyTransformer [10] используют в себе блоки трансформера, так как модели такого типа хорошо справляются с обработкой последовательностей данных. Ввиду чего, в данной работе была выбрана архитектура автокодировщика использующая в качестве кодировщика и декодировщика слои трансформерного кодировщика и декодировщика соответственно, предложенная в статье [4].

3.1 Автокодировщик (AE)

Структура модели автокодировщика состоит из двух частей: кодировщика и декодировщика соответственно.

Первым слоем кодировщика является полносвязный линейный слой, который масштабирует вход модели в несколько раз, для улучшения качества модели. После к данным добавляется позиционное скрытое представление. Далее данные обрабатываются одним слоем трансформерного кодировщика с количеством голов механизма внимания равным размерности одного объекта переданной на вход модели последовательности (количество датчиков в исследуемой системе). Так же в трансформерном кодировщике применяется обнуление 10% случайных нейронов для борьбы с переобучением модели. После чего выход кодировщика сжимается по координатам размера окна, а именно берётся среднее по второй оси тензора размера $BatchSize \times WindowSize \times FeaturesDemention$, преобразуя его в тензор размера $BatchSize \times 1 \times FeaturesDemention$.

В декодере используется трансформерный декодировщик с параметрами равными параметрам кодировщика. Далее идёт ещё один полносвязный линейный слой, преобразующий данные к исходному размеру. В случае, если входные данные отнормированы относительно минимума и максимума, то добавляется сигмоида в качестве функции активации.

3.2 Вариационный автокодировщик (VAE)

Модель вариационного автокодировщика схожа по структуре с обычным автокодировщиком. Части кодировщика и декодировщика во всём совпадают, а всё отличие заключается в структуре латентного пространства. Вместо усреднения по одной из осей используется два обучаемых линейных слоя, преобразующие выход кодировщика в пространство размеров $BatchSize \times FeaturesDemention$. Первый линейный слой отвечает за математическое ожидание μ распределения, второй за логарифм стандартного отклонения $\log \sigma$. После чего,

в декодировщик передаётся объект из нормального распределения, полученный, как $\mu + \varepsilon * \sigma$, где ε получается из стандартного нормального распределения $\mathcal{N}(0, I)$.

4 Экспериментальное исследование

4.1 Используемые данные

В данной работе все эксперименты поставлены на наборе данных "*Secure Water Treatment (SWaT)*" [5] за 2015-ый год. Данные описывают работу станции по очистке воды, на которой находится множество зависимых друг от друга систем. Вся информация представляет собой последовательности значений, полученных с 51 различного датчика в течение одиннадцати дней.

Различные аномалии в данных могут иметь различную физическую природу. Они могут быть зафиксированы в разных частях системы, могут произойти как в одном механизме, так и в нескольких одновременно. Примеры аномалий: переполнение или недостаток воды в резервуаре, изменения качества воды в определённых местах системы, недостаток или переизбыток химических веществ в воде и так далее. Однако, в данной работе типы аномалий не различаются. Считается, что модель должна отвечать в любой момент времени, имеется ли хоть в одном механизме аномалия или нет.

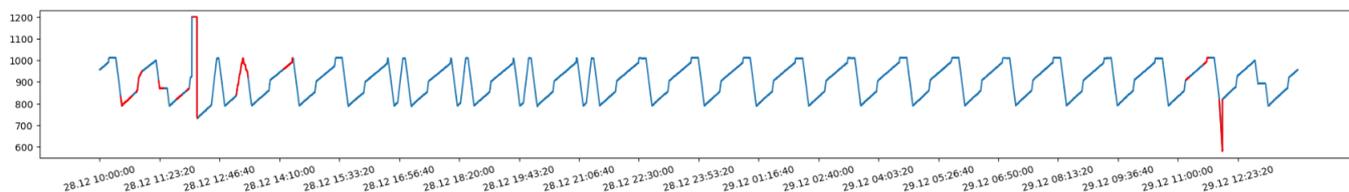


Рис. 4.1: График значений одного из датчиков. Синим и красным обозначены периоды нормальной работы системы и аномальной соответственно.

Данные о первых семи днях работы системы описывают нормальную работу всех механизмов (чистые данные без аномалий). Информация, отображающая показатели системы в эти дни, используются в качестве тренировочного набора для обучения используемых моделей. Оставшиеся четыре дня проходят с различными сценариями атак, данные о которых используются в качестве валидационной и тестовой выборки. Весь интервал времени на тестовой и валидационной выборках размечен на аномальные и нормальные периоды соответственно 4.1.

Разбиение данных с аномалиями на тестовую и валидационную выборку было осуществлено в соответствии с процентом аномальных периодов в данных. Процент аномальных

данных в валидационной выборке должен быть приближен к проценту аномалий в тестовой выборке, так как на валидации определяется пороговое значение для обнаружения аномалий. Таким образом, на валидацию было отведено 9% тестовых данных с процентом аномалий равным 11,0%. На оставшейся тестовой выборке процент аномалий равен 12,3%.

Так как исследуемые модели работают с последовательностями данных, то к ним на вход передаются окна из фиксированного количества подряд идущих значений показаний датчиков. Тогда вход модели можно задать тензором $\mathbb{R}^{d \times window}$, где d - количество датчиков, а $window$ - размер окна. Для увеличения размеров тренировочной выборки, данные нарезаются на окна с пересечением: следующее окно начинается с небольшим сдвигом от начала предыдущего окна. Таким образом из тренировочных данных размера 495000×51 , получилось увеличить количество окон с 4950 до 69270 размера $\mathbb{R}^{d \times window}$.

4.2 Функции потерь

В поставленной задаче используются модели **seq2seq**. Принимая на вход последовательность, характеризующую нормальную работу системы, модель должна возвращать эту же последовательность. Таким образом, во время обучения на нормальных данных без аномалий мы хотим штрафовать обучаемую модель за различия во входных и выходных последовательностях.

Пусть $e_t \in \mathbb{R}^d$ - вектор признаков. Предсказание модели $\hat{X} \in \mathbb{R}^{d \times T} = (e'_1, \dots, e'_T)$, где $e'_t \in \mathbb{R}^d$ - вектор предсказаний модели. В качестве функции потерь используется среднеквадратичная ошибка (MSE), которую можно записать в виде:

$$s_t = \|e_t - e'_t\|^2$$

Функция потерь для вариационного автокодировщика состоит из двух частей: ошибка восстановления, совпадает с описанной выше, и дивергенции Кульбака - Лейблера в латентном пространстве для текущего распределения $q(z|x) = \mathcal{N}(\mu, \Sigma)$ модели и стандартного нормального $\mathcal{N}(0, I)$, где x - вход модели, z - выход кодировщика. Таким образом функция потерь для вариационного автокодировщика, где M - размерность латентного пространства, имеет вид:

$$\mathcal{L} = \|e_t - e'_t\|^2 + D_{KL}(\mathcal{N}(\mu, \Sigma) \parallel \mathcal{N}(0, I)) = \|e_t - e'_t\|^2 + \frac{1}{2} \sum_{i=1}^M (\sigma_i^2 + \mu_i^2 - 1 - \ln \sigma_i^2)$$

4.3 Метрики

В качестве метрик обнаружения аномалий часто используют *поточечный подход*, который заключается в том, чтобы засчитывать правильность предсказания аномального участка, если хотя бы одна точка данного участка была классифицирована как аномальная. Однако при таком подходе модель, которая правильно предсказывает весь аномальный отрезок, будет иметь такой же результат, как и модель, которая предсказала одну точку. Это может привести к преувеличению качества модели. Поэтому в данной работе используются две другие метрики, основанные на *K-точечном подходе* [14]. Отрезок считается аномальным, если $K\%$ точек в отрезке классифицированы аномальными.

Первая метрика - $F1_K - AUC$ - для различных значений $K \in 1/100, 2/100 \dots, 99/100, 1$ считается значение F1 функции и находится площадь под графиком. $F1_K - AUC$ считается в зависимости от фиксированного порога значения функции ошибки δ . Параметр δ подбирается на валидационной выборке. Однако на практике, эта метрика очень не стабильна, потому что масштабы ошибок на валидационной и тестовой выборке могут отличаться. В виду чего используется вторая метрика - $ROC_k - AUC$. Здесь также используется идея фиксирования аномального отрезка при $K\%$ аномальных точек, для определённого K , но метрика считается не зависимо от обоих пороговых значений K и δ . Можно представить $ROC_k - AUC$ как трёхмерное обобщение обычной $ROC - AUC$ с добавлением оси со значением K .

4.4 Обнаружение аномалий

Для обнаружения аномалий используется поэлементная среднеквадратичная ошибка между переданной последовательностью данных и выходом модели. Для каждого объекта ошибка усредняется по всем признакам (показателям датчиков). Таким образом для каждого элемента последовательности считается показатель аномальности $r = \frac{1}{d} \sum_{i=1}^d (x_i - f(x_i))^2$, где d - число признаков, x_i - признак объекта, а $f(x_i)$ - выход модели. После значение r сравнивается с пороговым значением. Если значение показателя r выше порога, то точка считается аномальной.

Порог для каждой модели определяется на валидационной выборке. Для всех точек на валидации считаются показатели аномальности. Среди них выбирается минимальное и максимальное значение. Отрезок между этими двумя значениями разбивается на 50 равных частей, границы которых принимаются за пороги. Для каждого порога считается $F1_K - AUC$, после чего выбирается порог с максимальным значением метрики.

4.5 Процесс и параметры обучения

При обучении обеих моделей крайне важен набор гиперпараметров. Часто неправильный выбор начальных параметров или расписания темпов обучения приводил к взрыву градиентов, а следовательно и к наличию пропусков на выходе модели. Поэтому проводились эксперименты по подбору типов оптимизаторов обучения, начальных параметров и расписания темпов обучения модели, а так же штрафов на веса модели.

В ходе экспериментов использовались оптимизаторы SGD и Adam. Второй показал себя лучше с точки зрения стабильности обучения и качества восстановленных последовательностей.

В ходе подбора темпов обучения (lr) были использованы константные значения, линейные расписания, однако такие темпы приводили к взрыву градиентов или к плохим результатам модели. Лучшие результаты показали расписания с разогревом темпов обучения: первые несколько эпох темп растёт, а после падает 4.2. К таким расписаниям относится линейное расписание с разогревом и OneCycleLR. Далее во всех экспериментах использовалось линейное расписание с разогревом от трёх до пяти эпох.

Штраф на веса нейронов не улучшил результаты модели и не добавил ей устойчивости к взрыву градиентов. Поэтому далее штраф на веса не использовался.

Таким образом, во всех экспериментах далее использовался оптимизатор Adam, с линейным разогревом темпа обучения в пике от 10^{-4} до 10^{-3} .

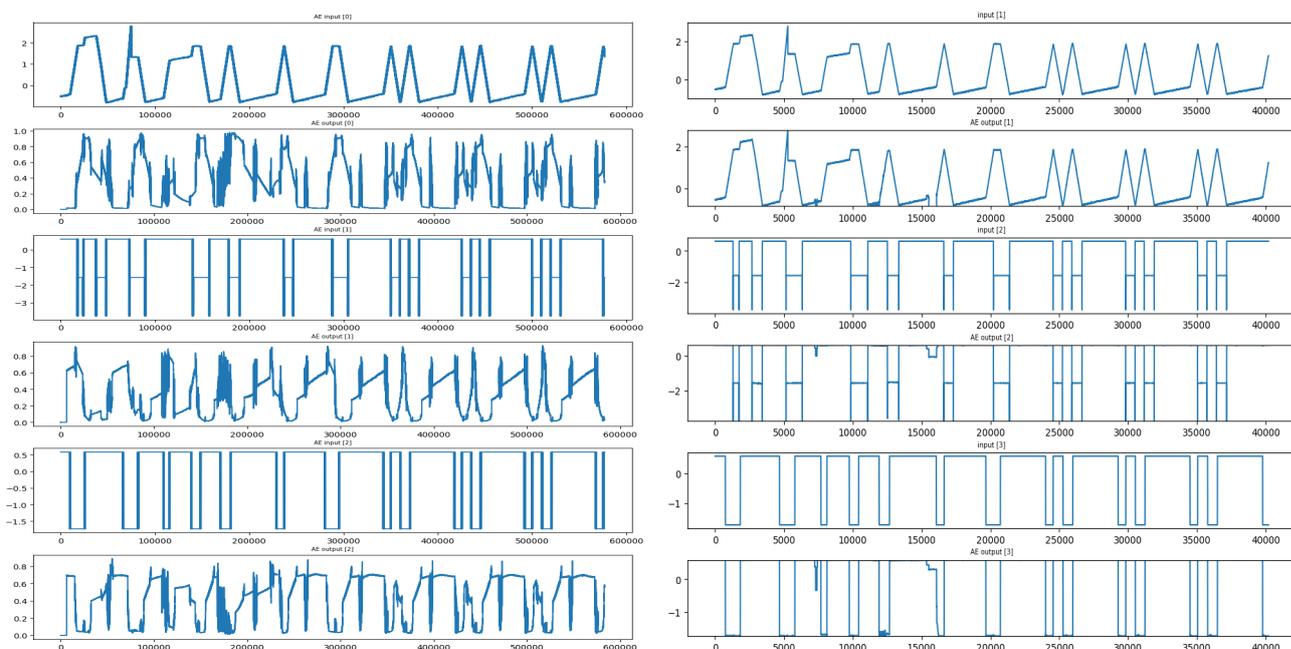


Рис. 4.2: Некоторые параметры валидационной выборки (нечётные графики) и выход АЕ на этой же выборке (чётные графики). Слева результат модели с линейным расписанием обучения, справа с разогревом линейного темпа обучения.

4.6 Линейные слои в модели

Для улучшения качества модели перед кодировщиком был добавлен линейный слой размера $features \times (features * scale)$, увеличивающий пространство параметров в $scale$ раз. Увеличение размера линейного слоя значительно увеличивает количество нейронов в модели, что увеличивает вес модели и уменьшает скорость обучения. Поэтому максимальный исследуемый параметр масштабирования был равен 16. По результатам экспериментов с вариационным автокодировщиком без активации 4.1 и с ней 4.2 можно сделать вывод, что увеличения размеров линейного слоя положительно влияет на качество модели.

Таблица 4.1: результаты VAE на тестовой выборке без функции активации

scale	F1 _к – AUC	ROC _к – AUC
1	0.2985	0.8784
16	0.3919	0.8859

Таблица 4.2: результаты VAE на тестовой выборке с функцией активации

scale	F1 _к – AUC	ROC _к – AUC
1	0.2307	0.8682
8	0.2590	0.8701
16	0.2681	0.8705

4.7 Размеры входных последовательностей

На вход моделям передаются последовательности данных. В зависимости от размера таких последовательностей меняются и размерности самой модели. Следовательно, размеры окон, на которые нарезаются данные, влияют и на качество модели. В данной работе были проведены эксперименты с несколькими значениями размеров окон 4.3.

Таблица 4.3: результаты VAE без функции активации в зависимости от размера окна данных на тестовой выборке

window size	F1 _к – AUC	ROC _к – AUC
70	0.3290	0.8789
85	0.2960	0.8712
100	0.3919	0.8859

Таким образом 100 - наилучший размер входной последовательности.

4.8 Функция активации

В статье с предложенной архитектурой автокодировщика [4] в качестве активации использовалась сигмоида $\sigma(x) = \frac{1}{1+e^{-x}}$. Следовательно выход модели всегда лежал в интервале $[0, 1]$. Для этого данные масштабировались относительно минимума и максимума тренировочной выборки. Подход с сигмойдой в качестве активации ограничивает выход модели, что может помочь при обнаружении аномалий точек. Однако, на рассматриваемых данных, некоторые не аномальные точки в валидационной и тестовой выборках превосходили по абсолютным значениям точки из тренировочной выборки (верхний график 4.3). Из-за чего модель не идеально восстанавливает структуру показаний датчиков (нижний график 4.3), а следовательно, может ложно предсказать аномалию, там, где её нет.

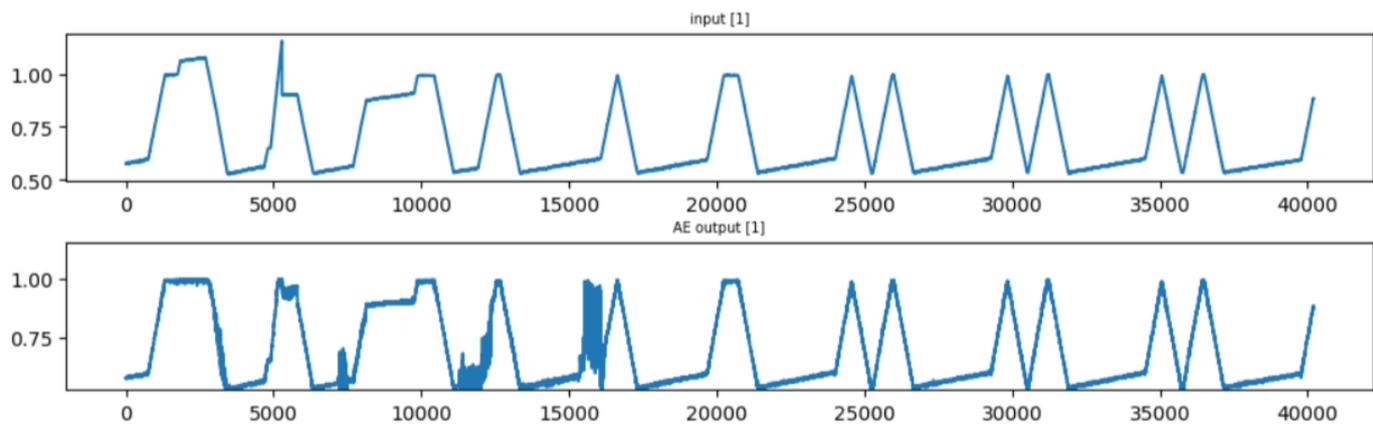


Рис. 4.3: Чистые валидационные данные с датчика (выше) и восстановленная моделью АЕ с активацией последовательность (ниже)

Ввиду этой проблемы были проведены эксперименты без функции активации после последнего линейного слоя. Оказалось, что на рассматриваемых метриках, модель без функции активации показывает лучшее качество 4.4.

Таблица 4.4: результаты VAE в зависимости от наличия функции активации

activation	F1 _к – AUC	ROC _к – AUC
-	0.2681	0.8700
+	0.3919	0.8859

4.9 Отбор признаков

В исходных данных описываются данные о 51 датчике. Однако не в каждом процессе системы за исследуемый аномальный период происходили сбои. В данной работе был проведён эксперимент, где на вход модели подавались данные, только о тех 19 признаках,

в которых когда-либо наблюдались аномалии. Результаты показали, что такие модели показывают результат на значительно выше, чем модели на полном пространстве признаков 4.5.

Таблица 4.5: Результаты моделей в зависимости от количества признаков

model	feature number	$F1_k - AUC$	$ROC_k - AUC$
AE	51	0.6907	0.8825
VAE	51	0.3919	0.8859
AE	19	0.8793	0.9103
VAE	19	0.8807	0.9103

Однако идея использования части параметров, на которых наблюдались аномалии, содержит утечку данных. А именно, изначально известно, что аномалия может произойти на каждом из датчиков. Но такая модель не фиксирует аномалию, происходящую в других частях системы или более сложные аномалии, зависящие от нескольких показателей одновременно. Поэтому такой метод вряд ли можно применить на практике.

5 Выводы и результаты

5.1 Восстановленные ряды

На графиках 5.1 расположены входные данные по нескольким датчикам с обозначенными красным аномальными участками и выходы моделей AE и VAE. Можно заметить, что обе модели хорошо восстанавливают структуру данных, однако последовательности полученные с помощью VAE чуть более шумные, а именно частота колебаний графика выше.

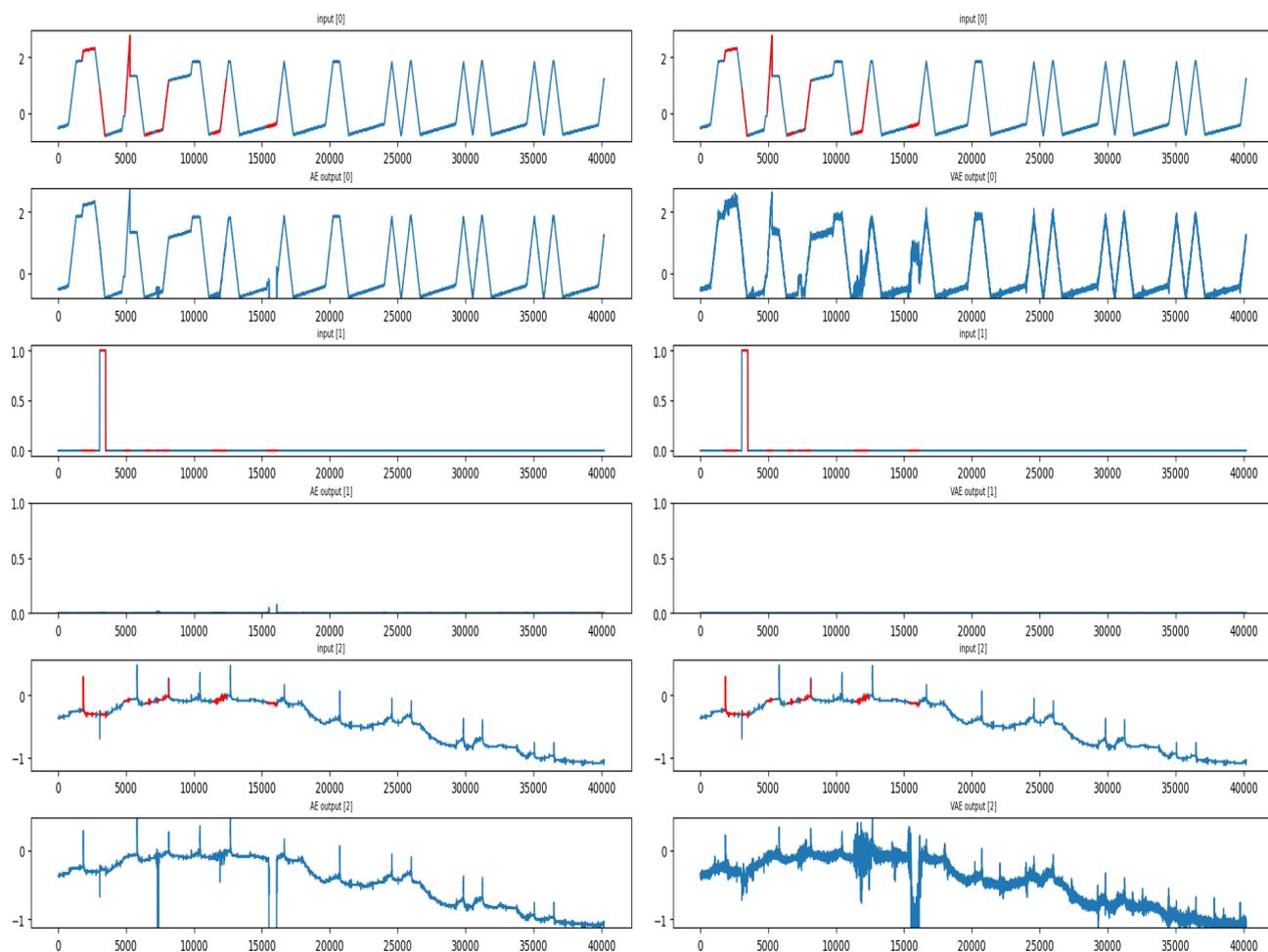


Рис. 5.1: Результаты работы AE и VAE при лучшем наборе параметров

5.2 Анализ скрытого пространства

Для анализа и визуализации скрытого пространства полученных моделей был выбран метод t-SNE. T-SNE отображает многомерные данные в меньшие размерности. На графиках [5.2](#) [5.3](#) изображено отображение скрытых пространств в двумерное. Жёлтым обозначены точки, соответствующие аномальному поведению системы, фиолетовыми - нормальному состоянию.

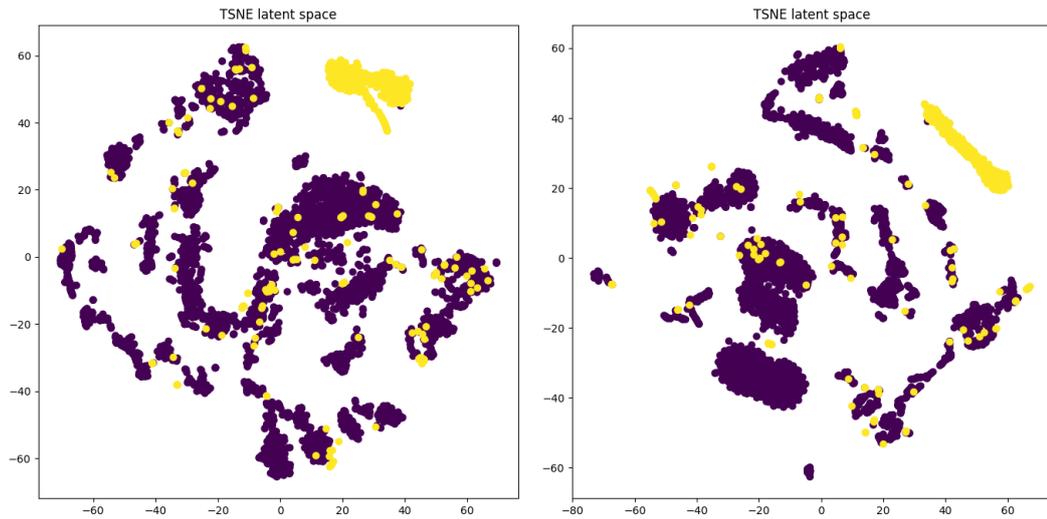


Рис. 5.2: t-SNE отображение скрытого пространства АЕ. На левом графике модель принимает на вход 51 признак, на правом - 19

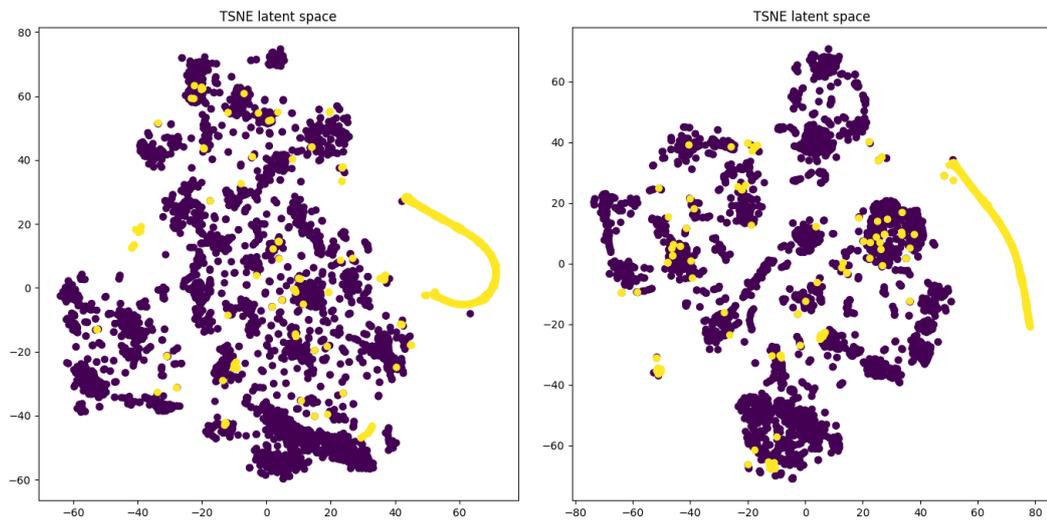


Рис. 5.3: t-SNE отображение скрытого пространства VAE. На левом графике результат модели, использующей масштабирование входных данных в 16 раз, на правом - без масштабирования

5.3 Заключение

В данной работе были реализованы модели автокодировщика и вариационного автокодировщика для обнаружения аномалий. По основной метрике ROC_K -AUC VAE показал результат выше, чем АЕ. Однако на дополнительной и менее стабильной метрике $F1_K$ -AUC АЕ наоборот лучше, чем VAE. Обе модели показали результат выше, чем диффузионная модель из статьи, где была предложена архитектура базовой модели АЕ для данной работы [5.1](#).

Был создан репозиторий с реализацией всех описанных выше моделей, функций для их обучения и средств для обработки и визуализации входных данных. Также был проведён анализ результатов прогноза аномалий и полученных скрытых представлений.

Таблица 5.1: Финальные результаты моделей

model	$F1_{\kappa} - AUC$	$ROC_{\kappa} - AUC$
AE	0.6907	0.8825
VAE	0.3919	0.8859
Diffusion [4]	0.505	0.767

Список литературы

- [1] Bryan Hooi Ailin Deng. “Graph Neural Network-Based Anomaly Detection in Multivariate Time Series”. B: *AAAI Conference on Artificial Intelligence (AAAI)*. 2021.
- [2] John Shawe-Taylor Alexander J. Smola Robert C. Williamson Bernhard Scholkopf John C. Platt. “Estimating Support of a High-Dimensional Distribution”. B: *Neural Computation* 13(7):1443-1471. 2001.
- [3] Wyatt J. Leach A. Schmon S.M. Willcocks C.G. “Anomaly detection with denoising diffusion probabilistic models using simplex noise”. B: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022*. 2022.
- [4] Florin Brad Ioana Pintilie Andrei Manolache. “Time Series Anomaly Detection using Diffusion-based Models”. B: *arXiv:2311.01452v1 [cs.LG]* (2023).
- [5] Singapore University of Technology iTrust Centre for Research in Cyber Security и Design. B: 22.01.2024.
- [6] Shreshth Tuli Giuliano Casale Nicholas R. Jennings. “TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data”. B: *arXiv:2201.07284v6* (2022).
- [7] Raymond T. Ng Joerg Sander Peer Kröger. “LOF: Identifying Density-Based Local Outliers”. B: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. 2000.
- [8] T.Carraro F.Aioll L.Bergamin M.Polato. “Novel Applications for VAE-based Anomaly Detection Systems”. B: *arXiv:2204.12577v1 [cs.LG]* (2023).
- [9] Ting Liu F.T. “Isolation forest”. B: *Eighth IEEE International Conference on Data Mining*. 2008.
- [10] Jiehui Xu Haixu Wu Jianmin Wang Mingsheng Long. “Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy”. B: *arXiv:2110.02642* (2022).
- [11] Roy Keyes Tailai Wen. “Time Series Anomaly Detection Using Convolutional Neural Networks and Transfer Learning”. B: *arXiv:1905.13628v1 [cs.LG]* (2019).
- [12] Theodoros Tziolas Konstantinos Papageorgiou Theodosios Theodosiou. “Autoencoders for Anomaly Detection in an Industrial Multivariate Time Series Dataset”. B: *ITISE2022: 8th International conference on Time Series and Forecasting*. 2022.
- [13] Ali H. Mirza Tolga Ergen и Suleyman S. Kozat. “Unsupervised and Semi-supervised Anomaly Detection with LSTM Neural Networks”. B: *arXiv preprint, arXiv:1710.09207* (25 Oct 2017).

- [14] Siwon Kim Kukjin Choi Hyun-Soo Choi Byunghan Lee Sungroh Yoon. “Towards a Rigorous Evaluation of Time-series Anomaly Detection”. B: *arXiv:2109.05257v2 [cs.LG]* (2022).