# 2.3. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ

## *Системный анализ, управление и обработка информации, статистика (технические науки) (2.3.1)*

УДК 004.89
DOI: 10.24160/1993-6982-2023-1-120-135

### Метод динамической классификации потоков данных с контролем области принятия решений

А.О. Гурина, Л.А. Гурина

Рассмотрена задача динамической классификации, предполагающая изменения в классифицируемых данных с течением времени. Для потоков данных, таких как данные компьютерной сети, данные с датчиков, банковские транзакции и др., характерны проблемы дрейфа, появления новых классов и аномалий. Изучены существующие методы классификации потоков данных. Отмечено, что единого и эффективного метода классификации, учитывающего одновременно проблемы обнаружения аномалий, дрейфа и адаптации модели к новым данным, нет. Установлена важность контроля области принятия решений классификаторов для качественного решения задачи. Предложен метод динамической классификации на основе масштабируемого ансамбля автокодировщиков с контролируемой с помощью критерия *EDCAP* областью принятия решений. Свойства автокодировщика использованы для решения проблем обнаружения дрейфа, аномалий и новых классов. Автокодировщики ансамбля обучены распознавать один класс. На основе критерия *EDCAP* проконтролирован размер области распознавания каждого автокодировщика. Результат классификации основан на анализе ответов всех участников ансамбля. При обнаружении данных нового класса ансамбль масштабируется путем добавления нового автокодировщика. При обнаружении дрейфа дообучаются лишь соответствующие автокодировщики. Выполнено сравнение качеств предлагаемого динамического классификатора и инкрементного алгоритма на основе адаптивного дерева Хёфдинга. Продемонстрированы преимущества предлагаемого метода на примере синтетического потока данных, включающего дрейф, новый класс и аномалии.

*Ключевые слова:* динамическая классификация, дрейф концепта, обнаружение аномалий, автокодировщик, ансамбль, инкрементное, онлайн- и машинное обучение, показатели качества классификации.

*Для цитирования:* Гурина А.О., Гурина Л.А. Метод динамической классификации потоков данных с контролем области принятия решений // Вестник МЭИ. 2023. № 1. С. 120—135. DOI: 10.24160/1993-6982-2022-5-120-135.

### A Data Stream Dynamic Classification Method with Control of the Decision-making Area

A.O. Gurina, L.A. Gurina

A dynamic classification problem in which a change with time in the data classified is assumed is considered. Data streams, such as computer network data, sensor data, bank transactions, etc., are characterized by problems of data drift, the emergence of new classes, and anomalies. The existing data streams classification methods are analyzed. It is pointed out that there is no a single and effective classification method

that would simultaneously take into account the problems of anomaly detection, drift, and model adaptation to new data. The importance of controlling the decision-making area of classifiers for obtaining a high-quality solution of the problem is noted. A dynamic classification method based on a scalable ensemble of autoencoders with a decision-making area controlled using the *EDCAP* criterion is proposed. The autoencoder properties are used to solve the problems of detecting drift, anomalies and new classes. The ensemble autoencoders were trained to recognize a single class. Based on the *EDCAP* criterion, the size of the recognition area of each autoencoder was controlled. The classification result is based on analyzing the responses of the ensemble's all autoencoders. When a new class of data is detected, the ensemble is scaled by adding a new autoencoder. When a drift is detected, only the corresponding autoencoders are retrained. The qualities of the proposed dynamic classifier and an incremental algorithm based on an adaptive Hoeffding tree are compared. The advantages of the proposed method are demonstrated on the example of a synthetic data stream that includes drift, a new class, and anomalies.

*Key words:* dynamic classification, concept drift, anomaly detection, autoencoder, ensemble, incremental online and machine learning, classification quality indicators.

*For citation:* Gurina A.O., Gurina L.A. A Data Stream Dynamic Classification Method with Control of the Decision-making Area. Bulletin of MPEI. 2023;1:120—135. (in Russian). DOI: 10.24160/1993-6982-2023-1-120-135.

**Введение**

Термин «динамическая классификация» в разных источниках трактуют по-разному. Изначально динамическую классификацию упоминали в контексте изменений в классифицируемых данных с течением времени, называемых концептуальным дрейфом.

Впервые проблема дрейфа, как изменения распределения классифицируемых данных, была обнаружена в реальной практической задаче М. Кубатом в 1992 г. [1 — 3]. Дрейф относят как к входным данным [4], так и к целевой функции, которую модель обучена предсказывать.

В зависимости от скорости изменения концепций известно несколько типов дрейфа [4, 5]:

- постепенный;
- периодический или циклический;
- внезапный или резкий.

Заметим, что, если периодический дрейф проходит постепенным образом, его также можно отнести к категории постепенного. При этом в случае резкого дрейфа задача сопоставления резко дрейфующего класса с исходным без участия эксперта является нетривиальной. В таком случае решением проблемы может стать отнесение резко дрейфующего класса к новому классу.

В настоящей работе под динамической классификацией понимается классификация, предусматривающая не только известные в настоящее время классы объектов (статическая классификация), но и их возможное развитие в будущем, будь то постепенные изменения признаков классифицируемых объектов (постепенный дрейф входных данных) или количества классов объектов (появление нового класса или резкий дрейф известного класса).

В настоящее время динамическая классификация чрезвычайно актуальна [6, 7], например, для приложений электронной коммерции, почтовых систем, социальных и компьютерных сетей, оборудования, обнаружения мошенничества и других, в которых генерируется огромное количество непрерывных и бесконечных данных, называемых потоком. Реализация необходимого управления и классификации наблюдений в потоке осложнена характерной проблемой дрейфа, приводящей к снижению производительности классификационной модели.

Большинство технических и информационных систем — многорежимны, и дрейф возникает из-за постепенного износа, либо накопления незначительных изменений, что остается незаметным на протяжении длительного времени. Тем не менее, процессы изменения режимов и характеристик функционирования в некоторых случаях могут быть значительными, например, в случае модернизации системы. Также на режимы работы технических систем влияют внешние возмущающие факторы. В информационных системах дрейф характеристик может быть обусловлен вмешательством человека-оператора, ошибками в реализации систем, изменением конфигурации программного и аппаратного обеспечения, а также деструктивными действиями вредоносных программ.

Другая проблема классификации потоков данных заключается в том, что обучающая выборка имеет ограниченный размер, и некоторые образы известных классов могут в неё не попасть. Кроме того, динамика характерна для задач обнаружения новизны, аномалий и одноклассовой классификации, поскольку понятие нормального поведения развивается, и текущее понятие такого поведения может быть недостаточно репрезентативным в будущем. Для корректного разрешения данной ситуации метод динамической классификации должен корректно опознавать такие образы и включать их в обучающую выборку для уточнения классификатора.

При обработке потока данных следует учитывать возможность аномалий, которые могут возникать в системе из-за кибератак, сбоев оборудования, ошибок и т. д. Во многих прикладных задачах крайне важно не допустить пропуска таких аномалий и отнесения их к целевому классу.

Разработка классификаторов для систем с изменяющимися характеристиками представляется сложной задачей, метод решения которой должен обладать как минимум следующими свойствами:

- динамической адаптацией алгоритма и области принятия решений к новым данным;
- обнаружением аномалий.

Поскольку результаты классификации часто служат для мониторинга, управления или принятия решений,

то метод эффективной классификации в динамических условиях может существенно повысить уровень автоматизации и качество решения целевой задачи, что имеет большое практическое значение.

**Обзор литературы**

Решением задач классификации в условиях дрейфа ещё с 1990-х гг. занимались такие исследователи, как М. Кубат [1], Г. Уидмер [3], Г. Нахаизаде [8], А. Цымбал [9], А.В. Жуков [10], Д.Н. Сидоров [10], В.А. Гимаров [11] и другие.

В настоящее время для решения задач динамической классификации используют адаптированные и специально разработанные методы машинного обучения. Адаптированные подходы [10, 12] расширяют возможности традиционных статических методов машинного обучения и включают механизмы обнаружения дрейфа. Чаще всего, чтобы обнаружить дрейф любого типа, контролируют текущую точность модели. Если точность уменьшается, то это означает, что модель становится неактуальной и нуждается в обновлении [13 — 15]. Также можно контролировать изменения в статистических свойствах самих данных [16, 17], в том числе, находить изменения в данных непараметрическими методами [18]. В [19] предложено использовать вариант метода кумулятивных сумм (исходно — метод обнаружения разладки) для обнаружения дрейфа. Перспективным инструментом поиска разных типов дрейфа считается одноклассовый классификатор, например, автокодировщик.

Автокодировщик, обученный должным образом на одной части потока данных, может быть использован для отслеживания возможных изменений в распределении данных в последующем потоке. Изменения анализируют путем мониторинга изменений ошибки реконструкции. Кроме того, экспериментальные результаты показывают, что детектор на основе автокодировщика способен обрабатывать различные типы дрейфа [20].

Обычно, после того, как дрейф обнаружен, классификатор переобучается на текущих данных в предположении, что они лучше описывают актуальное распределение данных.

Известны подходы, в которых механизм адаптации к дрейфу встроен в сам алгоритм классификации потока данных. Их делят на три основные группы: подходы, основанные на инкрементном обучении [21, 22], подходы, основанные на скользящем окне [23], и ансамблевые методы [24 — 26].

Анализ литературы показал, что наиболее эффективные методы решения проблемы дрейфа — ансамблевые методы и инкрементное обучение моделей [27, 28]. Рассмотрим их чуть подробнее.

В подходах, основанных на инкрементном обучении, процесс обучения происходит всякий раз, когда появляются новые примеры. Методы, включающие такую стратегию обучения, позволяют адаптировать границу принятия решения классификатора к изменениям во входящих данных.

Наиболее популярным методом классификации с обучением на потоке данных в реальном времени является адаптивное дерево Хёффдинга, учитывающее дрейф в данных (Hoeffding Adaptive Tree Classifier) [29]. Адаптивное дерево Хёффдинга [30] использует адаптивные окна (*ADWIN*) [23] для мониторинга производительности ветвей в дереве и замены их новыми ветвями при снижении точности, если точность новых ветвей выше.

Отмечается, что многие известные алгоритмы классификации можно адаптировать благодаря инкрементному обучению. Так, в [22] использован модифицированный метод взвешенного SVM одного класса, дополненный принципами инкрементного обучения и забывания. Свойства одноклассовой классификации могут служить для решения проблем дисбаланса классов и обнаружения дрейфа, поэтому одноклассовая классификация с инкрементным обучением является перспективным направлением исследований в области классификации потоков данных с дрейфом [22, 31].

Основной недостаток подобных подходов — риск обучения на аномалиях, вероятность появления которых высока во многих реальных средах (промышленность, компьютерные системы и сети, банковские транзакции и т. д.).

Кроме того, для инкрементного обучения характерно постепенное забывание старых данных, так называемая проблема катастрофического забывания. Авторы [32] в своей работе 2018 г. продемонстрировали, что проблема катастрофического забывания в парадигме инкрементного обучения не была решена. Такая особенность может привести к некорректной классификации при появлении в потоке исходных данных известного класса. Смягчить данную проблему можно, используя ансамбль базовых классификаторов и добавляя новый классификатор для хранения новых знаний [33].

Наиболее популярный развивающийся метод обработки дрейфа концепций в потоках данных — использование ансамбля классификаторов. Для принятия решения в таком подходе результаты классификации базовых участников ансамбля анализируются и объединяются для определения окончательного результата классификации. Способ объединения часто называют правилами слияния. Они чаще всего основаны на стратегии взвешивания или подсчете результатов базовых классификаторов. Примером может служить гетерогенный адаптивный ансамблевый классификатор с динамической схемой взвешивания, основанной на разнообразии его базовых классификаторов [25]. Данная модель ансамбля с адаптацией включает такие базовые классификаторы как наивный Байес, k-NN, деревья решений.

Известны отдельные способы решения указанных проблем динамической классификации: классификатор с адаптацией к новым данным с дрейфом *MLAW*

[24] и классификатор потоковых данных с дрейфом на основе Micro-Cluster Nearest Neighbour (MC-NN) [34].

Для борьбы с проблемой дрейфа применяют принцип динамического обучения ансамбля [35], заключающийся в разделении большого потока данных на небольшие блоки и независимом обучении классификаторов на отдельных блоках. Считается, что большой блок данных более надежен, в то время как небольшой блок лучше адаптируется к изменению данных. Подобная декомпозиция приводит к упрощению архитектур, сокращению времени обучения и повышению производительности и обобщающих свойств [33].

В [5] проанализированы модели динамического ансамблевого обучения, отмечена их способность работать как с неограниченно растущими объемами данных, так и с проблемами дрейфа концепций при интеллектуальном анализе потоков данных.

Сравнение инкрементного обучения и ансамблевого обучения с точки зрения эффективности решения проблемы дрейфа показало, что ансамблевый подход более стабилен и позволяет лучше адаптироваться к дрейфу [5].

Известны подходы, объединяющие обе стратегии [36, 37]. Например, в работе [36] 2021 г. для адаптации к проблеме дрейфа, как постепенного, так и внезапного, в потоковых данных предложен инкрементный ансамбль одноклассовых классификаторов. Модель оценивается с использованием реальных наборов данных и демонстрирует точность более 80%.

К недостаткам ансамблевого подхода следует отнести:

- проблему выбора набора разнообразных базовых классификаторов;
- проблему выбора размера блока данных, влияющего на производительность алгоритма;
- проблему выбора весовых значений для различных классификаторов ансамбля, напрямую влияющих на точность классификации;
- необходимость заново обучать весь ансамбль при появлении новых данных;
- проблему определения и исключения устаревших данных.

Многообразие существующих подходов динамической классификации не позволяет выявить однозначно лучший, поскольку каждый из них имеет различные достоинства и недостатки.

Отметим общий недостаток многих классификаторов — после обучения они делят пространство признаков на открытые области классов, что делает возможным отнесение аномалий к целевому классу. Разработка метода классификации, позволяющего строить компактные области классов, а также наличие показателей качества, оценивающих соответствие областей принятия решений и целевых классов, минимизировала бы риск неправильной классификации за пределами обучающей выборки.

Для решения обозначенных проблем наиболее целесообразно и перспективно применение ансамбля, включающего нейросетевые автокодировщики для детектирования дрейфа, новых классов и аномалий. В [38] предложен способ динамической классификации на основе масштабируемого ансамбля автокодировщиков, формирующий компактные области принятия решений в пространстве признаков, а также позволяющий управлять их величиной для точной классификации и выявления аномалий. Такой способ совместно с критерием качества *EDCAP* [39, 40] позволяет строить классификаторы, близкие к идеальным даже в пространстве высокой размерности, когда визуализация областей принятия решений затруднена.

Рассмотрим методологию динамической классификации на основе масштабируемого ансамбля автокодировщиков с контролируемой с помощью критерия *EDCAP* областью принятия решений.

**Описание метода динамической классификации**

Метод динамической классификации является расширением метода статической классификации [40] для обеспечения возможности классификации в условиях появления дрейфа, новых классов, аномалий, а также адаптации классификатора к новым данным.

В основе метода лежит ансамбль нейросетевых автокодировщиков, каждый из которых позволяет оценивать близость входных образов к известным классам и применяется в качестве детектора дрейфа, данных новых классов и аномалий.

Автокодировщик — многослойная нейронная сеть прямого распространения со специальной архитектурой в форме бабочки (рис. 1): входной и выходной слои должны иметь одинаковый, а промежуточный слой или слои — меньший размеры.

Автокодировщик обучается с помощью обратного распространения ошибки с тем, чтобы на выходе воспроизводить те же данные, что и на входе. Такая особенность позволяет условно разделить его архитектуру на кодер и декодер. На этапе обучения автокодировщика и по мере снижения размерности промежуточных слоёв кодера выявляется избыточность в поступающих на вход данных. В самом «узком» слое кодера обеспечивается сжатие данных до главных закономерностей, из которых в декодере по мере повышения размерности следующих промежуточных слоёв восстанавливается входной пример на выходном слое.

Восстановленные на выходном слое автокодировщика данные называют реконструкцией, а ошибку восстановления — ошибкой реконструкции. Обучение прекращается тогда, когда автокодировщик восстанавливает примеры обучающей выборки на выходном слое с требуемой минимальной ошибкой.

Применимость автокодировщика для обнаружения аномалий, новизны, дрейфа многократно подтверждена [20, 38, 41].

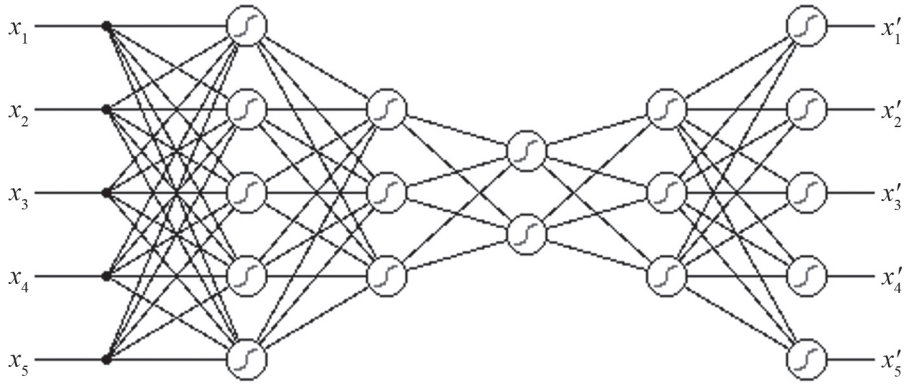



Рис. 1. Пример архитектуры автокодировщика

В предлагаемом способе автокодировщик используется для оценки близости поданных на вход примеров к обучающим данным $X_T$ следующим образом.

Обучение автокодировщика на обучающем множестве $X_T$ проходит по критерию минимизации ошибки реконструкции:

$$Training\{AE, X_T\} : \sum_{x \in X_T} RE(x) \to \min.$$

Результатом работы автокодировщика $AE$ над входным примером $x$ является восстановленный пример $\tilde{x}$. Степень близости входного примера $x(x_1, x_2, \ldots, x, \ldots, x_n)$ к обучающим данным определяется величиной ошибки реконструкции $RE(x)$, рассчитываемой по формуле

$$RE(x) = \sqrt{\sum_{j=1}^{n} \left(x_j - \tilde{x}_j\right)^2}, \quad \text{где} \quad \tilde{x} = AE(x).$$

Чем ближе $RE(x)$ к нулю, тем точнее автокодировщик восстанавливает входной вектор, и тем достовернее гипотеза о том, что входной вектор принадлежит целевому классу. Для обнаружения какой-либо новизны в данных устанавливается пороговое значение $RE_{th}$ как максимальное значение среди ошибок реконструкции, рассчитанных для примеров обучающего набора:

$$RE_{th} = \max_{x \in X_T} RE(x). \qquad (1)$$

Пороговое значение ошибки реконструкции можно интерпретировать как границу класса в пространстве признаков. Величина ошибки реконструкции $RE(x)$ рассматривается как метрика близости примера $x$ к границе класса, определяемой $RE(x) \le RE_{th}$. Таким образом, если $RE(x) > RE_{th}$, то пример находится за границей класса.

Задание порога ошибки реконструкции $RE_{th}$ позволяет построить решающее правило одноклассового классификатора:

$$CL^1(x) = \begin{cases} 0, \ RE(x) > RE_{th}; \\ 1, \ RE(x) \le RE_{th}. \end{cases}$$

В задаче динамической классификации важно различать два случая: входной пример располагается вблизи внешних границ класса или далеко от неё. Из общих соображений следует, что появление новых примеров вблизи внешней границы класса с течением времени может указывать на постепенный дрейф класса. В таком случае имеет смысл дообучить автокодировщик с учетом вновь поступивших примеров с включением вновь полученных выборок в обучающий набор.

При этом появление нового примера, находящегося на значительном удалении от границы класса ($RE(x) >> RE_{th}$), означает появление нового класса или аномалии. В таких случаях дообучение данного автокодировщика не требуется.

Чтобы отличить дрейф данных от других случаев классификации, вводится коэффициент пропорционального расширения границы $k_{drift}$. Он должен быть положительным числом больше единицы и может быть выбран эмпирически. Тогда значение $k_{drift} \cdot RE_{th}$ определяет допустимую для дрейфа данных дополнительную границу класса. Следует отметить, что форма внешней границы области дрейфа может быть более сложной, чем простое расширение границы класса в пространстве признаков объекта.

Таким образом, используя один автокодировщик, обученный на примерах определенного класса, а также значения $RE_{th}$ и $k_{drift}$ в зависимости от значения $RE(x)$, можно определить вновь поступающий пример $x$ как пример:

- целевого класса;
- целевого класса с дрейфом;
- нового класса или аномалии.

На рисунке 2 продемонстрировано применение введенной логики для таких случаев.

В таблице 1 отражено формальное описание условий, соответствующих результатов классификации и вид адаптации классификатора.

Различить случаи нового класса и аномалии можно с помощью дополнительного коэффициента расширения границы класса $k_{anom}$.

В случае одноклассовой классификации пример $X_3$ является аномалией, если ошибка реконструкции при-

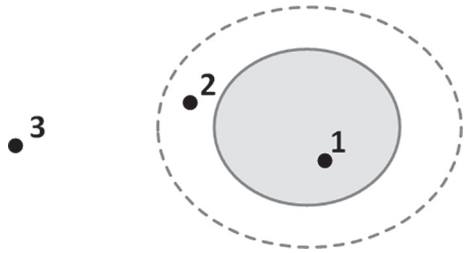



Рис. 2. Детектирование случаев классификации с использованием одного автокодировщика:

— — $RE_{th}$; - - - - — $k_{drift}{\cdot}RE_{th}$

*Таблица 1*

**Детектирование случаев классификации с использованием одного автокодировщика**

| Условие | Результат | Адаптация |
|---|---|---|
| $RE_{X1} \le RE_{th}$ | Целевой класс | Не требуется |
| $RE_{th} < RE_{X2} \le k_{drift}{\cdot}RE_{th}$ | Целевой класс с дрейфом | Дообучение |
| $RE_{X3} > k_{drift}{\cdot}RE_{th}$ | Новый класс/ аномалия | Масштабирование/ информирование |

мера $REX_3$ находится выше некоторой границы, которая может быть определена как $k_{anom}{\cdot}RE_{th}$. Значение $k_{anom}$ выбирается таким, чтобы значение $k_{anom}{\cdot}RE_{th}$ устанавливало границу, за пределами которой в пространстве признаков могут быть только аномалии. Подобрать $k_{anom}$ можно, исходя из предельно допустимых значений признаков классифицируемых объектов и критерия оценки области принятия решений *EDCAP*. Таким образом, если $REX_3 > k_{anom}{\cdot}RE_{th}$, то данный пример классифицируется как аномалия.

Можно рассматривать $k_{drift} = k_{anom}$ для поиска аномалий за границами допустимой области дрейфа целевого класса. Однако в некоторых задачах динамической классификации необходимо отличать случай нового класса от существенных аномалий.

В большинстве приложений ошибка реконструкции примера нового класса $RE(x)$ превысит $k_{drift}{\cdot}RE_{th}$, но не будет также велика, как в случае с аномалиями. Тогда, если $k_{anom}{\cdot}RE_{th} < REX_3 \le k_{anom}{\cdot}RE_{th}$, то с большой вероятностью пример $X_3$ будет представителем нового класса.

Случай обнаружения аномалии требует только соответствующего результата классификации и своевременного информирования. При этом случай определения нового класса требует масштабируемости классификатора, заключающейся в синтезе и обучении новых автокодировщиков и объединении всех автокодировщиков в ансамбль.

Рассмотрим подход, расширенный на случай двух и более классов с применением ансамбля автокодировщиков.

Пусть даны $M$ классов, заданные в векторном пространстве $X \subseteq R^n$ и описанные множествами примеров обучающей выборки:

$$X_T^{(i)} \subset X, \quad i = \overline{1, M}.$$

Для удобства введем обозначение множества обучающих примеров всех классов:

$$X_T^0 = \bigcup_{i=1}^{M} X_T^{(i)}.$$

Пусть существуют также тестовые множества для каждого из классов:

$$X_{Test}^{(i)} \subset X, \quad i = \overline{1, M}.$$

Для решения задачи динамической бинарной и многоклассовой классификации (далее — классификации) с обнаружением аномалий использован масштабируемый ансамбль автокодировщиков с рассмотренной выше логикой обнаружения целевого класса, дрейфа, новых классов и аномалий.

Пусть обучающая выборка состоит из примеров $M$ известных классов: $C_1$, $C_2$, …, $C_M$. Тогда требуется построение ансамбля из $M + 1$ автокодировщиков, причем каждый из $M$ автокодировщиков $AE_1$, $AE_2$, …, $AE_M$ обучается распознавать примеры одного из известных классов $C_1$, $C_2$, …, $C_M$. Кроме того, синтезируется и включается в ансамбль общий автокодировщик $AE_0$, обучающийся на всех примерах обучающей выборки $X_T^0$. Для обученных автокодировщиков определяют пороговые значения $RE_{th_i}$, где $i = 0, 1, \ldots, M$ по (1).

Установленные таким образом пороговые значения ошибки реконструкции для каждого автокодировщика позволяют очертить в пространстве признаков $M$ границы областей известных классов $C_1$, $C_2$, …, $C_M$, а также область $C_0$, охватывающую все известные классы (рис. 3).

Для обнаружения дрейфа в данных требуется задать коэффициент $k_{drift}$, который, в общем случае, может быть подобран для каждого класса. Для простоты описания метода использован один и тот же $k_{drift}$ для всех классов. С помощью значения $k_{drift}{\cdot}RE_{th_i}$ отслеживается дрейф данных каждого $C_i$ класса, что позволяет верно классифицировать входные данные, а также своевременно переобучать соответствующий $AE_i$ автокодировщик, где $i = 0, 1, \ldots, M$. Дообучение происходит в момент, когда дрейфующих данных поступает больше половины имеющегося обучающего набора. Если же дрейф класса выходит за границу области известных классов, характеризующуюся $RE_{th0}$, то также требуется дообучение $AE_0$ с учётом новых данных. В общем случае, правила для дообучения автокодировщиков должны определяться для предметной области.

Если классы в пространстве признаков расположены близко друг к другу, как на рис. 3, то возможен случай, когда входной пример оказывается в области допустимого дрейфа для нескольких классов ($i = k_1, k_2, \ldots$) одновременно, т. е. ошибка реконструкции входного примера $RE(x)$ не превышает пороговых значений $k_{drift}{\cdot}RE_{th}$, установленных для нескольких автокодировщиков ($AE^{(i)}$). В таком случае входной пример будет

отнесен к тому классу $k_j$, для которого значение $RE(x)$ относительно порога минимально согласно формуле

$$k_j = \arg \min_{i=k_1,k_2,\ldots} RE^{(i)} \Big/ RE_{th}^{(i)} .$$

В рассмотренных ранее случаях предполагалось, что известные классы могут расширяться. Стоит проанализировать случай, когда детектируются постепенный, периодический или внезапный дрейфы класса, тогда с течением времени некоторые данные, на которых автокодировщик был обучен изначально, станут неактуальны для данного класса и должны быть забыты. Это решается полным переобучением автокодировщика, когда необходимый объем актуальных обучающих данных будет собран.

Рассмотрим случай, когда новый пример не распознается ни одним из участников ансамбля, тогда, для различия случаев появления нового класса или аномалии, используется пороговый критерий $k_{anom} \cdot RE_{th0}$. Если ошибка реконструкции $RE$ для нового примера не превышает значение $k_{anom} \cdot RE_{th0}$, то данный пример классифицируют как пример нового класса. В этом случае следует создать и обучить новый автокодировщик $AE_{M+1}$, а также переобучить $AE_0$ с учетом примеров нового класса. Если ошибка реконструкции $RE$ для нового примера превышает $k_{anom} \cdot RE_{th0}$, то этот пример классифицируют как аномалию. В случае обнаружения аномалии рекомендуется сгенерировать сигнал тревоги. Для множества предметных областей это имеет критически важное значение.

Возможные случаи классификации продемонстрированы на рис. 3 и формально описаны с помощью условий в табл. 2.

Отметим, что метод также включает оценку качества и оптимизацию областей принятия решений каждого участника ансамбля в соответствии с критерием *EDCAP*. Ансамбль автокодировщиков готов к использованию только тогда, когда каждый участник ансамбля по критерию *EDCAP* соответствует требуемому для конкретной задачи уровню качества и устойчивости к ошибкам классификации, которые могут быть вызваны аномалиями или состязательными атаками.

Таким образом, ансамблем автокодировщиков в настоящей работе называется комбинация автокодировщиков, каждый из которых обучен распознавать примеры одного из целевых классов, за исключением общего автокодировщика, обученного распознавать близость примера к области целевых классов. Результат классификации ансамблем определяется на основании анализа ответов каждого участника ансамбля

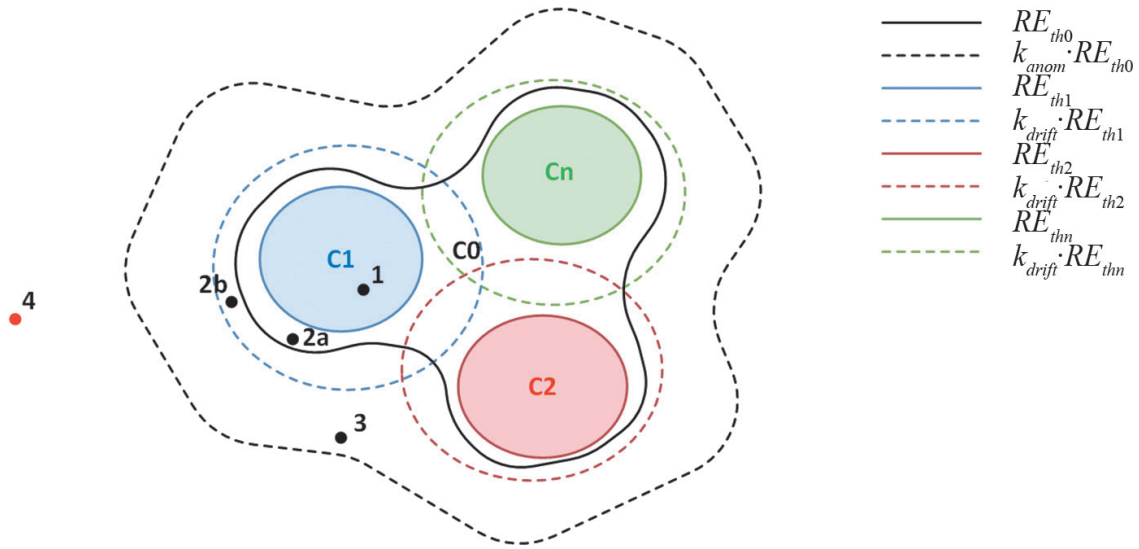



Рис. 3. Детектирование случаев классификации с использованием ансамбля автокодировщиков

*Таблица 2*

**Детектирование случаев классификации с использованием ансамбля автокодировщиков**

| Условие | Результат | Адаптация |
|---|---|---|
| $RE_{X1} \le RE_{th(1\|2\|\ldots M)}$ | Целевой класс | Не требуется |
| $RE_{th(1\|2\|\ldots M)} < RE_{X2a} \le k_{drift} \cdot RE_{th(1\|2\|\ldots M)}$, $RE_{X2a} \le RE_{th0}$ | Целевой класс с дрейфом | Дообучение $AE_{(1\|2\|\ldots M)}$ |
| $RE_{th(1\|2\|\ldots M)} < RE_{X2b} \le k_{drift} \cdot RE_{th(1\|2\|\ldots M)}$, $RE_{th0} < RE_{X2b} \le k_{anom} \cdot RE_{th0}$ | | Дообучение $AE_{(1\|2\|\ldots M)}$ и $AE_0$ |
| $RE_{X3} > k_{drift} \cdot RE_{th(1\|2\|\ldots M)}$, $RE_{th0} < RE_{X3} \le k_{anom} \cdot RE_{th0}$ | Новый класс | Синтез $AE_{M+1}$, дообучение $AE_0$ |
| $RE_{X4} > k_{anom} \cdot RE_{th0}$ | Аномалия | Информирование |

согласно установленным правилам принятия решения. Масштабирование ансамбля, т. е. синтез новых автокодировщиков и включение их в ансамбль, происходит автоматически при обнаружении достаточного количества примеров нового класса.

Метод динамической классификации на основе масштабируемого ансамбля автокодировщиков с контролируемой с помощью критерия *EDCAP* областью принятия решений, названного *SEAEs* (Scalable Ensemble Of AutoEncoders), опубликован в [38].

Опишем предлагаемый метод динамической классификации на основе SEAEs в виде алгоритма.

Пусть обучающие $X_T^{(i)}$ и тестовые $X_{Test}$ данные прошли необходимые процедуры предобработки и нормализации.

Разработаем классификатор $SEAEs(x)$, который бы для любого $x \in X$ сообщал метку класса $i$, если $x$ похож на примеры из $X_T^{(i)}$, или 0 — если $x$ не похож ни на один из классов, или $M + 1$ — если $x$ представляет собой новый класс.

Процедура синтеза классификатора $SEAEs(x)$:

1. Задание архитектуры $AE^0, AE^{(i)}, i = \overline{1, M}$.

2. Обучение нейросетевых автокодировщиков:

   a. Обучение $Training\{AE^0, X_T^0\}$.

   b. Обучение $Training\left\{AE^{(i)}, X_T^{(i)}\right\}$, $i = \overline{1, M}$.

3. Расчет величин, определяющих принятие решений:

   a. Расчет порога для одноклассового классификатора $CL^0(x)$ на основе $AE^0$.

   b. Расчет порогов $RE_{th}^{(i)} = \max\limits_{x \in X_T^{(i)}} RE^{(i)}(x)$ для одноклассовых классификаторов $CL^{(i)}(x)$ на основе $AE^{(i)}, i = \overline{1, M}$.

   c. Выбор $k_{drift}$ и расчет порогов обнаружения дрейфа $k_{drift} \cdot RE_{th}^{(i)}$, $k_{drift} \cdot RE_{th}^0$.

   d. Выбор $k_{anom}$ и расчет порогов обнаружения аномалий $k_{anom} \cdot RE_{th}^{(i)}$, $k_{anom} \cdot RE_{th}^0$.

4. Расчет характеристик качества полученного классификатора с детализацией по классам:

   a. По критерию *EDCAP*: $\overline{Excess^{(i)}}$, $Deficit^{(i)}$, $Coating^{(i)}$, $Approx^{(i)}$, $Pref^{(i)}$, $i = \overline{1, M}$.

   b. По тестовому множеству $X_{Test}$: *Precision*, *Recall*, *Fscore.*

5. Анализ характеристик качества классификации отдельных автокодировщиков $AE^0, AE^{(i)}$, $i = \overline{1, M}$, корректировка их архитектуры, параметров обучения и повторение шагов 2 — 4 до получения необходимого уровня качества.

Для компактного изложения алгоритма обозначим $\min\limits_{i=\overline{1,M}}\left(RE_{th}^{(i)}\right)$ как $RE_{th}^{\min}$, а $\max\limits_{i=\overline{1,M}}\left(RE_{th}^{(i)}\right)$ как $RE_{th}^{\max}$. Приведем алгоритм функционирования масштабируемого ансамбля автокодировщиков $SEAEs(x)$, реализующего метод динамической классификации.

1. Если $CL^0(x) = 0$, то пример $x$ находится за границей области известных классов и классифицируется как:

   a. аномалия, если $RE(x) > k_{anom} \cdot RE_{th}^{\min}$.

   b. новый класс, если $k_{drift} \cdot RE_{th}^{\min} < RE(x) \le k_{anom} \cdot RE_{th}^{\min}$.

   c. целевой класс $i$ с дрейфом, если $RE_{th}^i < RE(x) \le k_{drift} \cdot RE_{th}^i$, причем, если есть несколько таких $i$, то выбирается тот, у которого $RE(x)/RE_{th}^i$ минимально.

   d. целевой класс $i$, если $RE(x) \le RE_{th}^i$, причем, если есть несколько таких $i$, то выбирается тот, у которого $RE_{th}^i/RE(x)$ максимально.

2. Иначе если $CL^0(x) = 1$, то пример $x$ принадлежит области известных классов и классифицируется как:

   a. целевой класс $i$, если $RE(x) \le RE_{th}^i$, причем, если есть несколько таких $i$, то выбирается тот, у которого $RE_{th}^i/RE(x)$ максимально.

   b. целевой класс $i$ с дрейфом, если $RE_{th}^i < RE(x) \le k_{drift} \cdot RE_{th}^i$, причем, если есть несколько таких $i$, то выбирается тот, у которого $RE(x)/RE_{th}^i$ минимально.

   c. новый класс, если $k_{drift} \cdot RE_{th}^{\max} < RE(x) \le k_{anom} \cdot RE_{th}^{\max}$.

3. Если количество примеров нового класса $X^{(M+1)}$ больше, чем половина примеров одного из классов $X_T^i$, то дообучается общий автокодировщик $AE^0$ и синтезируется и добавляется в ансамбль новый автокодировщик $AE^{(M+1)}$.

4. Если обнаружено достаточное количество примеров дрейфующего класса $i$ для дообучения, то дообучается автокодировщик $AE^i$ и общий автокодировщик $AE^0$.

Для решения проблем аномалий в данном методе для каждого участника ансамбля оценивается и корректируется область принятия решения, создаваемая им после обучения. Кроме того, метод включает стратегии обнаружения дрейфа, новых классов и аномалий с помощью свойств автокодировщиков и адаптации классификатора к новым данным.

Реализованное в алгоритме частичное дообучение участников ансамбля на актуальных данных позволяет быстрее адаптировать границу классов классификатора к новым данным. Предусмотренные в методе правила обнаружения примеров новых классов позволяют определять примеры классов, неизвестных из обучающей выборки. Тем не менее, при поиске новых классов желательна верификация результатов экспертом.

Таким образом, в предлагаемый метод интегрированы:

- оценка качества обученных моделей ансамбля по критерию *EDCAP*;
- механизмы обнаружения дрейфа, новых классов и аномалий с использованием свойств автокодировщиков;
- механизм адаптации классификатора к новым данным, включая масштабирование в случае появления новых классов и частичное дообучение — в случае новых классов и дрейфа данных.

Отметим, что дополнительно в методе может быть использован способ борьбы с проблемой отравления обучающих данных [42].

Предполагается, что разработанный метод сделает возможной достоверную классификацию потоков данных, расширив сферу применения нейросетевых классификаторов на сложные задачи, касающиеся, например, интеллектуальных систем поддержки принятия решений в технической диагностике или обнаружении компьютерных атак.

Для более подробного изучения свойств предложенного метода проведена экспериментальная апробация на синтетических двумерных данных, позволяющих визуализировать исходные данные и область принятия решений классификаторов.

**Экспериментальная апробация SEAEs**

Рассмотрим результаты, полученные при апробации предлагаемого метода динамической классификации с контролируемой по *EDCAP* областью принятия решения (*SEAEs*) на примере синтетических данных, включающих дрейф, новые классы и аномалии, а также результаты сравнения *SEAEs* с инкрементным алгоритмом на основе адаптивного дерева Хёффдинга (Hoeffding Adaptive Tree Classifier, *HAT*). Динамический классификатор *SEAEs* реализован с помощью библиотеки *keras*, а *HAT* — с помощью библиотеки *skmultiflow* [43].

В первый момент времени обучающие данные содержат данные только двух классов. Количество обучающих примеров $D = 400$. Два тестовых набора $T1$ и $T2$ построены таким образом, что дрейф данных происходит после обработки 400 примеров данных. Тестовый набор данных $T1$ дополнительно включает примеры нового класса. В тестовый набор данных $T2$ дополнительно входят примеры аномалий.

В начальный момент времени ансамбль *SEAEs* состоял из трех автокодировщиков: $AE^1$, $AE^2$, $AE^0$. В экспериментах для всех участников ансамбля использована типовая архитектура $NN_{2,3,7,4,7,3,2}$ и параметры обучения $4 \cdot 10^4$ эпох с помощью алгоритма обучения *ADAM* со скоростью обучения равной $\eta = 0{,}01$. Для реализации классификатора *SEAEs* параметры *kdrift* и *kanom* установлены на 15 и 100 при среднем уровне порога ошибки реконструкции, равном 0,05.

В первом эксперименте ансамбль *SEAEs* и *HAT* обучаются только на первых 400-х примерах данных. Поскольку рассматривается случай не пересекающихся классов, чтобы оценить качество обученного *SEAEs*, для каждого автокодировщика ансамбля были рассчитаны значения только четырех из пяти показателей качества: *Excess*, *Deficit*, *Coating*, *Approx* (табл. 3). Также для удобства сравнения в табл. 3 приведены интегральные значения оценки для *SEAEs*, *HAT* и эталонные значения критерия *EDCAP*.

Сравнение полученных значений для *SEAEs* с критерием *EDCAP* позволили убедиться, что в пространстве признаков нет больших областей, которые могут включать в себя аномалии. После успешной проверки

*Таблица 3*

**Результаты оценки качества классификаторов по критерию *EDCAP***

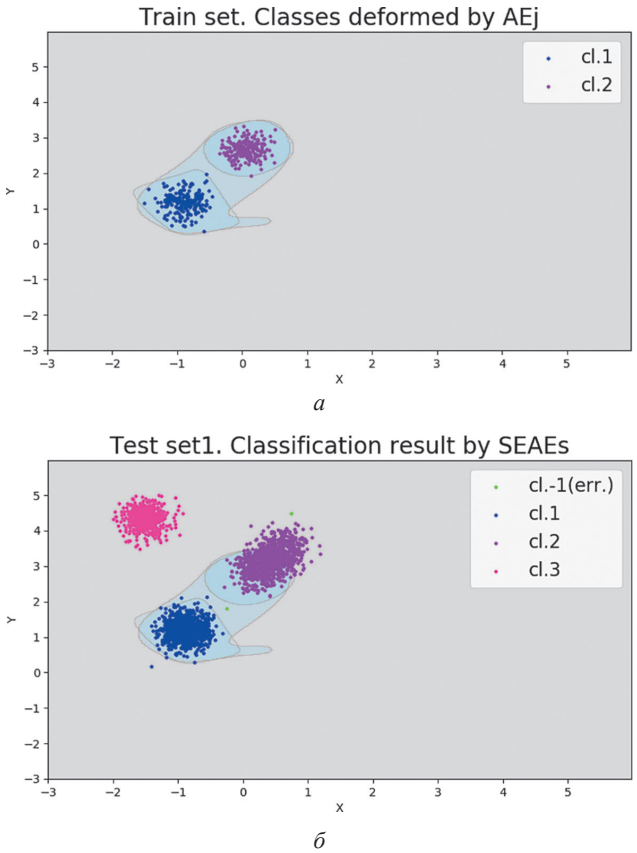
| Классификатор | Показатели качества | | | |
|---|---|---|---|---|
| | ***Excess*** | ***Deficit*** | ***Coating*** | ***Approx*** |
| *SEAEs* | 1,240 | 0,00 | 1,00 | 0,57 |
| *HAT* | 17,60 | 0,00 | 1,00 | 0,08 |
| Критерий *EDCAP* | 0,000 | 0,00 | 1,00 | 1,00 |

ансамбль *SEAEs* может быть использован для классификации новых данных.

Высокое значение показателя *Excess* и низкое значение *Approx* говорит о том, что модель *HAT* имеет открытые области принятия решений и уязвима для пропуска аномалий.

Протестируем обученные классификаторы на тестовом наборе данных $T1$, визуализируем области принятия решений, а также результаты классификации исследуемых классификаторов *SEAEs* (рис. 4) и *HAT* (рис. 5).

Примеры обучающего набора, а также границы классов, построенные автокодировщиками в пространстве признаков, даны на рис. 4, *а*. На рисунке 4, *б* продемонстрирован результат классификации тестового




*а*

*б*

Рис. 4. Результат обучения (*а*) и классификации тестового набора $T1$ (*б*) с использованием *SEAEs*

набора *T*1 с использованием классификатора *SEAEs*. Ошибки классификации выделены зеленым цветом. Видно, что ансамбль *SEAEs* с высокой точностью классифицировал дрейфующие данные класса 2 (cl.2) и данные нового класса (cl.3), которого не было в обучающей выборке.

Рассмотрим аналогичные результаты для классификатора *HAT*.

Из данных рис. 5, *а* следует, что область принятия решений, построенная в пространстве признаков обученным классификатором *HAT*, сильно отличается от областей обучающих данных: области классов не компактны, и аномалии, далеко отстоящие от примеров обучающей выборки, отнесены классификатором к целевому классу. В данном случае визуальный анализ подтверждает вывод, сделанный по рассчитанным характеристикам *EDCAP*. На рисунке 5, *б* показан результат *HAT* при классификации тестового набора *T*1. Видно, что при использовании данного классификатора новый класс или, например, резкий дрейф класса был бы некорректно распознан.

Качество классификации изучаемых классификаторов подтверждается традиционными характеристиками качества (см. табл. 4).

Рассчитанные значения традиционных характеристик качества показывают, что данная модель *SEAEs* превосходит инкрементный алгоритм *HAT*.

После того, как было собрано достаточное количество примеров новых данных, среди которых не было обнаружено аномалий, классификатор был дообучен. Пул классификаторов ансамбля *SEAEs* расширился за счет нового автокодировщика $AE^3$ для распознавания нового класса (cl.3). На обучающих данных класса 3 синтезирован и оценен новый автокодировщик той же архитектуры. Обучающая выборка для дообучения общего автокодировщика $AE^0$ состояла из актуальных данных, включая исходные, а также дрейф класса 2 и новый класс 3.

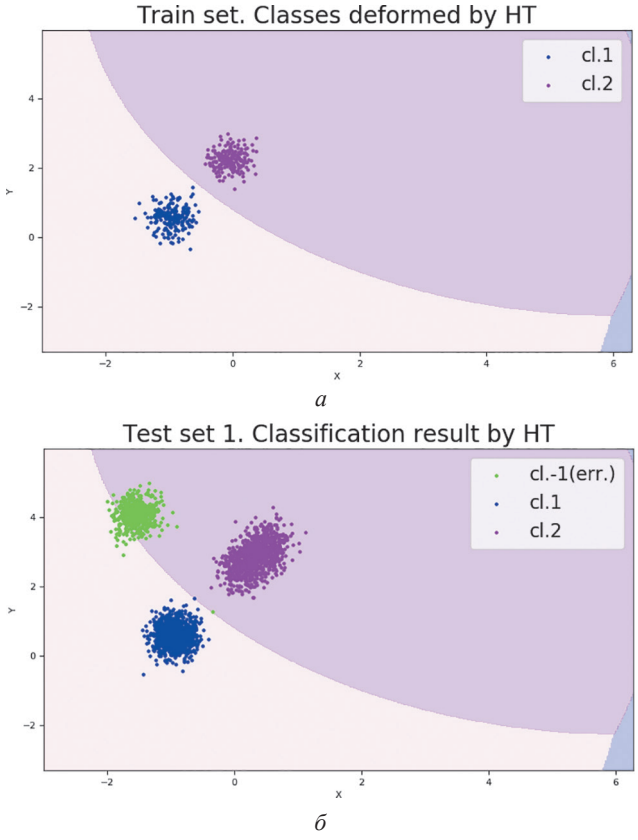
Инкрементный алгоритм *HAT* также был дообучен на том же наборе актуальных данных. После адаптации качество исследуемых классификаторов оценивали по критерию *EDCAP* (см. табл. 5).

Значения показателей для *SEAEs* демонстрируют, что область принятия решений незначительно увеличилась, что объясняется появлением области для определения нового класса и большим разнообразием в виду дрейфа данных. Для алгоритма *HAT* область принятия решений после дообучения стала более компактной, однако стала гораздо более уязвимой к аномалиям по сравнению с моделью *SEAEs*.

Покажем, как изменились области принятия решений и качество классификации тестового набора *T*2 после дообучения исследуемых классификаторов (рис. 6, 7).

Выводы, сделанные по анализу значений показателей качества *EDCAP*, хорошо согласуются и подтверждены визуально (см. рис. 6, *а*, *б*). На рисунке 6, *б* показаны результаты классификации для тестового набора *T*2, который, помимо дрейфа всех классов, включает и аномалии. Своевременно дообученный классификатор *SEAEs* успешно классифицировал данные с дрейфом и аномалиями.

Рассмотрим результат классификации тестового набора *T*2 с помощью инкрементного алгоритма *HAT* (см. рис. 7).




*а*

*б*

Рис. 5. Результат обучения (*а*) и классификации тестового набора *T*1 (*б*) с использованием *HAT*

*Таблица 4*

**Результаты оценки качества *SEAEs* и *HAT* по традиционным показателям**

| Классификатор | *Precision* | *Recall* | *Fscore* |
|---|---|---|---|
| *SEAEs* | 1,00 | 0,99 | 0,99 |
| *HAT* | 0,65 | 0,80 | 0,72 |

*Таблица 5*

**Результаты оценки качества классификаторов по критерию *EDCAP* (после дообучения)**

| Классификатор | Показатели качества | | | |
|---|---|---|---|---|
| | *Excess* | *Deficit* | *Coating* | *Approx* |
| *SEAEs* | 2,290 | 0,00 | 1,00 | 0,49 |
| *HAT* | 12,48 | 0,00 | 1,00 | 0,12 |

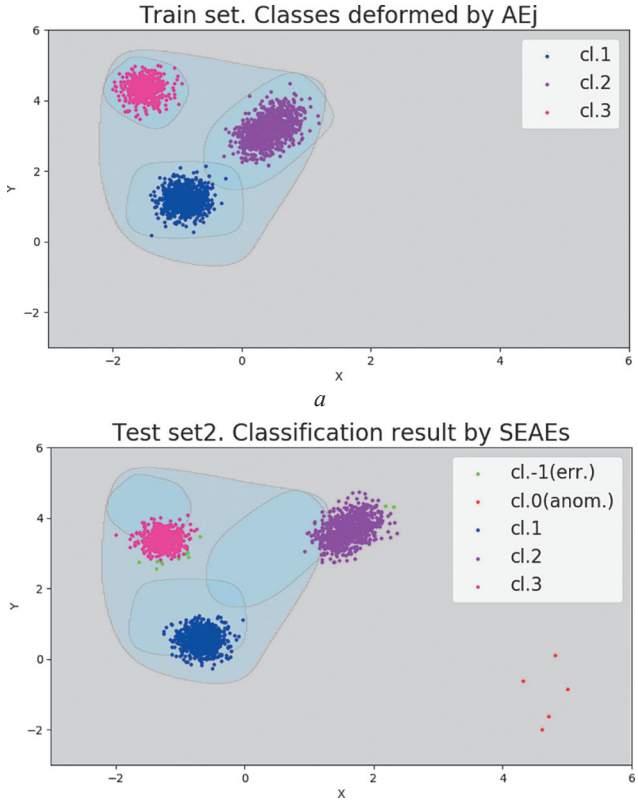



Рис. 6. Результат дообучения (*а*) и классификации тестового набора *T2* (*б*) с использованием *SEAEs*


Рис. 7. Результат дообучения (*а*) и классификации тестового набора *T2* (*б*) с использованием *HAT*

По данным рис. 7, *а* видно, что область принятия решений, скорректированная алгоритмом *HAT* после дообучения, осталась значительной за пределами обучающих данных. Из рисунка 7, *б* следует, что некомпактность области принятия решений алгоритма *HAT* привела к пропуску аномалий: они были отнесены ко второму классу, кроме того, очевидно, что использование алгоритма *HAT* после дообучения привело бы к некорректной классификации данных дрейфующего третьего класса (cl.3).

Выводы о качестве классификации, сделанные в результате визуального анализа, доказаны значениями традиционных показателей качества классификации (табл. 6).

Рассмотрим, как менялось качество классификации для инкрементного алгоритма *HAT* в разные моменты времени (рис. 8).

На рисунке 8 красной пунктирной линией отмечено снижение показателей качества алгоритма при появлении в потоке данных нового класса, дрейфа данных и аномалий. При этом, согласно визуальному анализу (см. рис. 4, 6) и традиционным критериям (см. табл. 4, 6), качество динамического классификатора *SEAEs* после первого обучения и дообучения оставалось высоким.

Таким образом, полученные экспериментальные результаты на синтетическом наборе данных подтвердили эффективность предложенного метода динамической классификации на основе масштабируемого ансамбля автокодировщиков *SEAEs* с контролируемой с помощью критерия *EDCAP* областью принятия решений в сравнении с инкрементным алгоритмом *HAT*, также учитывающим дрейф.

*Таблица 6*

**Результаты оценки дообученных классификаторов по традиционным показателям**

| Классификатор | *Precision* | *Recall* | *Fscore* |
|---|---|---|---|
| *SEAEs* | 1,00 | 0,99 | 0,99 |
| *HAT* | 0,78 | 0,85 | 0,81 |

## Обсуждение

Особенность подхода состоит в прямой взаимосвязи между количеством классов и количеством автокодировщиков в ансамбле, однако это необходимо для обеспечения динамической и достоверной классификации новых образов. Следовательно, для предложенного подхода характерна большая ресурсоемкость на этапе подготовки комплекса и его дообучения в задачах со значительным и постоянно растущим числом классов.

Отметим, что для адаптации классификатора к новым данным в случае дрейфа только одного класса необходимо дообучение только соответствующего этому классу участника ансамбля и общего автокодировщи-

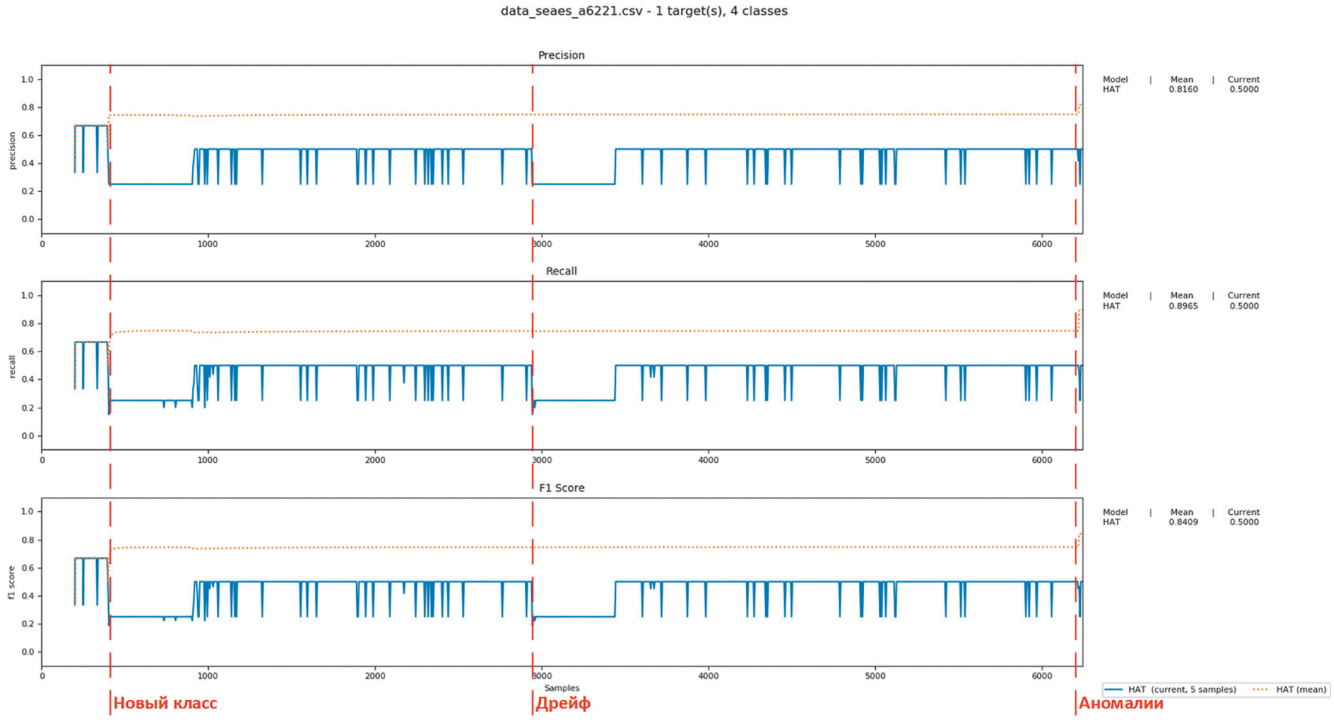



Рис. 8. График качества классификации синтетического потока данных, включающего новый класс, дрейф и аномалии алгоритмом *HAT*

ка, а в случае появления нового класса — требуется синтез нового участника ансамбля и дообучение общего автокодировщика. Такой подход существенно ослабляет требования к ресурсам в сравнении с подходами, где для адаптации многоклассового классификатора, реализованного одной сложной моделью или ансамблем моделей, необходимо полное переобучение всей модели на всех данных, что является избыточным.

К объективным недостаткам подхода можно отнести:

- ресурсоемкость, обусловленную обучением $M+1$ нейросетевых автокодировщиков для задачи с $M$ классами и вычислительно сложным в пространстве высокой размерности методом оценки качества *EDCAP*;
- появление новых классов, ведущее к добавлению новых автокодировщиков в ансамбль;
- необходимость выбора параметров $k_{drift}$ и $k_{anom}$ для достоверной классификации в условиях дрейфа новых классов и аномалий.

Отметим, что недостатки, связанные с ресурсоемкостью, могут быть устранены с помощью аппаратных средств (*GPGPU*).

Помимо этого, предлагаемый подход обладает рядом достоинств:

- повышает устойчивость к ошибкам классификации за счет применения критерия качества *EDCAP* и использования свойств автокодировщиков для контроля области принятия решений и детектирования дрейфа, новых классов и аномалий;
- риск обучения на аномалиях может быть сведен к минимуму, что важно для многих прикладных систем;
- объединяет в себе решение нескольких задач, ранее вычисляемых отдельными методами;
- параметров алгоритма классификации немного, и они имеют понятный смысл;
- ансамбль адаптируется только по необходимости и частично, что снижает требования к ресурсам.

Поскольку результаты экспериментов на синтетических двумерных данных доказали эффективность предложенного метода, дальнейшие исследования будут направлены на тестирование алгоритма *SEAEs* на реальных многомерных потоках данных и сравнение его эффективности с традиционными динамическими классификаторами.

## Заключение

Предложен метод динамической классификации на основе масштабируемого ансамбля автокодировщиков *SEAEs* с контролируемым с помощью критерия *EDCAP* качеством, обеспечивающий достоверную классификацию в условиях дрейфа данных, новых классов, аномалий и состязательных атак. Продемонстрирована эффективность предлагаемого метода по сравнению с инкрементным алгоритмом на основе адаптивного дерева Хёффдинга при решении специальной тестовой задачи классификации синтетического набора данных, включающего дрейф, аномалии и новый класс. Предложенный метод динамической классификации более предсказуем, позволяет повысить качество классификации и скорость адаптации классификатора к новым данным.

**Литература**

1. **Kubát M.** A Machine Learning-based Approach to Load Balancing in Computer Networks // Cybernetics and Systems. 1992. V. 23(3—4). Pp. 389—400.

2. **Widmer G., Kubát M.** Effective Learning in Dynamic Environments by Explicit Context Tracking // Proc. Machine Learning Conf. 1993. V. 667. Pp. 227—243.

3. **Widmer G., Kubát M.** Learning in the Presence of Concept Drift and Hidden Contexts // Machine Learning. 1996. V. 23(1). Pp. 69—101.

4. **Machine** Learning Monitoring. Pt. 5. Why You Should Care about Data and Concept Drift [Электрон. ресурс] https://evidentlyai.com/blog/machine-learning-monitoring-data-and-concept-drift (дата обращения 14.06.2022).

5. **Zang W. e. a.** Comparative Study between Incremental and Ensemble Learning on Data Streams: Case Study // J. Big Data. 2014. V. 1(5). Pp. 1—5.

6. **Žliobaitė I., Pechenizkiy M., Gama J.** An Overview of Concept Drift Applications // Big Data Analysis: New Algorithms for a New Society. N.-Y.: Springer, 2016. V. 16. Pp. 91—114.

7. **Webb G. I. e. a.** Understanding Concept Drift [Электрон. ресурс] https://www.researchgate.net/publication/315765198_Understanding_Concept_Drift (дата обращения 14.06.2022).

8. **Nakhaeizadeh G., Taylor C.C., Kunisch G.** Dynamic Supervised Learning: Some Basic Issues and Application Aspects // Classification and Knowledge Organization. Studies in Classification, Data Analysis, and Knowledge Organization. Berlin: Springer, 1997. Pp. 123—135.

9. **Tsymbal A.** The Problem of Concept Drift: Definitions and Related Work. Dublin: Trinity College, 2004. V. 106(2). P. 7.

10. **Жуков А.В., Сидоров Д.Н.** Модификация алгоритма случайного леса для классификации нестационарных потоковых данных // Вестник Южно-Уральского гос. ун-та. Серия «Математическое моделирование и программирование». 2016. № 4(9). С. 86—95.

11. **Гимаров В.А.** Методы и автоматизированные системы динамической классификации сложных техногенных объектов: автореф. дис. … доктора техн. наук. М.: Изд-во Российского химико-технолог. ун-та им. Д.И. Менделеева, 2004.

12. **Nakhaeizadeh G., Taylor C., Lanquillon C.** Evaluating Usefulness for Dynamic Classification // KDD. 1998. V. 1. Pp. 87—93.

13. **Gama J. e. a.** Learning with Drift Detection Lecture Notes in Computer Science // Proc. Brazilian Symp. Artifical Intelligence. 2004. V. 3171. Pp. 286—295.

14. **Baena-Garc M. e. a.** Early Drift Detection Method // Proc. IV Intern. Workshop Knowledge Discovery from Data Streams. 2006. V. 6. Pp. 77—86.

**References**

1. **Kubát M.** A Machine Learning-based Approach to Load Balancing in Computer Networks. Cybernetics and Systems. 1992;23(3—4):389—400.

2. **Widmer G., Kubát M.** Effective Learning in Dynamic Environments by Explicit Context Tracking. Proc. Machine Learning Conf. 1993;667:227—243.

3. **Widmer G., Kubát M.** Learning in the Presence of Concept Drift and Hidden Contexts. Machine Learning. 1996;23(1):69—101.

4. **Machine** Learning Monitoring. Pt. 5. Why You Should Care about Data and Concept Drift [Elektron. Resurs] https://evidentlyai.com/blog/machine-learning-monitoring-data-and-concept-drift (Data Obrashcheniya 14.06.2022).

5. **Zang W. e. a.** Comparative Study between Incremental and Ensemble Learning on Data Streams: Case Study. J. Big Data. 2014;1(5):1—5.

6. **Žliobaitė I., Pechenizkiy M., Gama J.** An Overview of Concept Drift Applications. Big Data Analysis: New Algorithms for a New Society. N.-Y.: Springer, 2016;16:91—114.

7. **Webb G. I. e. a.** Understanding Concept Drift [Elektron. Resurs] https://www.researchgate.net/publication/315765198_Understanding_Concept_Drift (Data Obrashcheniya 14.06.2022).

8. **Nakhaeizadeh G., Taylor C.C., Kunisch G.** Dynamic Supervised Learning: Some Basic Issues and Application Aspects. Classification and Knowledge Organization. Studies in Classification, Data Analysis, and Knowledge Organization. Berlin: Springer, 1997: 123—135.

9. **Tsymbal A.** The Problem of Concept Drift: Definitions and Related Work. Dublin: Trinity College, 2004; 106(2). P. 7.

10. **Zhukov A.V., Sidorov D.N.** Modifikatsiya Algoritma Sluchaynogo Lesa dlya Klassifikatsii Nestatsionarnykh Potokovykh Dannykh. Vestnik Yuzhno-Ural'skogo Gos. Un-ta. Seriya «Matematicheskoe Modelirovanie i Programmirovanie». 2016;4(9):86—95. (in Russian).

11. **Gimarov V.A.** Metody i Avtomatizirovannye Sistemy Dinamicheskoy Klassifikatsii Slozhnykh Tekhnogennykh Ob'ektov: Avtoref. Dis. … Doktora Tekhn. Nauk. M.: Izd-vo Rossiyskogo Khimiko-tekhnolog. Un-ta im. D.I. Mendeleeva, 2004. (in Russian).

12. **Nakhaeizadeh G., Taylor C., Lanquillon C.** Evaluating Usefulness for Dynamic Classification. KDD. 1998;1:87—93.

13. **Gama J. e. a.** Learning with Drift Detection Lecture Notes in Computer Science. Proc. Brazilian Symp. Artifical Intelligence. 2004;3171:286—295.

14. **Baena-Garc M. e. a.** Early Drift Detection Method. Proc. IV Intern. Workshop Knowledge Discovery from Data Streams. 2006;6:77—86.

15. **Althabiti M., Abdullah M.** CDDM: Concept Drift Detection Model for Data Stream // Intern. J. Interactive Mobile Technol. 2020. V. 14(10). Pp. 90—106.

16. **Dong F. e. a.** Fuzzy Competence Model Drift Detection for Data-Driven Decision Support Systems // Knowledge-based Syst. 2018. V. 143. Pp. 284—294.

17. **Boracchi G. e. a.** QuantTree: Histograms for Change Detection in Multivariate Data Streams // Proc. XXXV Intern. Conf. Machine Learning. 2018. V. 80. Pp. 639—648.

18. **Kifer D., Ben-David S., Gehrke J.** Detecting Change in Data Streams // Proc. XXX Intern. conf. Very Large Data Bases. 2004. V. 4. Pp. 180—191.

19. **Sethi T.S., Kantardzic M.** On the Reliable Detection of Concept Drift from Streaming Unlabeled Data // Expert Systems with Appl. 2017. V. 82. Pp. 77—99.

20. **Jaworski M., Rutkowski L., Angelov P.** Concept Drift Detection Using Autoencoders in Data Streams Processing // Proc. Intern. Conf. Artificial Intelligence and Soft Computing. 2020. V. 12415. Pp. 124—133.

21. **Ditzler G.** Incremental Learning of Concept Drift from Imbalanced Data. Glassboro: Rowan University, 2011.

22. **Krawczyk B., Woźniak M.** One-class Classifiers with Incremental Learning and Forgetting for Data Streams with Concept Drift // Soft Comput. 2015. V. 19. Pp. 3387—3400.

23. **Bifet A., Gavaldà R.** Learning from Time-changing Data with Adaptive Windowing // Proc. SIAM Intern. Conf. Data Mining. 2007. Pp. 443—448.

24. **Sun Y., Shao H., Wang S.** Efficient Ensemble Classification for Multi-label Data Streams with Concept Drift // Information. 2019. V. 10(5). Pp. 158—172.

25. **Sarnovsky M., Kolarik M.** Classification of the Drifting Data Streams Using Heterogeneous Diversified Dynamic Class-weighted Ensemble // PeerJ Computer Sci. 2021. V. 7. Pp. 459—490.

26. **Ludwig S.** Applying a Neural Network Ensemble to Intrusion Detection // J. Artificial Intelligence and Soft Computing Research. 2019. V. 9(3). Pp. 177—188.

27. **Kolter J.Z., Maloof M.** Dynamic Weighted Majority: an Ensemble Method for Drifting Concepts // J. Mach. Learn. Res. 2007. V. 8. Pp. 2755—2790.

28. **Best** Practices for Dealing with Concept Drift [Электрон. ресурс] https://opendatascience.com/best-practices-for-dealing-with-concept-drift/ (дата обращения 19.06.2022).

29. **Montiel J.** Learning from Evolving Data Streams // Proc. IXX Python in Sci. Conf. 2020. Pp. 70—77.

30. **Bifet A., Gavaldà R.** Adaptive Learning from Evolving Data Streams // Advances in Intelligent Data Analysis VIII Lecture Notes in Computer Sci. Berlin, Heidelberg: Springer, 2009. Pp. 249—260.

31. **Gözüaçik Ö., Can F.** Concept Learning Using One-class Classifiers for Implicit Drift Detection in Evolving Data Streams // Artifical Intelligence Rev. 2021. V. 54(3). Pp. 1—23.

15. **Althabiti M., Abdullah M.** CDDM: Concept Drift Detection Model for Data Stream. Intern. J. Interactive Mobile Technol. 2020;14(10):90—106.

16. **Dong F. e. a.** Fuzzy Competence Model Drift Detection for Data-Driven Decision Support Systems. Knowledge-based Syst. 2018;143:284—294.

17. **Boracchi G. e. a.** QuantTree: Histograms for Change Detection in Multivariate Data Streams. Proc. XXXV Intern. Conf. Machine Learning. 2018;80: 639—648.

18. **Kifer D., Ben-David S., Gehrke J.** Detecting Change in Data Streams. Proc. XXX Intern. conf. Very Large Data Bases. 2004;4:180—191.

19. **Sethi T.S., Kantardzic M.** On the Reliable Detection of Concept Drift from Streaming Unlabeled Data. Expert Systems with Appl. 2017;82:77—99.

20. **Jaworski M., Rutkowski L., Angelov P.** Concept Drift Detection Using Autoencoders in Data Streams Processing. Proc. Intern. Conf. Artificial Intelligence and Soft Computing. 2020;12415:124—133.

21. **Ditzler G.** Incremental Learning of Concept Drift from Imbalanced Data. Glassboro: Rowan University, 2011.

22. **Krawczyk B., Woźniak M.** One-class Classifiers with Incremental Learning and Forgetting for Data Streams with Concept Drift. Soft Comput. 2015;19:3387—3400.

23. **Bifet A., Gavaldà R.** Learning from Time-changing Data with Adaptive Windowing. Proc. SIAM Intern. Conf. Data Mining. 2007:443—448.

24. **Sun Y., Shao H., Wang S.** Efficient Ensemble Classification for Multi-label Data Streams with Concept Drift. Information. 2019;10(5):158—172.

25. **Sarnovsky M., Kolarik M.** Classification of the Drifting Data Streams Using Heterogeneous Diversified Dynamic Class-weighted Ensemble. PeerJ Computer Sci. 2021;7:459—490.

26. **Ludwig S.** Applying a Neural Network Ensemble to Intrusion Detection. J. Artificial Intelligence and Soft Computing Research. 2019;9(3):177—188.

27. **Kolter J.Z., Maloof M.** Dynamic Weighted Majority: an Ensemble Method for Drifting Concepts. J. Mach. Learn. Res. 2007;8:2755—2790.

28. **Best** Practices for Dealing with Concept Drift [Elektron. Resurs] https://opendatascience.com/best-practices-for-dealing-with-concept-drift/ (Data Obrashcheniya 19.06.2022).

29. **Montiel J.** Learning from Evolving Data Streams. Proc. IXX Python in Sci. Conf. 2020:70—77.

30. **Bifet A., Gavaldà R.** Adaptive Learning from Evolving Data Streams. Advances in Intelligent Data Analysis VIII Lecture Notes in Computer Sci. Berlin, Heidelberg: Springer, 2009:249—260.

31. **Gözüaçik Ö., Can F.** Concept Learning Using One-class Classifiers for Implicit Drift Detection in Evolving Data Streams. Artifical Intelligence Rev;2021;54(3): 1—23.

32. **Kemker R. e. a.** Measuring Catastrophic Forgetting in Neural Networks // Proc. Conf. Artifical Intelligence. 2017. V. 32(1). Pp. 1—15.

33. **Sharkey A.** On Combining Artificial Neural Nets // Connection Sci. 1996. V. 8(3—4). Pp. 299—314.

34. **Tennant M. e. a.** Scalable Real-time Classification of Data Streams with Concept Drift // Future Generation Computer Syst. 2017. V. 75. Pp. 187—199.

35. **Alam K.R., Siddique N., Adeli H.** A Dynamic Ensemble Learning Algorithm for Neural Networks // Neural Computing and Appl. 2020. V. 32(1). Pp. 8675—8690.

36. **Suryawanshi S., Goswami A., Patil P.** Incremental Ensemble of One Class Classifier for Data Streams with Concept Drift Adaption // Proc. Intern. Advanced Computing Conf. Communications in Computer and Information Sci. 2021. V. 1467. Pp. 407—416.

37. **Li Z. e. a.** Incremental Learning Imbalanced Data Streams with Concept Drift: The Dynamic Updated Ensemble Algorithm // Knowledge-Based Syst. 2020. V. 195(4). P. 105694.

38. **Gurina A.O., Eliseev V.L., Kolpinskiy S.V.** Dynamic Classification Approach Using Scalable Ensemble of Autoencoders to Classify Data with Drift // J. Phys.: Conf. Ser. 2021. V. 2134(1). P. 012009.

39. **Гурина А.О., Елисеев В.Л.** Эмпирический критерий качества одноклассового классификатора // Информационные системы и технологии: Материалы XXVII Междунар. науч.-техн. конф. Нижний Новгород: Изд-во Нижегородского гос. техн. ун-та им. Р.Е. Алексеева, 2021. С. 648—657.

40. **Gurina A., Eliseev V.** Quality Criteria and Method of Synthesis for Adversarial Attack-resistant Classifiers // Machine Learning and Knowledge Extraction. 2022. V. 4(2). Pp. 519—541.

41. **Гурина А.О., Гузев О.Ю., Елисеев В.Л.** Обнаружение аномальных событий на хосте с использованием автокодировщика // Intern. J. Open Information Technol. 2020. Т. 8(8). С. 26—36.

42. **Пат. № 2773010 РФ.** Способ обнаружения аномалий в многомерных данных / А.О. Гурина, О.Ю. Гузев // Бюл. изобрет. 2022. № 16.

43. **Skmultiflow**.trees.HoeffdingAdaptiveTreeClassifier — Scikit-multiflow 0.5.3 Documentation [Электрон. ресурс] https://scikit-multiflow.readthedocs.io/en/stable/api/generated/skmultiflow.trees.HoeffdingAdaptiveTreeClassifier.html?highlight=HoeffdingAdaptiveTreeClassifier()#skmultiflow.trees.HoeffdingAdaptiveTreeClassifier (дата обращения 23.06.2022).

32. **Kemker R. e. a.** Measuring Catastrophic Forgetting in Neural Networks. Proc. Conf. Artifical Intelligence. 2017;32(1):1—15.

33. **Sharkey A.** On Combining Artificial Neural Nets. Connection Sci. 1996;8(3—4):299—314.

34. **Tennant M. e. a.** Scalable Real-time Classification of Data Streams with Concept Drift. Future Generation Computer Syst. 2017;75:187—199.

35. **Alam K.R., Siddique N., Adeli H.** A Dynamic Ensemble Learning Algorithm for Neural Networks. Neural Computing and Appl. 2020;32(1):8675—8690.

36. **Suryawanshi S., Goswami A., Patil P.** Incremental Ensemble of One Class Classifier for Data Streams with Concept Drift Adaption. Proc. Intern. Advanced Computing Conf. Communications in Computer and Information Sci. 2021;1467:407—416.

37. **Li Z. e. a.** Incremental Learning Imbalanced Data Streams with Concept Drift: The Dynamic Updated Ensemble Algorithm. Knowledge-Based Syst. 2020;195(4):105694.

38. **Gurina A.O., Eliseev V.L., Kolpinskiy S.V.** Dynamic Classification Approach Using Scalable Ensemble of Autoencoders to Classify Data with Drift. J. Phys.: Conf. Ser. 2021;2134(1):012009.

39. **Gurina A.O., Eliseev V.L.** Empiricheskiy Kriteriy Kachestva Odnoklassovogo Klassifikatora. Informatsionnye Sistemy i Tekhnologii: Materialy XXVII Mezhdunar. Nauch.-tekhn. Konf. Nizhniy Novgorod: Izd-vo Nizhegorodskogo Gos. Tekhn. Un-ta im. R.E. Alekseeva, 2021:648—657. (in Russian).

40. **Gurina A., Eliseev V.** Quality Criteria and Method of Synthesis for Adversarial Attack-resistant Classifiers. Machine Learning and Knowledge Extraction. 2022;4(2):519—541.

41. **Gurina A.O., Guzev O.Yu., Eliseev V.L.** Obnaruzhenie Anomal'nykh Sobytiy na Khoste s Ispol'zovaniem Avtokodirovshchika. Intern. J. Open Information Technol. 2020;8(8):26—36. (in Russian).

42. **Pat. № 2773010 RF.** Sposob Obnaruzheniya Anomaliy v Mnogomernykh Dannykh. A.O. Gurina, O.Yu. Guzev. Byul. Izobret. 2022;16. (in Russian).

43. **Skmultiflow**.trees.HoeffdingAdaptiveTreeClassifier—Scikit-multiflow0.5.3Documentation[Elektron.Resurs] https://scikit-multiflow.readthedocs.io/en/stable/api/generated/skmultiflow.trees.HoeffdingAdaptiveTreeClassifier.html?highlight=HoeffdingAdaptiveTreeClassifier()#skmultiflow.trees.HoeffdingAdaptiveTreeClassifier (Data Obrashcheniya 23.06.2022).

**Сведения об авторах:**

**Гурина Анастасия Олеговна** — старший исследователь Центра научных исследований и перспективных разработок АО «ИнфоТеКС», e-mail: asya.gurina001512@yandex.ru

**Гурина Лариса Анатольевна** — научный сотрудник Научно-исследовательского центра топогеодезического и навигационного обеспечения, 27-й Центральный научно-исследовательский институт Министерства обороны Российской Федерации, e-mail: larisa.gurina0702@gmail.com

**Information about authors:**

**Gurina Anastasiya O.** — Senior Researcher at the Center for Scientific Research and Advanced Development, JSC «InfoTeX», e-mail: asya.gurina001512@yandex.ru
**Gurina Larisa A.** — Researcher at the Research Center for Topogeodesic and Navigation Support, 27th Central Research Institute of the Ministry of Defense of the Russian Federation, e-mail: larisa.gurina0702@gmail.com

**Работа выполнена при поддержке:** РФФИ (научный проект № 20-37-90073)
**The work is executed at support:** RFBR (Scientific Project No. 20-37-90073)

**Конфликт интересов:** авторы заявляют об отсутствии конфликта интересов
**Conflict of interests:** the authors declare no conflict of interest

**Статья поступила в редакцию:** *30.06.2022*
**The article received to the editor:** *30.06.2022*

**Статья принята к публикации:** *24.10.2022*
**The article has been accepted for publication:** *24.10.2022*