ВЫЯВЛЕНИЕ ОДИНОЧНЫХ АНОМАЛИЙ В ДАННЫХ ОБ ЭНЕРГОПОТРЕБЛЕНИИ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ БЕЗ УЧИТЕЛЯ

Марьясин О. Ю.[1], **Тихомиров Л. И.**[2]

(*Ярославский государственный технический университет, Ярославль*)

Описаны исследования по выявлению одиночных аномалий в данных об энергопотреблении на примере двух разных наборов данных. Рассмотрены способы построения типовых шаблонов энергопотребления и представлен авторский способ построения типового суточного профиля энергопотребления. Для проведения численных экспериментов авторами был выбран 21 метод машинного обучения без учителя, подходящий для выявления одиночных аномалий. По результатам численных экспериментов были отмечены методы, наиболее удачно справившиеся с задачей выявления одиночных аномалий. Особое внимание в работе уделялось методам, не требующим дополнительных параметров, и современным перспективным методам на базе искусственных нейронных сетей. Лучшими алгоритмами по результатам испытаний оказались статистические алгоритмы, основанные на построении гистограмм. Одной из главных проблем, затронутых в работе, является проблема настройки параметра contamination для каждого рассмотренного алгоритма. Одним из решений данной проблемы являются использование пороговых алгоритмов. Показано, что если исходный алгоритм выявляет аномалии недостаточно хорошо (параметр contamination не настроен), то применение пороговых алгоритмов может существенно повысить точность обнаружения аномалий. Отмечены пороговые алгоритмы, использование которых для задач анализа аномалий в данных об энергопотреблении чаще других обеспечивает повышение точности. Применять пороговые алгоритмы можно как к результатам работы отдельных алгоритмов выявления аномалий, так и к результатам работы ансамблей алгоритмов, полученных с использованием различных стратегий комбинирования.

Ключевые слова: обнаружение одиночных аномалий, типовой суточный профиль энергопотребления, машинное обучение, пороговый алгоритм, ансамбли алгоритмов.

---

[1] *Олег Юрьевич Марьясин, к.т.н., доцент* (*maryasin2003@list.ru*).

[2] *Леонид Игоревич Тихомиров, аспирант* (*lenusscik@yandex.ru*).

1. Введение

В настоящее время сетевые организации должны устанавливать приборы учета (ПУ) электроэнергии, обеспечивающие возможность присоединения к интеллектуальной системе учета электрической энергии (мощности) [2]. Такие системы обеспечивают непрерывную передачу данных от потребителей к энергосбытовым компаниям, из-за чего исчезает необходимость ручного снятия показаний с ПУ и появляются новые возможности, такие как возможность более детально следить за объёмом потребления электроэнергии, быстрее обнаруживать сбои и предоставлять потребителям больше доступной информации об их энергопотреблении.

Интеллектуальная система учета позволяет реализовать углубленный анализ данных об энергопотреблении. Такой анализ может использоваться для выявления показаний, сильно отклоняющихся от нормы, т.е. аномалий. Появление аномалий может быть связано как с перегрузками и внеплановыми отключениями электрической сети, так и с кражами электроэнергии из-за незаконных подключений к электросети, негативных воздействий на ПУ электроэнергии, с использованием энергоресурсов предприятий или организаций в личных целях. Выявление аномалий может стать основанием для более пристального наблюдения за профилем энергопотребления как со стороны энергоснабжающей организации, так и со стороны энергопотребителя.

Для анализа аномалий необходимо сначала определить, что будет считаться аномалией. Согласно [5] аномалия представляет собой данные, которые не соответствует ожидаемому нормальному поведению. Следовательно, для обнаружения аномалий необходимо определить область, представляющую нормальное поведение, а любые данные, не принадлежащие этой области, объявить аномалиями. Но есть целый ряд факторов, усложняющих применение такого подхода [5]. Отметим только те факторы, которые можно отнести к анализу аномалий в данных об энергопотреблении:

– определить область, охватывающую все возможные варианты нормального поведения, очень сложно. Кроме того, граница между нормальным и аномальным поведением часто бывает размытой. Таким образом, аномальное наблюдение, лежащее близко к границе, может быть нормальным, и наоборот;

– злоумышленники часто стараются сделать так, чтобы аномальные наблюдения казались нормальными, тем самым усложняя задачу определения нормального поведения;

– доступность размеченных данных для обучения/проверки моделей, используемых при обнаружении аномалий, обычно является серьезной проблемой;

– часто данные содержат шум, который может быть похожим на аномалии.

В данной работе под аномалией мы будем понимать значительное отклонение энергопотребления от типовых значений энергопотребления или типового шаблона энергопотребления (ТШЭ), полученного для конкретного энергопотребителя и календарного периода времени. К сожалению, в настоящее время нет какой-либо общепринятой методики для построения ТШЭ. Некоторые известные из литературы способы представлены далее. Построение ТШЭ потребителей электроэнергии является начальным этапом при решении задачи выявления аномалий.

Все аномалии данных можно разделить на три группы [17]:

– одиночные (точечные) аномалии (выбросы) данных. Это одиночные данные, которые резко отличаются от других данных и выходят за допустимые пределы. Одиночные аномалии в данных об энергопотреблении могут быть связаны, например, с кратковременными перегрузками или отключениями от электросети у энергопотребителей;

– коллективные аномалии. Коллективные аномалии могут быть охарактеризованы как последовательности данных, имеющих похожий характер, но которые выходят за допустимые пределы. Коллективные аномалии могут быть связаны с длительными отключениями от электросети, с кражами электроэнергии, использованием энергоресурсов предприятий или организаций

в личных целях и другими факторами;

– контекстные аномалии. Контекстные аномалии могут быть представлены как последовательности данных, которые резко отличаются от других данных в том же контексте, но находятся в допустимых пределах. Контекстные аномалии могут быть связаны, например, с ошибками, возникающими при передаче данных от приборов учета.

В данной работе рассматривается применение различных методов машинного (и глубокого машинного обучения) для анализа только одиночных аномалий. При этом для выявления аномалий применяются исключительно методы машинного обучения, использующие обучение без учителя. Это позволяет избежать трудоемкого этапа разметки наборов данных, на основе которых производится обучение и проверка моделей машинного обучения. Основное внимание в работе отводится вопросу отбора наиболее результативных и перспективных методов выявления одиночных аномалий и способам повышения их возможностей.

2. Обзор литературы

2.1. Методы построения ТШЭ

Если для представления ТШЭ использовать «сырые» данные, а затем для анализа закономерностей применять методы кластеризации, например, метод k-средних, то при большом количестве признаков могут возникнуть проблемы, связанные с большой трудоемкостью расчетов [30]. Поэтому исследователи стремятся использовать для представления ТШЭ не «сырые» данные, а ограниченное число наиболее важных характеристик временного ряда данных об энергопотреблении. В [25] авторы для представления недельных шаблонов поведения временных последовательностей определили такие статистические показатели: среднее, стандартное отклонение, коэффициент асимметрии (skewness), коэффициент эксцесса (kurtosis), энергетическая характеристика, период сезонной составляющей временного ряда. Энергетическая характеристика рассчитывалась как сумма квадратов величин дискретных компонентов быстрого преобразова-

ния Фурье сигнала, деленная на длину окна нормализации.

В [11] авторы поделили сутки на четыре временных интервала: ночь, утро, день и вечер. Для каждого энергопотребителя вычислялось среднее значение мощности P_i в каждый период времени $i = 1, 2, 3, 4$ и соответствующее стандартное отклонение σ_i. Кроме того, вычислялись: среднесуточная мощность для каждого энергопотребителя P_d, средние мощности для летнего P_{Si} и зимнего P_{Wi} сезонов в каждый период времени, средние мощности в выходные P_{WEi} и будние дни P_{WDi} в каждый период времени за все годы данных. На основании этих показателей для каждого энергопотребителя были определены следующие атрибуты:

– относительная средняя мощность в каждый период времени в течение всего года $P_{Ri} = P_i/P_d,\ i = 1, 2, 3, 4$;

– среднее относительное стандартное отклонение за год как среднее значение от σ_i/P_i по всем периодам;

– сезонный показатель как сумма отношений $|P_{Wi} - P_{Si}|\,/P_i$ по всем периодам;

– оценка разницы между выходными и будними днями как сумма отношений $|P_{WDi} - P_{WEi}|\,/P_i$ по всем периодам.

Полученные атрибуты использовались в [11] для описания годового шаблона поведения энергопотребителей.

При построении ТШЭ важным моментом является задание периода времени, для которого определяются показатели. Это может быть неделя или год, как в рассмотренных ранее примерах, или сутки и месяц. Соответственно, могут быть построены суточные, недельные, месячные и годовые ТШЭ. Важность того или иного шаблона зависит от конкретного энергопотребителя, его профиля деятельности, климатических условий местности, где находится энергопотребитель, и других факторов. Для отдельных энергопотребителей может быть целесообразно использовать при анализе сразу несколько различных ТШЭ.

Данные энергопотребления для конкретного энергопотребителя, полученные в течение суток с периодом в один час, мы будет называть суточным профилем энергопотребления (СПЭ). То-

гда шаблон, представляющий профиль энергопотребления в течение суток и составленный на основе данных, собранных за различный период времени, мы будет называть типовым суточным профилем энергопотребления (ТСПЭ). Построение ТСПЭ дает важную информацию для выявления общих или специфических характеристик функционирования энергетических объектов [15].

В [1] для построения ТСПЭ был использован способ, основанный на идеях, представленных в [11] и [19]. В соответствии с данным способом календарные сутки разбиваются на определенное число временных интервалов. Количество и ширина интервалов должны быть связаны с графиком работы энергопотребителя. В [1] ТСПЭ строился для зданий Ярославского государственного технического университета (ЯГТУ). Как и в [11], сутки были разделены на четыре временных интервала: 00:00–07:00, 07:00–12:00, 12:00–18:00, 18:00–24:00. Для каждого интервала определялось среднее значение и среднеквадратическое отклонение энергопотребления. Кроме того, для каждых суток определялось минимальное и максимальное значение энергопотребления. Определение показателей за одни сутки показано на рис. 1.

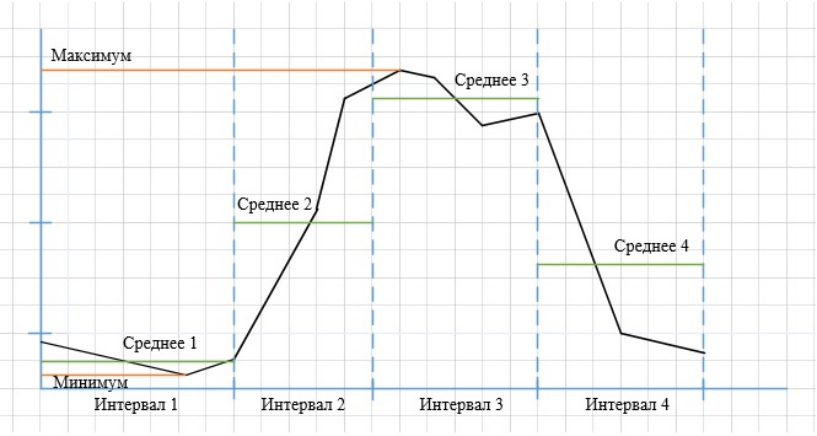

Рис. 1. Разбиение СПЭ на интервалы и определение средних значений

Далее рассчитывались подобные показатели для каждого интервала за весь месяц. Данные параметры были использованы для построения ТСПЭ энергопотребителя за каждый месяц года. На рис. 2 показана диаграмма, представляющая ТСПЭ для общежития номер 1 ЯГТУ за ноябрь 2021 года. Поскольку ТСПЭ строятся и хранятся для каждого месяца каждого года, то этим учитывается сезонность энергопотребления. По мнению авторов это дает больше информации, чем построение ТСПЭ за весь год.

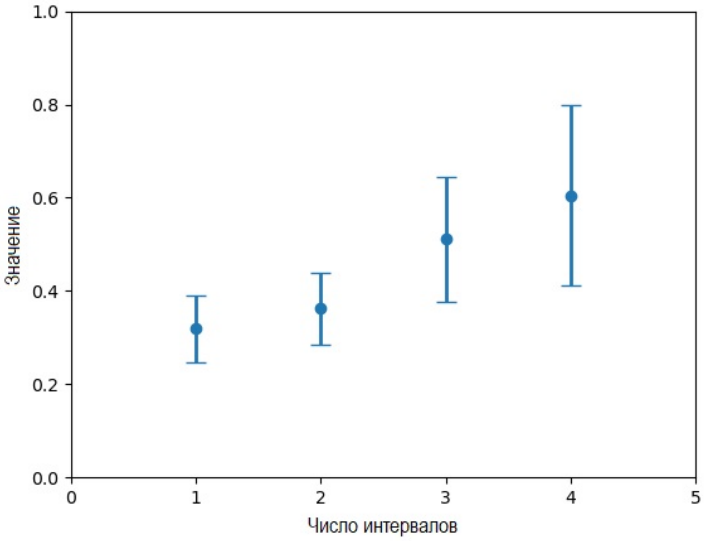

Рис. 2. Диаграмма ТСПЭ для общежития номер 1 ЯГТУ за ноябрь 2021 года

2.2. Методы выявления аномалий на основе машинного обучения

Методы машинного обучения, которые могут применяться для выявления аномалий в данных об энергопотреблении, включают как методы обучения без учителя (unsupervised learning), так и методы обучения с учителем (supervised learning) [12]. Методы обучения без учителя можно условно разделить на следующие

группы:

– методы кластеризации;
– методы понижения размерности;
– методы на основе линейных моделей;
– методы на основе локальной плотности;
– вероятностные методы;
– ансамблевые методы машинного обучения;
– методы на базе искусственных нейронных сетей (ИНС).

В данном разделе будут описаны только первые две группы методов, которые не применяются далее в данной работе. Это связано в основном с тем, что для работы этих методов требуется несколько признаков в данных. Представители остальных групп методов обучения без учителя будут рассмотрены в следующем разделе.

Классическим методом обучения без учителя для выявления закономерностей изменения электрической нагрузки, определения ТШЭ и наличия аномалий является кластеризация [24]. Кластеризация позволяет разделить данные об энергопотреблении на различные кластеры и, следовательно, помогает классифицировать их на нормальные или ненормальные. Среди методов кластеризации следует отметить метод иерархической кластеризации и алгоритм DBSCAN (Density-Based Spatial Clustering of Applications with Noise). Иерархическая кластеризация была использована для анализа профилей электрической нагрузки двух зданий университетской библиотеки [16]. Алгоритм DBSCAN был применен для определения одиночных аномалий в данных об энергопотреблении зданий ЯГТУ [1].

Другими методами обучения без учителя, традиционно используемыми для выявления аномалий, являются методы снижения размерности. Алгоритмы снижения размерности, такие как анализ главных компонент (Principal Component Analysis – PCA), уменьшают размерность данных, одновременно пытаясь минимизировать ошибку их реконструкции. Наибольшая ошибка реконструкции будет генерироваться на тех данных, которые сложнее всего смоделировать, т.е. на данных, которые могут являться

аномальными. Применение PCA для выявления аномалий в данных об энергопотреблении описано в [28].

Среди методов обучения с учителем, которые могут использоваться для выявления аномалий, в последнее время все чаще применяются ИНС. Наиболее популярными нейросетевыми архитектурами, которые используются для решения этой задачи, являются сети долгой краткосрочной памяти (Long Short Term Memory – LSTM) и сверточные нейронные сети (convolutional NN – CNN).

В [31] сеть LSTM использовалась для прогнозирования аномалий в данных об энергопотреблении и их отличия от отклонений, возникающих в зависимости от сезонности, погоды и праздников. CNN также продемонстрировали свою эффективность при обнаружении аномалий в данных временных рядов. В [18] автор решил объединить CNN и случайный лес для отслеживания аномалий потребления электроэнергии из-за краж и тем самым помочь энергоснабжающим организациям решить проблемы, связанные с неэффективным контролем использования электроэнергии.

Использование методов обучения с учителем для выявления аномалий в данных об энергопотреблении требует обучения классификаторов (бинарных или многоклассовых) на основе размеченных наборов данных, где отмечено нормальное или аномальное энергопотребление. Хотя обнаружение аномалий с применением методов обучения с учителем может дать более точные результаты, его использование ограничено на практике по сравнению с методами без учителя из-за большой трудоемкости создания размеченных наборов данных об энергопотреблении. Другой проблемой, возникающей при применении методов обучения с учителем, является то, что аномалии возникают относительно редко. Это приводит к тому, что, при классификации, классы нормальных и аномальных данных будут сильно не сбалансированы, что оказывает влияние на результаты анализа и делает их менее достоверными.

3. Выявление одиночных аномалий в данных об энергопотреблении

3.1. Методы обнаружения одиночных аномалий

В данной работе для анализа одиночных аномалий рассматриваются различные методы машинного обучения без учителя, среди которых есть как уже хорошо зарекомендовавшие себя, так и методы, появившиеся совсем недавно. Далее в работе понятия «метод» и «алгоритм» будут считаться синонимами и подменять друг друга, хотя с более строгой точки зрения это не так. Различные алгоритмы выявления одиночных аномалий приведены в таблице 1. На выбор методов повлияли такие обстоятельства, как опыт работы авторов с данными алгоритмами, красота и перспективность идеи, заложенной в алгоритм, и то, что большинство из них хорошо показали себя в численных экспериментах. Данные алгоритмы далее будут называться детекторами. Методы распределены по категориям, но некоторые из них можно отнести сразу к нескольким категориям. Для того чтобы не указывать ссылки на литературу для каждого метода и, соответственно, сократить объем статьи, здесь будет указан только один источник [8], где можно найти ссылки на литературу для каждого из перечисленных алгоритмов.

Особое внимание в работе будет уделяться алгоритмам, не требующим дополнительных параметров, таких как ROD, COPOD и ECOD, а также детекторам на базе ИНС. Эти методы, по мнению авторов, являются наиболее перспективными для применения на практике. Ну и конечно будут отмечены те алгоритмы, которые покажут хорошие результаты в обнаружении аномалий. Для них далее будет кратко приведен некоторый теоретический материал, лежащий в основе данных алгоритмов.

Детектор Histogram-based Outlier Score (HBOS) [9] определяет аномалии на основе построения гистограммы данных. В HBOS для каждого отдельного признака сначала строится одномерная гистограмма. Для категориальных признаков выполняется простой подсчет значений каждой категории и вычисляется отно-

Таблица 1. Методы анализа одиночных аномалий

Категория	Метод	Год	Описание
Линейная модель	MCD	1999	Minimum Covariance Determinant
	OCSVM	2001	One-Class Support Vector Machines
На основе плотности	LOF	2000	Local Outlier Factor
	KNN	2000	K-Nearest Neighbors
	CBLOF	2003	Clustering-Based Local Outlier Factor
	HBOS	2012	Histogram-based Outlier Score
	ROD	2020	Rotation-based Outlier Detection
Ансамблевые методы	IF	2008	Isolation Forest
	LODA	2016	Lightweight On-line Detector of Anomalies
	DIF	2023	Deep Isolation Forest
Вероятностные методы	QMCD	2001	Quasi-Monte Carlo Discrepancy
	KDE	2007	Outlier Detection with Kernel Density Functions
	Sampling	2013	Rapid distance-based outlier detection via sampling
	COPOD	2020	Copula-Based Outlier Detection
	ECOD	2022	Outlier Detection Using Empirical Cumulative Distribution Functions
ИНС	VAE	2013	Variational AutoEncoder
	AE	2015	Fully connected AutoEncoder
	DeepSVDD	2018	Deep Support Vector Data Description
	ALAD	2018	Adversarially learned anomaly detection
	SOGAAL	2019	Single-Objective Generative Adversarial Active Learning
	LUNAR	2022	Unifying Local Outlier Detection Methods via Graph Neural Networks

сительная частота. Для числовых признаков можно использовать два разных метода: статические гистограммы ширины бина и динамические гистограммы ширины бина. Первый метод – это стандартный метод построения гистограмм с использованием k бинов одинаковой ширины. Частота (относительное количество) экземпляров, попадающих в каждый бин, используется в качестве оценки плотности вероятности (высоты бинов). Динамическая ширина бина определяется следующим образом. Сначала значения сортируются, а затем фиксированное количество из N/k последовательных значений группируется в один бин, где N – общее количество экземпляров, а k – количество бинов. Поскольку ширина ячейки определяется первым и последним значением, а площадь одинакова для всех ячеек, можно вычислить высоту каждой отдельной ячейки. Это означает, что ячейки, охватывающие больший интервал диапазона значений, имеют меньшую высоту и, таким образом, представляют меньшую плотность. Однако есть одно исключение: при определенных обстоятельствах более k экземпляров данных могут иметь одинаковое значение, например, если признак является целым числом, и необходимо оценить распределение с длинным хвостом. В этом случае алгоритм должен допускать наличие более N/k значений.

Поскольку задачи обнаружения аномалий обычно включают пробелы в диапазонах значений из-за того, что выбросы далеки от нормальных данных, рекомендуется использовать режим динамической ширины, особенно если распределения неизвестны или имеют длинный хвост. Кроме того, необходимо также задать количество бинов k. Часто используемое практическое правило заключается в задании значения k как квадратный корень из количества экземпляров N.

Для каждого признака d строится индивидуальная гистограмма (независимо от того, категориальная ли она, фиксированной ширины или динамической ширины), где высота каждого отдельного бина представляет собой оценку плотности. Затем гистограммы нормализуются таким образом, чтобы максимальная высота была равна 1.0. Наконец, оценка вероятностей обнару-

жения аномалий HBOS для каждого экземпляра p вычисляется с использованием соответствующей высоты бинов, в которых находится экземпляр

$$HBOS(p) = \sum_{i=1}^{d} \log\left(\frac{1}{hict_i(p)}\right).$$

Rotation-based Outlier Detection (ROD) [3] рекомендуется к применению для обнаружения аномалий в многомерных данных, где количество признаков больше или равно 3. Сначала полное пространство признаков разбивается на комбинации 3D-подпространств, в которых векторы, представляющие точки данных в 3D-подпространстве, вращаются вокруг геометрической медианы два раза против часовой стрелки с использованием формулы вращения Родригеса. Результатом вращения являются параллелепипеды, объемы которых интерпретируются как функции стоимости и используются для определения медианных абсолютных отклонений, которые, в свою очередь, служат для получения оценок аномалий для каждого 3D-подпространства. Затем оценки аномалий полного пространства реконструируются путем взятия среднего значения оценок аномалий всех 3D-подпространств. Наконец, все данные ранжируются в порядке убывания их оценок, и данные с наивысшими оценками рассматриваются как перспективные кандидаты на аномалии.

Связь объема параллелепипеда с функцией стоимости устанавливает следующая теорема.

Теорема 1 (см. [3, с. 3])**.** *Пусть $D = \{v_1, v_2, \ldots, v_n\} \in \mathbb{R}^3$ – набор векторов, представляющих точки данных трехмерного набора данных. Если $m \in \mathbb{R}^3$ – единичный вектор геометрической медианы в D, описывающий ось вращения, то можно доказать, что $\forall v \in D$ независимого от m, ориентированный объем параллелепипеда, образованного вращением v два раза вокруг m, согласно правилу правой руки, на два последовательных угла $\theta_1 < \theta_2 \in (0, 2\pi)$, используя формулу вращения Родригеса, может быть аппроксимирован функцией стоимости, заданной как*

$$f(v, \gamma) = \| v \|^3 (\cos(\gamma) \sin(\gamma^2)), \quad (1)$$

где γ – угол между v и m.

Уравнение (1) описывает различия между векторами в наборе данных относительно их величин и угла γ, который отражает степень отклонения от m.

Детектор Lightweight On-line Detector of Anomalies (LODA) [23] определяет набор k одномерных гистограмм $\{h_i\}_{i=1}^{k}$ каждая из которых аппроксимирует плотность вероятности данных, спроецированных на один вектор проекции $\{w_i \in \mathbb{R}^d\}_{i=1}^{k}$. Векторы проекции $\{w_i\}_{i=1}^{k}$ диверсифицируют отдельные гистограммы, что является существенным требованием для улучшения производительности одного детектора для ансамблевых систем.

Выходы $LODA(x)$ представляют собой среднее значение логарифма вероятностей, вычисленных для отдельных векторов проекций. Приняв $\hat{p}_i$ для обозначения вероятности, оцененной по i-й гистограмме, и w_i для обозначения соответствующего вектора проекции, выход $LODA(x)$ можно записать как

$$LODA(p) = -k \sum_{i=1}^{k} \log(\hat{p}_i(x^\top w_i)),$$

что можно переформулировать как

$$LODA(p) = -\log(\prod_{i=1}^{k} \hat{p}_i(x^\top w_i))^{\frac{1}{k}}. \tag{2}$$

Уравнение (2) показывает, что выход LODA пропорционален отрицательному логарифмическому правдоподобию, что означает, что чем менее вероятна точка выборки, тем выше значение выхода, которое она получает, и тем вероятнее она является аномалией. Это выполняется при сильном предположении, что распределения вероятностей на векторах проекций w_i и w_j независимы $\forall i, j \in k, i \neq j$. Несмотря на то, что это почти никогда не бывает верно на практике, LODA все равно дает очень хорошие результаты. Авторы алгоритма считают, что причины этого аналогичны причинам в наивных байесовских классификаторах, которые дают условия, при которых эффекты условных зависимостей аннулируются.

Copula-Based Outlier Detection (COPOD) [20] основан на использовании функции «копула» для моделирования многомерно-

го распределения данных. COPOD использует непараметрический подход, основанный на подгонке эмпирических кумулятивных функций распределения (Cumulative Distribution Functions – CDF), называемых «эмпирическими копулами». Пусть X – d-мерный набор данных с n наблюдениями. Эмпирическая CDF определяется как

$$\hat{F}(x) = P((-\infty, x]) = \frac{1}{n}\sum_{i=1}^{n} I(X_i < x), \tag{3}$$

где I – так называемая индикаторная функция.

COPOD – это трехэтапный алгоритм, который принимает d-мерный входной набор данных $X = (X_{1,i}, X_{2,i}, \ldots, X_{d,i})$, $i = 1, \ldots, n$, и выдает вектор оценок $O(X)$. На первом этапе COPOD подгоняет d эмпирических левых хвостовых CDF $\hat{F}_1(x), \ldots, \hat{F}_d(x)$, используя уравнение (3), и d эмпирических правых хвостовых CDF $\bar{F}_1(x), \ldots, \bar{F}_d(x)$, заменяя X на $-X$. Также вычисляется вектор асимметрии, $b = [b1, \ldots, b_d]$, где

$$b_j = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x}_i)^3}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x}_i)^2}^3},$$

где x_i – i-е наблюдение, $\bar{x}_i$ – математическое ожидание i-го наблюдения.

На втором этапе для каждого X_i вычисляют наблюдения эмпирических копул $\hat{U}_{d,i} = \hat{F}_d(x_i)$ и $\hat{V}_{d,i} = \bar{F}_d(x_i)$ для левого и правого хвостов соответственно. Далее вычисляют скорректированные на асимметрию наблюдения эмпирических копул $\hat{W}_{d,i} = \hat{U}_{d,i}$, если $b_j < 0$, в противном случае $\hat{W}_{d,i} = \hat{V}_{d,i}$.

На третьем этапе вычисляется оценка

$$O_d(X_i) = max\{(-log(\hat{U}_{d,i}) - log(\hat{V}_{d,i}))/2, -log(\hat{W}_{d,i})\},$$

которая представляет собой степень аномальности измерения d. Затем оценки по всем измерениям агрегируются. Чем выше значение оценки, тем выше вероятность того, что точка окажется аномалией.

Outlier Detection Using Empirical Cumulative Distribution Functions (ECOD) [21] еще один алгоритм от авторов COPOD.

Он также основан на том факте, что аномалии являются «редкими событиями», которые проявляются в хвостах распределения. Отличается от COPOD тем, что финальная оценка вычисляется как

$$O_d(X_i) = max\{-log(\hat{U}_{d,i}), -log(\hat{V}_{d,i}), -log(\hat{W}_{d,i})\}.$$

Применение обычных автоэнкодеров (AutoEncoders – AE) и вариационных автоэнкодеров (Variational AutoEncoder – VAE) основано на том, что обученные на большом количестве нормальных данных они будут давать значительную ошибку реконструкции при подаче на вход аномальных данных. Структура нейронных сетей энкодера и декодера для AE принимается зеркально-симметричной. У VAE структура нейронных сетей энкодера и декодера может отличаться. Главное, чтобы число нейронов на входе энкодера совпадало с числом нейронов на выходе декодера.

VAE стремится построить некоторый генеративный процесс [4], предельное правдоподобие которого может быть описано как

$$\log(p_\theta(x \mid z)) = D_{kl}(q_\phi(z \mid x) \parallel p_\theta(z)) + L(\theta, \phi, x, z),$$

где $D_{kl}(\|)$ – расстояние Кульбака – Лейблера; $p_\theta(x \mid z)$ – распределение вероятности данных x при заданных латентных переменных z; $q_\phi(z \mid x)$ – распределение вероятности латентных переменных z при заданных x; $p_\theta(z)$ – распределение вероятности латентных переменных z; $L(\theta, \phi, x, z)$ – нижняя граница предельного правдоподобия; ϕ, θ – параметры распределений энкодера и декодера VAE соответственно. Тогда задача VAE сводится к максимизации нижней границы предельного правдоподобия, которое удовлетворяет условию

$$(4) \quad \begin{cases} \log(p_\theta(x \mid z)) \geqslant L(\theta, \phi, x, z) = E_{q_\phi(z|x)}[\log(p_\theta(x \mid z))] - \\ -D_{kl}(q_\phi(z \mid x) \parallel p_\theta(z)). \end{cases}$$

Чтобы сделать оптимизацию в уравнении (4) практически реализуемой, обычно делаются следующие предположения. Апостериорное $q_\phi(z \mid x)$ распределение задается как нормальное распределение с диагональной ковариационной матрицей. Априорное распределение вероятностей принимается равной $N(0, I)$. Представление распределений таким образом позволяет исполь-

зовать «трюк репараметризации», где случайная величина z_i параметризуется как дифференцируемое преобразование

$$z_i = \mu_i + \sigma_i \epsilon,$$

где μ_i, σ_i – среднее значение и среднеквадратическое отклонение, предсказанное энкодером; ϵ – случайное число из $N(0, I)$.

β-VAE представляет собой модификацию VAE [13], которая вводит регулируемый гиперпараметр β в выражение (4):

$$\begin{cases} \log(p_\theta(x \mid z)) \geqslant L(\theta, \phi, x, z) = E_{q_\phi(z|x)}[\log(p_\theta(x \mid z))] - \\ -\beta D_{kl}(q_\phi(z \mid x) \parallel p_\theta(z)). \end{cases}$$

Детектор Deep Support Vector Data Description (Deep SVDD) или Deep One-Class Classification основан на методе Support Vector Data Description (SVDD) [26]. Задачей SVDD является нахождение гиперсферы минимального объема, которая охватывает большую часть данных в пространстве признаков, а аномальные данные выходят за пределы гиперсферы. В Deep SVDD для построения гиперсферы используется специально обученная нейронная сеть.

Примем, что $X \subseteq \mathbb{R}^d$ пространство входных данных, а $F \subseteq \mathbb{R}^p$ пространство выходных признаков. Пусть отображение $\phi(\cdot, W)$: $X \to F$ будет нейронной сетью с $L \in N$ скрытыми слоями и набором весов $W = \{W^1, \ldots, W^L\}$, где W^l – веса слоя $l \in \{1, \ldots, L\}$. Тогда $\phi(x, W) \in F$ – это преобразование признаков данных $x \in X$, заданное сетью ϕ с параметрами W. Задачей Deep SVDD является совместное обучение параметров сети W и минимизация объема гиперсферы, содержащей данные, в выходном пространстве F, характеризующейся радиусом $R > 0$ и центром $c \in F$.

Для набора обучающих данных $D_n = x_1, \ldots, x_n$ на X цель обучения Deep SVDD с мягкой границей (soft-boundary Deep SVDD) определяется как

$$(5) \quad \begin{cases} \min\limits_{R,W} \{R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} max\{0, \parallel \phi(x_i, W) - c \parallel^2 - R^2\} + \\ + \frac{\lambda}{2} \sum_{l=1}^{L} \parallel W^l \parallel_F^2\}. \end{cases}$$

Первое слагаемое R^2 минимизирует размер гиперсферы. Второе слагаемое – это штрафной член для точек, лежащих вне

сферы после прохождения через сеть, т.е. если расстояние от точки до центра $\parallel \phi(x_i, W) - c \parallel$ больше радиуса R. Гиперпараметр $\nu \in (0, 1]$ контролирует компромисс между размером сферы и нарушениями границы, т.е. позволяет некоторым точкам отображаться вне сферы. Последний член – это регуляризатор на параметрах сети W с гиперпараметром $\lambda > 0$, где $\| \cdot \|_F$ обозначает норму Фробениуса.

Оптимизация цели (5) позволяет обучить параметры W таким образом, чтобы точки данных были сопоставлены с центром гиперсферы. В результате нормальные примеры данных будут располагаться близко к центру c, тогда как аномальные примеры будут находится дальше от центра или за пределами гиперсферы.

Для случая, когда большинство данных D_n являются нормальными, что часто бывает в задачах анализа аномалий, предлагается упрощенный вариант цели (5). Тогда цель One-Class Deep SVDD определяется как

$$\begin{cases} \min\limits_{W} \{\frac{1}{n}\sum_{i=1}^{n} \parallel \phi(x_i, W) - c \parallel^2 + \\ +\frac{\lambda}{2}\sum_{l=1}^{L} \parallel W^l \parallel_F^2\}. \end{cases} \tag{6}$$

One-Class Deep SVDD использует квадратичную функцию потерь для штрафования расстояния каждого представления, полученного путем преобразования сети $\phi(x, W)$, до центра $c \in F$. Второй член (6), так же как и в (5), является регуляризатором на весах сети с гиперпараметром $\lambda > 0$. Можно думать о One-Class Deep SVDD как о поиске гиперсферы минимального объема с центром c. Но в отличие от Deep SVDD с мягкой границей, где гиперсфера сжимается путем штрафования радиуса напрямую и представлений данных, которые выпадают за пределы сферы, One-Class Deep SVDD сжимает сферу путем минимизации среднего расстояния всех представлений данных до центра.

Детоктор Adversarially learned anomaly detection (ALAD) применяет для обнаружения аномалий двунаправленную генеративно-состязательную сеть (Bidirectional Generative Adversarial Networks – BiGAN) [32]. ALAD использует ошибки реконструкции, чтобы определить, является ли выборка данных аномальной.

Формально, определим $p_X(x)$ как распределение по данным x в пространстве данных X, а $p_Z(z)$ – распределение по скрытым переменным генератора z в латентном пространстве Z. Тогда обучение GAN заключается в поиске такого дискриминатора D и генератора G, которые решают задачу нахождения седловой точки $\min\limits_G \max\limits_D V(D,G)$, где

$$V(D,G) = E_{X\sim P_X}[\log(D(x))] + E_{Z\sim P_Z}[\log(1 - D(G(x)))].$$

Решение задачи поиска седловой точки описываются Леммой 1, которая показывает, что оптимальный генератор позволяет получить распределение $p_G(x)$, которое соответствует истинному распределению данных $p_X(x)$.

***Лемма* 1** (см. [32, с. 2])**.** *Для фиксированного G оптимальный дискриминатор D^*_G равен*

$$D^*_G = \frac{p_X(x)}{p_X(x) + p_Z(z)},$$

и для этого оптимального дискриминатора D^*_G глобальный минимум критерия обучения $C(G) = \max\limits_D V(D,G)$ достигается тогда и только тогда, когда $p_G(x) = p_X(x)$.

Существует несколько подходов, которые можно использовать для адаптации GAN для обнаружения аномалий. Один из подходов заключается в «инвертировании» генератора для поиска латентных переменных z, которые минимизируют ошибку реконструкции с помощью стохастического градиентного спуска [27]. Такой подход является вычислительно затратным, поскольку каждое вычисление градиента требует обратного распространения через сеть генератора. Сеть ALAD объединяет два подхода: Adversarially Learned Inference (ALI) [6] и Adversarially Learned Inference Conditional Entropy (ALICE) [14]. Основная идея подхода ALI состоит в замене генератора в GAN на пару «энкодер E и декодер (генератор) G». Вычисление латентного представления z для точки данных x в этом случае выполняется просто путем пропускания x через сеть энкодера, что приводит к значительному снижению вычислительных затрат.

Формально модель BiGAN сравнивает совместные распределения $p_G(x,z) = p_Z(z)p_G(x \mid z)$ и $p_E(x,z) = p_X(x)p_E(z \mid x)$.

Для этого разыгрывается состязательная игра. Сеть дискриминатора D_{xz}, принимает x и z в качестве входных данных и учится различать их совместные распределения, в то время как сети энкодера и генератора обучаются обманывать дискриминатор. BiGAN определяют дискриминатор D_{xz}, генератор G и энкодер E как решение задачи поиска седловой точки $\min\limits_{GE}\max\limits_{D_{xz}} V(D_{xz}, E, G)$ с $V(D_{xz}, E, G)$, определяемой как

$$\begin{cases} V(D_{xz}, E, G) = E_{X\sim P_X}[\log(D_{xz}(x, E(x)))] + \\ +E_{Z\sim P_Z}[\log(1 - (D_{xz}(G(z), z))]. \end{cases}$$

Решения задачи поиска седловой точки и свойство соответствия распределений $p_E(x, z) = p_G(x, z)$ описываются Леммой 2.

***Лемма* 2** (см. [32, с. 3])**.** *При фиксированных E и G оптимальный дискриминатор D^*_{xz} равен*

$$D^*_{xz} = \frac{p_E(x, z)}{p_E(x, z) + p_G(x, z)},$$

и для этого оптимального дискриминатора D^*_{xz} глобальный минимум критерия обучения $C(E, G) = \max\limits_{D_{xz}} V(D_{xz}, E, G)$ достигается тогда и только тогда, когда $p_E(x, z) = p_G(x, z)$.

На практике совместные распределения $p_E(x, z)$ и $p_G(x, z)$ могут быть не идентичны, поскольку обучение не обязательно сходится к решению проблемы поиска седловой точки. Это приводит к нарушению условия циклической согласованности по переменным x, когда $G(E(x)) \neq x$ [6]. Такое нарушение создаст проблемы для методов обнаружения аномалий, основанных на реконструкции.

Для решения этой проблемы подход ALICE [14] предлагает использовать условную энтропию $H^\pi(x \mid z) = -E_{\pi(x,z)}[log_\pi(x \mid z)]$ (где $\pi(x, z)$ – это совместное распределение по x и z) так, чтобы способствовать выполнению условия циклической согласованности. Это соответствует задаче поиска седловой точки $\min\limits_{GE}\max\limits_{D_{xz}} V_{ALICE}(D_{xz}, E, G)$, которая включает регуляризацию условной энтропией (V_{CE}) наложенной на энкодер E и генератор G:

$$V_{ALICE}(D_{xz}, E, G) = V(D_{xz}, E, G) + V_{CE}(E, G).$$

Регуляризацию условной энтропией можно аппроксимировать с помощью дополнительной сети дискриминатора $D_{xx}(x, x)$:

$$\begin{cases} V(D_{xx}, E, G) = E_{X\sim P_X}[\log(D_{xx}(x, x))]+ \\ +E_{Z\sim P_Z}[\log(1 - (D_{xx}(G(E(x)))))], \end{cases}$$

и, как показано в [13], такой дискриминатор действительно обеспечит циклическую согласованность.

Чтобы улучшить обучение базовой модели ALICE, авторы [32] ввели еще одну регуляризацию условной энтропией $H^{\pi}(z \mid x) = -E_{\pi(x,z)}[log_{\pi}(z \mid x)]$ с помощью дополнительного дискриминатора D_{zz}

$$\begin{cases} V(D_{zz}, E, G) = E_{X\sim P_X}[\log(D_{zz}(z, z))]+ \\ +E_{Z\sim P_Z}[\log(1 - (D_{zz}(E(G(x)))))]. \end{cases}$$

Объединяя все вместе, метод ALAD решает следующую задачу поиска седловой точки: $\min\limits_{GE}\max\limits_{D_{xz},D_{xx},D_{zz}} V(D_{xz}, D_{xx}, D_{zz}, E, G)$, где $V(D_{xz}, D_{xx}, D_{zz}, E, G)$ определяется как

$$\begin{cases} V(D_{xz}, D_{xx}, D_{zz}, E, G) = V(D_{xz}, E, G) + V(D_{xx}, E, G)+ \\ +V(D_{zz}, E, G). \end{cases}$$

Таким образом модель ALAD содержит целых пять ИНС. Это сети для энкодера E, генератора G и трех дискриминаторов D_{xx}, D_{zz}, D_{xz}, обучаемые состязательным образом.

3.2. Наборы данных

В настоящее время не наблюдается большого выбора публичных датасетов с данными об энергопотреблении, которые можно использовать для тестирования алгоритмов обнаружения аномалий. Наиболее известным из таких наборов данных является датасет LEAD1.0 (Large-scale Energy Anomaly Detection) [10]. По словам авторов, этот набор данных на данный момент является крупнейшим для анализа энергетических аномалий в открытом доступе. Датасет LEAD1.0 использовался в соревнованиях Great Energy Predictor III, проходивших в 2019 году на платформе Kaggle.

В LEAD1.0 собраны данные почасовых показаний с 1413 ПУ электроэнергии, охватывающих 16 различных типов зданий за один год. Главным достоинством датасета LEAD1.0 является то,

что он размечен на нормальные и аномальные данные. По словам авторов, процесс разметки включал ручную проверку приблизительно 12 миллионов точек данных. Всего было выявлено 199 640 аномальных случаев. Хотя авторы [10] провели титаническую работу, некоторые важные моменты они оставили неописанными в своей статье. Например, не ясно, как определялись недельные и месячные ТШЭ, которые использовались для определения аномалий.

Несмотря на это, датасет LEAD1.0 был использован в данной работе в качестве основного для тестирования алгоритмов обнаружения одиночных аномалий. Недостатком этого набора данных при анализе одиночных аномалий является то, что в нем присутствуют как одиночные, так и коллективные аномалии. Причем для некоторых зданий коллективные аномалии являются преобладающими. Можно рассматривать коллективные аномалии как множество одиночных, но такой подход имеет следующие недостатки:

– для некоторых типов коллективных аномалий снижается эффективность методов, рассчитанных на выявление одиночных аномалий;

– для выявления коллективных аномалий существуют отдельные эффективные методы.

Поэтому далее при выполнении численных экспериментов с датасетом LEAD1.0 один из типов коллективных аномалий был исключен.

В качестве дополнительного набора данных использовался датасет, составленный из почасовых данных энергопотребления двух зданий ЯГТУ за шесть месяцев 2020 и 2021 годов. Данный набор данных будем называть датасет ЯГТУ. Он удобен тем, что в нем присутствуют только одиночные аномалии. Датасет был размечен авторами ранее с применением описанного в разделе 2.1 ТСПЭ на основании предыдущих исследований [22].

3.3. Инструменты для выявления одиночных аномалий

Язык Python сейчас является наиболее популярным языком программирования для анализа данных, в том числе и в области

анализа аномалий. Функции, реализующие алгоритмы выявления одиночных аномалий, имеются во многих библиотеках языка Python. Например, популярная библиотека scikit-learn включает функции для алгоритмов MCD, LOF и Isolation Forest. Но, несомненно, самая большая коллекция алгоритмов собрана в библиотеке PyOD [8]. Библиотека PyOD включает в себя более 50 алгоритмов обнаружения аномалий, начиная с алгоритмов, разработанных в 1977 году, и до современных алгоритмов 2022 и 2023 годов. Все детекторы из таблицы 1 представлены в библиотеке PyOD. Поэтому авторы выбрали эту библиотеку в качества основного инструмента для выявления одиночных аномалий.

3.4. Contamination и другие параметры

Все алгоритмы, входящие в состав библиотеки PyOD, имеют параметр contamination, который используется для определения порога (threshold). В результате работы детекторов для всех точек данных устанавливаются значения вероятностей отнесения данной точки к аномальной. Если для каких-то точек значения вероятностей превышают порог, то такие точки считаются аномалиями. Отсюда возникает проблема настройки параметра contamination для каждого конкретного детектора. Даже если исходный алгоритм, например такой как ROD или COPOD, вообще не имеет дополнительных параметров, то для него все равно необходима настройка параметра contamination. Далее будут описаны некоторые варианты решения проблемы настройки данного параметра.

Большинство детекторов кроме параметра contamination имеют другие и параметры, связанные с природой самого алгоритма. В зависимости от алгоритма количество параметров может изменяться от одного–двух до десяти и более. Например, у алгоритма VAE имеется 16 основных параметров, не считая параметра contamination и вспомогательных параметров. Понятно, что чем больше параметров у алгоритма, тем сложнее найти такое сочетание параметров, которое будет давать наилучший результат.

3.5. Результаты численных экспериментов по выявлению одиночных аномалий с помощью детекторов

Описанные в разделе 3.1 детекторы были использованы для выявления одиночных аномалий в данных датасетов LEAD1.0 и ЯГТУ. Данные датасетов разбивались по отдельным зданиям, обрабатывались, а затем разделялись на обучающее и тестовое множества. На обучающем множестве производилось обучение моделей, а на тестовом проверялось качество их работы по различным метрикам. Из-за различия используемых наборов данных этапы обработки данных для каждого из датасетов отличались. Этапы обучения моделей и проверочного тестирования для обеих датасетов выполнялись одинаково.

Сначала рассмотрим результаты анализа аномалий для датасета LEAD1.0. В отличие от соревнований Kaggle, где использовались данные сразу по всем зданиям датасета LEAD1.0 и целью было получить наибольшее совпадение с размеченными данными, авторы решают более практичные задачи. Для энергопотребителей и энергоснабжающих организаций часто полезнее получение аналитики по отдельным энергообъектам. К тому же это значительно уменьшает объем анализируемых данных, а следовательно, ускоряет процесс анализа. Это становится особенно важным, когда применяется не один или два алгоритма, как на соревнованиях, а 21, как в данном исследовании. Поэтому все данные датасета LEAD1.0 были разбиты по отдельным зданиям. При этом разбивка данных по зданиям показала, что у одних зданий число аномалий получилось слишком мало, а у других зданий преобладающими являются коллективные аномалии. Все такие здания были исключены из рассмотрения.

Поскольку, как было сказано ранее, авторы датасета LEAD1.0 не представили полную информацию, как производилась разметка данных, то первичная обработка данных для данного датасета была сведена к удалению из данных одного из типов коллективных аномалий, заполнению пропусков и нормализации данных. Встречающиеся коллективные аномалии в данных были разделены на два типа. К первому типу были отнесены коллек-

тивные аномалии, в которых все данные в пределах аномалии имеют одинаковые значения с учетом определенной погрешности. Второй тип включал аномалии, в которых все данные в пределах аномалии имеют разные значения. Анализ данных показал, что в датасете LEAD1.0 преобладающим является первый тип коллективных аномалий. Такой тип аномалий чаще всего связан с отключениями или значительным снижением нагрузки у энергопотребителей на длительное время и на практике не возникает проблем с выявлением таких аномалий (энергопотребители часто сами сообщают о них). Поэтому данный тип коллективных аномалий был исключен из рассмотрения.

Анализ аномалий производился только по данным об энергопотреблении (показаниям ПУ электроэнергии) для отдельных зданий, т.е. набор данных включал только один признак. Оценка качества работы алгоритмов на тестовом множестве производилось с использованием следующего набора метрик: Precision, Recall, F1 (F-мера), AUC. Эти метрики чаще всего используются при оценке результатов работы моделей, обученных методом обучения без учителя и используемых для выявления аномалий. Хотя далее основной метрикой, по которой будет сравниваться работа алгоритмов, будет метрика F1, остальные метрики также могут дать информацию об их работе.

Для каждого из используемых детекторов производилась индивидуальная настройка параметра contamination путем грубого прямого перебора вариантов с целью достижения максимума метрики F1 по размеченным данным. В данном случае необходимо было получить хотя бы приблизительную оценку параметра contamination. Далее будет рассмотрено, как можно обойти данную ситуацию. Для алгоритмов, имеющих множество других параметров, оптимизация производилась только для ограниченного числа наиболее важных параметров. Значение остальных параметров принималось равным значениям по умолчанию.

Результаты численных экспериментов, полученных для зданий с номерами 1319 и 1258 из датасета LEAD1.0 показаны в таблицах 2 и 3.

Таблица 2. Результаты анализа одиночных аномалий для здания 1319

Детектор	Precision	Recall	F1	AUC
MCD	0,3158	0,4390	0,3673	0,7083
OCSVM	0,3704	0,4878	0,4211	0,7341
LOF	0,0070	0,0488	0,0123	0,4432
KNN	1,0	0,171	0,2917	0,5854
CBLOF	0,1795	0,5122	0,2658	0,7285
HBOS	0,7407	0,4878	0,5882	0,7419
ROD	0,875	0,1707	0,2857	0,5851
IF	0,3846	0,4878	0,4301	0,7347
LODA	0,72	0,4390	0,5455	0,7175
DIF	1,0	0,2927	0,4528	0,6463
QMCD	0,3272	0,4390	0,375	0,7089
KDE	0,3333	0,4390	0,3789	0,7092
Sampling	0,2656	0,4146	0,3238	0,6938
COPOD	0,2653	0,3171	0,2889	0,6482
ECOD	0,4878	0,4878	0,4878	0,7379
VAE	0,3704	0,4878	0,4211	0,7341
AE	0,1842	0,3415	0,2393	0,6529
DeepSVDD	1,0	0,2683	0,4231	0,6341
ALAD	0,3	0,4390	0,3564	0,7075
SOGAAL	0,3278	0,4878	0,392	0,7321
LUNAR	0,1852	0,2439	0,2105	0,6093

Анализ данных таблиц 2 и 3 показывает, что большинство детекторов неплохо справилось с задачей выявления аномалий. Для первого здания 8 детекторов показали значение F1 от 0,4 и выше и еще 4 алгоритма дали значение F1 близкие к 0,4. Для второго здания 16 детекторов показали значение F1 выше 0,5. Несмотря на разные подходы к обработке данных это в целом совпадает с результатами, представленными в [10], с учетом того, что там было представлено только 7 из рассмотренных здесь алгоритмов. Из алгоритмов без дополнительных параметров для первого здания только ECOD показал хороший результат. Для второго здания к нему присоединился COPOD. Из нейросетевых детекторов для первого здания хорошие результаты показа-

Таблица 3. Результаты анализа одиночных аномалий для здания 1258

Детектор	Precision	Recall	F1	AUC
MCD	0,525	0,6	0,56	0,7946
OCSVM	0,525	0,6	0,56	0,7946
LOF	–	–	–	–
KNN	0,833	0,5714	0,678	0,7846
CBLOF	0,525	0,6	0,56	0,7946
HBOS	1,0	0,6	0,5882	0,75
ROD	–	–	–	–
IF	0,525	0,6	0,56	0,7946
LODA	1,0	0,6	0,75	0,8
DIF	1,0	0,6	0,75	0,8
QMCD	0,525	0,6	0,56	0,7946
KDE	0,525	0,6	0,56	0,7946
Sampling	0,2656	0,4146	0,3238	0,7991
COPOD	0,8077	0,6	0,6885	0,7986
ECOD	0,5882	0,5714	0,5797	0,7817
VAE	0,75	0,6	0,6667	0,798
AE	0,0540	0,0571	0,0556	0,5186
DeepSVDD	1,0	0,6	0,75	0,8
ALAD	0,4468	0,6	0,5122	0,7926
SOGAAL	0,1618	0,6286	0,2573	0,7817
LUNAR	0,3208	0,4857	0,3863	0,7326

ли VAE и DeepSVDD. Для второго здания к ним присоединился ALAD. Лучшие результаты для первого здания показали детекторы HBOS и LODA, для второго – алгоритмы LODA, DIF и DeepSVDD.

Теперь рассмотрим результаты анализа аномалий для датасета ЯГТУ. Результаты численных экспериментов, полученных по данным об энергопотреблении для здания общежития номер 1 ЯГТУ, показаны в таблице 4. Прочерки в таблицах 3 и 4 означают, что детекторы выдали нулевые значения метрик из-за внутренних исключений, связанных с отсутствием предсказанных образцов.

Анализ данных таблицы 4 показывает, что результаты определения аномалий для датасета ЯГТУ в целом лучше, чем для

Таблица 4. Результаты анализа одиночных аномалий для здания ЯГТУ

Детектор	Precision	Recall	F1	AUC
MCD	0,5833	0,6364	0,6087	0,8147
OCSVM	0,5833	0,6364	0,6087	0,8147
LOF	–	–	–	–
KNN	0,1667	0,5454	0,2553	0,7516
CBLOF	0,4285	0,5454	0,48	0,767
HBOS	0,75	0,5454	0,6316	0,7713
ROD	0,5833	0,6364	0,6087	0,8147
IF	0,5384	0,6364	0,5833	0,8139
LODA	0,75	0,5454	0,6316	0,7713
DIF	1,0	0,0909	0,1667	0,5454
QMCD	0,5	0,7272	0,5926	0,8579
KDE	0,5833	0,6364	0,6087	0,8147
Sampling	0,5833	0,6364	0,6087	0,8147
COPOD	0,5385	0,6364	0,5833	0,8139
ECOD	0,0769	0,0909	0,0833	0,537
VAE	0,5833	0,6364	0,6087	0,8147
AE	0,5833	0,6364	0,6087	0,8147
DeepSVDD	0,5833	0,6364	0,6087	0,8147
ALAD	0,5833	0,6364	0,6087	0,8147
SOGAAL	–	–	–	–
LUNAR	0,074	0,1818	0,1053	0,5733

датасета LEAD1.0. При этом 9 детекторов из 21 показывают одинаковые высокие результаты. Здесь так же, как и для первого здания датасета LEAD1.0, лучшие результаты показали детекторы HBOS и LODA. Из алгоритмов без дополнительных параметров алгоритмы ROD и COPOD показали хорошие результаты. Из нейросетевых детекторов все алгоритмы, за исключением SOGAAL и LUNAR, показали одинаково высокие результаты.

Хорошая работа большинства детекторов аномалий для датасета ЯГТУ объясняется тем, что данный набор данных был размечен с использованием описанного в разделе 2.1 ТСПЭ так, чтобы аномалии представляли собой данные, не соответствующие данному ТШЭ. Кроме того, в датасете присутствуют только

отдельные одиночные аномалии и нет коллективных аномалий. Все это значительно облегчило работу детекторов аномалий.

3.6. Алгоритмы определения порога

Самая большая проблема детекторов аномалий, рассмотренных в предыдущем разделе, – это необходимость постоянной настройки параметра contamination. Как показали численные эксперименты, значение этого параметра зависит как от данных, с которыми работает алгоритм, так и от типа детектора. Такая ситуация значительно усложняет анализ одиночных аномалий на практике.

Ранее было сказано о том, что параметр contamination используется для определения порога, который применяется для разделения нормальных и аномальных данных. Порог в библиотеке PyOD определяется по формуле

$$threshold = percentile(p_s, q),$$

где $percentile$ – функция для вычисления доли наблюдений, отделяемых значением процентили; p_s – значения вероятностей обнаружения аномалий; q – значение процентили. При этом q вычисляется как

$$q = 100(1 - contamination).$$

Точки данных, для которых значения вероятностей превышают порог, будут отнесены к аномалиям. Возникает вопрос: а нельзя ли определять порог, исходя из значений вероятности обнаружения аномалий, каким-нибудь другим способом без использования параметра contamination? Одним из решений данной проблемы являются алгоритмы пороговой оценки вероятностей обнаружения аномалий в данных. Такие алгоритмы далее будут называться пороговыми алгоритмами (thresholding algorithms). Подход заключается в применении различных статистических и других алгоритмов непосредственно к оценкам вероятностей обнаружения аномалий, генерируемых детекторами. Пороговые алгоритмы заменяют необходимость устанавливать параметр contamination или заставлять пользователя заранее угадывать количество аномалий, которые могут существовать в наборе данных.

Существует библиотека PyThresh, которая включает более 30 различных пороговых алгоритмов [7]. Эта библиотека была написана для работы в тандеме с библиотекой PyOD и имеет похожий синтаксис и структуры данных. Различные пороговые алгоритмы из библиотеки PyThresh, которые применялись в данной работе, приведены в таблице 5. Ссылки на литературу для каждого из методов можно найти в [7]. Как видно из таблицы 5 некоторые методы могут выступать и как детекторы, и как пороговые алгоритмы.

Таблица 5. Пороговые алгоритмы

Алгоритм	Описание
AUCP	Area Under Curve Percentage
CLF	Trained Linear Classifier
CLUST	Clustering Based
DSN	Distance Shift from Normal
EB	Elliptical Boundary
FILTER	Filtering Based
HIST	Histogram Based
IQR	Inter-Quartile Region
MAD	Median Absolute Deviation
META	Meta-model Trained Classifier
MIXMOD	Normal & Non-Normal Mixture Models
OCSVM	One-Class Support Vector Machines
QMCD	Quasi-Monte Carlo Discrepancy
REGR	Regression Based
VAE	Variational AutoEncoder
ZSCORE	Z-score
COMB	Thresholder Combination

Пороговые алгоритмы анализируют вероятности обнаружения аномалий, предварительно сгенерированных детекторами. При этом возникает вопрос: какие пороговые алгоритмы лучше подходят для того или иного детектора? Можно ли подобрать наилучшую в плане повышения точности выявления аномалий пару детектора и порогового алгоритма? Нужно ли использовать (если такие есть) одинаковые детекторы и пороговые алгоритмы? Даниель Кулик, один из разработчиков библиотеки PyThresh,

утверждает, что чем более похожи два алгоритма, тем меньше вероятность определения иррационального порогового значения [29]. Однако он же отмечает, что это далеко не всегда так.

Найденные на основании численных экспериментов наилучшие сочетания детекторов и пороговых алгоритмов и точность, достигаемая при этом по метрикам F1 и AUC для зданий 1319 и 1258 датасета LEAD1.0, показаны в таблицах 6 и 7.

Таблица 6. Наилучшие сочетания алгоритмов для здания 1319

Детектор	Пороговый алгоритм	F1	AUC
MCD	DSN(metric='RES')	0,5806	0,7186
OCSVM	–	–	–
LOF	–	–	–
KNN	MTT()	0,4528	0,6463
CBLOF	CLF(method='simple')	0,3922	0,622
HBOS	–	–	–
ROD	DSN(metric='JS')	0,2917	0,5854
IF	–	–	–
LODA	CLF(method='simple')	0,5882	0,7419
DIF	MIXMOD(method='mean')	0,5714	0,7413
QMCD	FILTER(method='detrend')	0,3871	0,7098
KDE	FILTER(method='detrend')	0,3922	0,622
Sampling	META(method='GNBC')	0,5806	0,7187
COPOD	DSN(metric='KS')	0,4561	0,6577
ECOD	DSN(metric='KS')	0,6333	0,7317
VAE	META(method='LIN')	0,5614	0,6951
AE	FILTER(method='savgol')	0,2785	0,7417
DeepSVDD	DSN(metric='RES')	0,6452	0,7436
ALAD	FILTER(method='detrend')	0,6	0,7192
SOGAAL	–	–	–
LUNAR	–	–	–

В таблице 8 показана аналогичная информация для датасета ЯГТУ. Для пороговых алгоритмов также приведено значение основного параметра (это может быть название метода или метрики). Прочерки в таблице означают, что для детектора не было найдено подходящего порогового алгоритма, способного повысить точность выявления аномалий.

Таблица 7. Наилучшие сочетания алгоритмов для здания 1258

Детектор	Пороговый алгоритм	F1	AUC
MCD	DSN(metric='RES')	0,75	0,8
OCSVM	FILTER(method='hilbert')	0,7368	0,7997
LOF	–	–	–
KNN	FILTER(method='savgol')	0,7272	0,7857
CBLOF	CLF(method='simple')	0,75	0,8
HBOS	–	–	–
ROD	DSN(metric='JS')	0,75	0,8
IF	FILTER(method='hilbert')	0,75	0,8
LODA	CLF(method='simple')	0,75	0,8
DIF	CLUST(method='birch')	0,75	0,8
QMCD	CLUST(method='birch')	0,75	0,8
KDE	FILTER(method='resample')	0,75	0,8
Sampling	CLF(method='simple')	0,7241	0,7994
COPOD	HIST(method='minimum')	0,75	0,8
ECOD	DSN(metric='KS')	0,6153	0,7829
VAE	META(method='LIN')	0,75	0,8
AE	FILTER(method='savgol')	0,1	0,7
DeepSVDD	DSN(metric='RES')	0,75	0,8
ALAD	FILTER(method='hilbert')	0,7368	0,7997
SOGAAL	–	–	–
LUNAR	–	–	–

Сравнивая информацию таблиц 6, 7, 8 и таблиц 2, 3, 4, можно отметить, что применение пороговых алгоритмов во многих случаях может существенно повысить точность обнаружения аномалий по сравнению с точностью, обеспечиваемой детекторами. Там же, где детектор уже получил максимальную точность, применение порогового алгоритма не позволяет улучшить результат. Это хорошо видно на примере здания ЯГТУ. Также можно заметить, что есть повторяющиеся пары детекторов и пороговых алгоритмов. Это пары: MCD – DSN(metric='RES'), KNN – FILTER(method='savgol'), CBLOF – CLF(method='simple'), ROD – DSN(metric='JS'), LODA – CLF(method='simple'), ECOD – DSN(metric='KS'), VAE – META(method='LIN'), AE – FILTER(method='savgol'),

Таблица 8. Наилучшие сочетания алгоритмов для здания ЯГТУ

Детектор	Пороговый алгоритм	F1	AUC
MCD	–	–	–
OCSVM	–	–	–
LOF	–	–	–
KNN	FILTER(method='savgol')	0,2716	0,9584
CBLOF	–	–	–
HBOS	–	–	–
ROD	REGR(method='siegel')	0,6315	0,7713
IF	–	–	–
LODA	–	–	–
DIF	CLUST(method='mshift')	0,3142	0,9661
QMCD	–	–	–
KDE	–	–	–
Sampling	–	–	–
COPOD	–	–	–
ECOD	DSN(metric='TMT')	0,2528	0,9542
VAE	–	–	–
AE	–	–	–
DeepSVDD	–	–	–
ALAD	–	–	–
SOGAAL	–	–	–
LUNAR	DSN(metric='LK')	0,2222	0,0866

DeepSVDD – DSN(metric='RES'). К пороговым алгоритмам, применение которых чаще других обеспечивает повышение точности, относятся DSN, FILTER и CLF.

3.7. Ансамбли детекторов

Как хорошо известно, комбинации моделей (ансамбли) в машинном обучении часто позволяют получить более высокую точность решения задачи по сравнению с точностью достигаемой отдельными моделями. Поэтому можно попробовать скомбинировать возможности детекторов, которые показали высокие результаты в результате численных экспериментов. В библиотеке PyOD комбинирование возможно как на уровне самих детекторов, так и полученных в результате работы детекторов вероятно-

стей обнаружения аномалий. Первый вариант можно реализовать с помощью алгоритма Locally Selective Combination of Parallel Outlier Ensembles (LSCP). LSCP реализует стратегию Average of Maximum, о которой будет сказано далее.

Во втором варианте комбинируются вероятности обнаружения аномалий, полученные в результате работы детекторов. К вероятностям могут применяться различные стратегии комбинирования. К ним относятся: усреднение (Average), взвешенное усреднение (Weighted Average), взятие максимума (Maximization), Average of Maximum и Maximum of Average. Первые три операции выполняются поточечно для всех использованных детекторов. В двух последних операциях детекторы делятся на подгруппы. В операции Average of Maximum берутся максимумы по каждой подгруппе, а затем они усредняются. В операции Maximum of Average рассчитываются средние значения по каждой подгруппе, а затем по ним находится максимум. Для реализации различных стратегий комбинирования в библиотеке PyOD есть соответствующие функции. Для нахождения аномальных точек к вероятностям обнаружения аномалий, полученным в результате комбинирования, необходимо применить какой-нибудь пороговый алгоритм.

В таблице 9 показаны результаты, полученные для различных датасетов и зданий с помощью описанных стратегий комбинирования и применения порогового алгоритма VAE.

Таблица 9. Результаты комбинирования

Стратегия	1319	1258	ЯГТУ
Average	0,6452	0,6774	0,4888
Maximization	0,4528	0,75	0,6087
Average of Maximum	0,6452	0,6774	0,4888
Maximum of Average	0,4528	0,75	0,6087

Для здания 1319 комбинировались вероятности обнаружения аномалий детекторов: HBOS, LODA, DIF, ECOD, VAE и DeepSVDD. Для здания 1258 комбинировались вероятности детекторов: HBOS, LODA, DIF, COPOD, VAE и DeepSVDD.

Для здания ЯГТУ комбинировались вероятности детекторов: HBOS, LODA, MCD, ROD, VAE и DeepSVDD. Пороговый алгоритм VAE оказался одним из группы пороговых алгоритмов, обеспечивающих наилучший результат одновременно для всех рассматриваемых зданий.

Для каждого здания из всех стратегий комбинирования необходимо выбирать те, которые дают наивысшую точность. Сравнивая результаты для лучших стратегий из таблицы 9 с результатами, полученными ранее в данной работе, можно отметить, что применение стратегий комбинирования совместно с пороговым алгоритмом позволяет получить точность на уровне той, которая была достигнута для каждого здания с помощью лучших детекторов или пар «детектор – пороговый алгоритм».

4. Выводы

Таким образом, в работе описаны исследования по выявлению одиночных аномалий в данных об энергопотреблении на примере двух разных наборов данных. Рассмотрены способы построения ТШЭ, одним из которых является ТСПЭ. Изложен авторский способ построения ТСПЭ, показавший свою эффективность в численных экспериментах. Приведен обзор методов анализа аномалий на основе машинного обучения, из которых авторами был выбран 21 метод, подходящий для выявления одиночных аномалий. Описаны наборы данных и инструменты, которые были использованы для проведения численных экспериментов.

По результатам численных экспериментов были отмечены методы, наиболее удачно справившиеся с задачей выявления одиночных аномалий. Особое внимание в работе уделялось методам, не требующим дополнительных параметров, и современным перспективным методам на базе ИНС. К сожалению, алгоритмы ROD, COPOD и ECOD, не требующие дополнительных параметров, не показали стабильно высокого качества работы для всех вариантов данных. Во всех случаях один или два алгоритма показывали хорошие результаты, а один из них, наоборот, плохие. Тем не менее отсутствие дополнительных параметров делает эти ал-

горитмы привлекательными для использования при анализе аномалий. Из нейросетевых методов следует отметить алгоритмы VAE, DeepSVDD и ALAD, применение которых во всех случаях позволяло достичь высоких показателей. Авторы считают эти алгоритмы наиболее перспективными для анализа аномалий из всех методов на базе ИНС.

Лучшим алгоритмом по результатам численных экспериментов оказался алгоритм LODA. Также очень хорошо себя зарекомендовал алгоритм HBOS. Эти два метода показали стабильно высокие результаты во всех испытаниях. Оба метода имеют одинаковую природу и основаны на построении гистограмм. Авторы рекомендуют всем исследователям, кто столкнется с задачей анализа аномалий, попробовать эти алгоритмы в первую очередь.

Одной из главных проблем, затронутых в работе, является проблема настройки параметра contamination для каждого рассмотренного алгоритма. Одним из решений данной проблемы является использование пороговых алгоритмов. Пороговые алгоритмы позволяют пользователю не заботиться о поиске необходимого значения параметра contamination. В работе показано, что если детектор выявляет аномалии недостаточно хорошо (параметр contamination не настроен), то применение пороговых алгоритмов может существенно повысить точность обнаружения аномалий. Это особенно актуально при анализе аномалий для не размеченных наборов данных, когда нельзя проконтролировать полученную точность вычислений. При этом остается проблема выбора подходящего порогового алгоритма, обеспечивающего наиболее высокую точность. К сожалению, здесь нет определенного решения, подходящего для всех случаев. В работе отмечено, что к пороговым алгоритмам, применение которых для задач анализа аномалий в данных об энергопотреблении,чаще других обеспечивает повышение точности, относятся DSN, FILTER и CLF. Следовательно, эти алгоритмы необходимо попробовать в первую очередь.

Применять пороговые алгоритмы можно как к результатам работы отдельных детекторов, так и к результатам работы ансам-

блей детекторов, полученных с использованием различных стратегий комбинирования. В работе показано, что применение стратегий комбинирования совместно с пороговым алгоритмом позволяет обеспечить точность на уровне той, которая может быль достигнута с помощью отдельных детекторов или пар «детектор – пороговый алгоритм». Таким образом, возможны два варианта, обеспечивающие примерно одинаковый результат. Первый вариант заключается в использовании отдельных пар «детектор – пороговый алгоритм», а затем в выборе наилучшей пары. Второй вариант – это применение стратегий комбинирования детекторов совместно с пороговым алгоритмом. Во втором варианте по сравнению с первым дополнительно требуется использование стратегий комбинирования, но зато в этом случае необходим только один пороговый алгоритм. Это можно отнести к преимуществам второго варианта.

Полученные результаты могут использоваться как энергоснабжающими организациями, так и энергопотребителями для повышения качества и надежности энергоснабжения.

Литература

1. МАРЬЯСИН О.Ю., ЛУКАШОВ А.И., ТИХОМИРОВ Л.И. *Обнаружение аномальных отклонений от типового профиля энергопотребления* // Математические методы в технике и технологиях. – 2022. – №8. – Р. 108–113.
2. *О внесении изменений в некоторые акты Правительства Российской Федерации по вопросам совершенствования организации учета электрической энергии. Постановление Правительства Российской Федерации от 29 июня 2020 г.* – №950. – 32 с.
3. ALMARDENY Y., BOUJNAH N., CLEARY F. *A Novel Outlier Detection Method for Multivariate Data* // IEEE Trans. on Knowledge and Data Engineering. – Vol. 32. – 2020. – P. 1–13.

4. BURGESS C.P., HIGGINS I., PAL A. et al. *Understanding disentangling in β-VAE* // arXiv:1804.03599v1. – 2018. – P. 1–11.
5. CHANDOLA V., BANERJEE A., KUMAR V. *Anomaly Detection: A Survey* // ACM Computing Surveys. – 2009. – Vol. 41. – P. 1–58.
6. DUMOULIN V., BELGHAZI I., POOLE B. et al. *Adversarially learned inference* // arXiv:1606.00704v3. – 2017. – P. 1–18.
7. *GitHub – KulikDM/pythresh: Outlier Detection Thresholding*. – URL: https://github.com/KulikDM/pythresh (дата обращения: 30.08.2024).
8. *GitHub – yzhao062/pyod: A Python Library for Outlier and Anomaly Detection, Integrating Classical and Deep Learning Techniques*. – URL: https://github.com/yzhao062/pyod (дата обращения: 30.08.2024).
9. GOLDSTEIN M., DENGEL A. *Histogram-based Outlier Score (HBOS): A fast Unsupervised Anomaly Detection Algorithm* // Conference KI-2012. – 2012. – P. 1–6.
10. GULATI M., ARJUNAN P. *LEAD1.0: A large-scale annotated dataset for energy anomaly detection in commercial buildings* // arXiv:2203.17256v1. – 2022. – P. 1–5.
11. HABEN S., SINGLETON C., GRINDROD P. *Analysis and clustering of residential customers energy behavioral demand using smart meter data* // IEEE Trans. on Smart Grid. – 2016. – Vol. 1. – P. 136–144.
12. HIMEUR Y., GHANEM K., ALSALEMI A. et al. *Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives* // Applied Energy. – 2021. – Vol. 287. – P. 1–26.
13. KINGMA D.P., WELLING M. *Auto-Encoding Variational Bayes* // Int. Conf. on Learning Representations. – 2013. – P. 1–14.
14. LI C., LIU H., CHEN C. et al. *ALICE: Towards Understanding Adversarial Learning for Joint Distribution Matching* // arXiv:1709.01215v2. – 2017. – P. 1–22.

15. LI K., MA Z., ROBINSON D. et al. *Identification of typical building daily electricity usage profiles using Gaussian mixture model-based clustering and hierarchical clustering* // Applied Energy. – 2018. – Vol. 231. – P. 331–342.
16. LI K., YANG R.J., ROBINSON D. et al. *An agglomerative hierarchical clustering-based strategy using Shared Nearest Neighbours and multiple dissimilarity measures to identify typical daily electricity usage profiles of university library buildings* // Energies. – 2019. – Vol. 174. – P. 735–748.
17. LINDEMANN B., MASCHLER B., SAHLAB N. et al. *A survey on anomaly detection for technical systems using LSTM networks* // Computers in Industry. – 2021. – Vol. 131. – P. 1–11.
18. LI S., HAN Y., YAO X. et al. *Electricity theft detection in power grids with deep learning and random forests* // Journal of Electrical and Computer Engineering. – 2019. – P. 1–12.
19. LIU X., DING Y., TANG H. et al. *A data mining-based framework for the identification of daily electricity usage patterns and anomaly detection in building electricity consumption data* // Energy and Buildings. – 2021. – Vol. 231. – P. 1–22.
20. LI Z., ZHAO Y., BOTTA N. et al. *COPOD: Copula-Based Outlier Detection* // arXiv:2009.09463v1. – 2020. – P. 1–6.
21. LI Z., ZHAO Y., HU X. et al. *ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions* // arXiv:2201.00382v3. – 2022. – P. 1–13.
22. MARYASIN O.YU., TIHOMIROV L. *Analysis of Point and Collective Anomalies in Energy Consumption Data* // Int. Russian Automation Conf. (RusAutoCon–2023). – 2023. – P. 431–436.
23. PEVNY T. *Loda: Lightweight on-line detector of anomalies* // Mach Learn. – 2016. – Vol. 102. – P. 275–304.
24. RAJABI A., ESKANDARI M., GHADI M.J. et al. *A comparative study of clustering techniques for electrical load pattern segmentation* // Renewable and Sustainable Energy Reviews. – 2020. – Vol. 120. – P. 1–20.

25. RASANEN T., KOLEHMAINEN M. *Feature-Based Clustering for Electricity Use Time Series Data* // Int. Conf. on Adaptive and Natural Computing Algorithms. – 2009. – P. 401–412.
26. RUFF L., VANDERMEULEN R.A., GORNITZ N. et al. *Deep One-Class Classification* // 35th Int. Conf. on Machine Learning. – 2018. – P. 1–10.
27. SCHLEGL T., SEEBOCK P., WALDSTEIN S.M. et al. *Unsupervised anomaly detection with generative adversarial networks to guide marker discovery* // Int. Conf. on Information Processing in Medical Imaging. – 2017. – P. 146–157.
28. SIAL A., SINGH A., MAHANTI A. *Detecting anomalous energy consumption using contextual analysis of smart meter data* // Wireless Networks. – 2019. – P. 1–18.
29. *Thresholding Outlier Detection Scores with PyThresh*. – URL: https://towardsdatascience.com/thresholding-outlier-detection-scores-with-pythresh-f26299d14fa (дата обращения: 30.08.2024).
30. VERLEYSEN M., FRANCOIS D. *The curse of dimensionality in data mining and time series prediction* // Int. Work-Conf. on Artificial Neural Networks. – 2005. – P. 758–770.
31. WANG X., ZHAO T., LIU H., HE R. *Power consumption predicting and anomaly detection based on long short-term memory neural network* // IEEE 4th Int. Conf. on Cloud Computing and Big Data Analysis. – 2019. – P. 487–491.
32. ZENATI H., ROMAIN M., FOO C.S. et al. *Adversarially Learned Anomaly Detection* // arXiv:1812.02288v1. – 2018. – P. 1–11.

DETECTING POINT ANOMALIES IN ENERGY CONSUMPTION DATA USING UNSUPERVISED MACHINE LEARNING METHODS

Oleg Maryasin, Yaroslavl State Technical University, Yaroslavl, PhD in technique, associate professor (maryasin2003@list.ru).

Leonid Tihomirov, Yaroslavl State Technical University, Yaroslavl, graduate student (lenusscik@yandex.ru).

Abstract: The paper describes studies on detecting point anomalies in energy consumption data using two different data sets as an example. Methods for constructing typical energy consumption patterns are considered and the authors' method for constructing a typical daily energy consumption profile is presented. To conduct numerical experiments, the authors selected 21 unsupervised machine learning methods suitable for detecting point anomalies. Based on the results of numerical experiments, the methods that most successfully coped with the task of detecting point anomalies were noted. Particular attention in the work was paid to methods that do not require additional parameters and modern, promising methods based on artificial neural networks. According to the test results, the best algorithms were statistical algorithms based on constructing histograms. One of the main problems addressed in the work is the problem of setting the contamination parameter for each considered algorithm. One of the solutions to this problem is the use of threshold algorithms. It is shown that if the original algorithm does not detect anomalies well enough (the contamination parameter is not configured), then the use of threshold algorithms can significantly improve the accuracy of anomaly detection. Threshold algorithms are noted, the use of which for the tasks of analyzing anomalies in energy consumption data, most often ensures an increase in accuracy. Threshold algorithms can be applied both to the results of individual anomaly detection algorithms and to the results of ensembles of algorithms obtained using various combination strategies.

Keywords: point anomaly detection, typical daily energy consumption profile, machine learning, threshold algorithm, ensembles of algorithms.

УДК 004.67
ББК 32.97

Статья представлена к публикации
членом редакционной коллегии В.В. Клочковым.

Поступила в редакцию 02.09.2024.
Дата опубликования 31.01.2025.