Научная статья
УДК 004.732.056
https://doi.org/10.31854/1813-324X-2024-10-6-111-120

Снижение размерности массивов данных с помощью многослойных автокодировщиков в задаче классификации мобильных приложений

Олег Иванович Шелухин, sheluhin@mail.ru
Фёдор Андреевич Маторин ✉, f.matorin@mail.ru

Московский технический университет связи и информатики,
Москва, 111024, Российская Федерация

Аннотация

Рассматривается задача уменьшения размерности исходных массивов данных для улучшения эффективности обработки трафика мобильных приложений. ***Актуальность*** *исследования обусловлена необходимостью оптимизации объемов передаваемых и хранимых данных при работе в условиях ограниченных вычислительных ресурсов, а также повышения скорости и качества аналитических операций. Для решения поставленной задачи применяются многослойные автокодировщики, способные формировать компактные представления исходных данных с минимальными потерями в их информативности. Подход базируется на идее обучения нейросетевых моделей, извлекающих наиболее существенные признаки из исходных массивов и способных восстанавливать их с заданным уровнем точности.*

Используемые методы. *В ходе экспериментов применялись различные архитектуры многослойных автокодировщиков, отличающиеся количеством слоев и размерностями скрытых представлений. Исследования проводились на реальных наборах данных, собранных из мобильных приложений широкого спектра функционала. Анализ осуществлялся путем варьирования внутренних параметров сетей и оценки результатов через интегральный статистический показатель, отражающий степень сжатия. Данный показатель позволяет выявить, насколько сильно изменяется разброс атрибутов при пропускании данных через автокодировщик.*

Результаты. *Для оценки фильтрующих свойств многослойных автокодировщиков предложен интегральный показатель сжатия, характеризующий изменение разброса атрибутов мобильных приложений при пропускании их через автокодировщик заданной структуры. Показатель рассчитывается как отношение среднеквадратического отклонения атрибутов на входе и на выходе, что позволяет оценить степень сжатия данных и степень сохранности информации после обработки. Показано, что увеличение интегрального показателя сжатия свидетельствует о более значительном сжатии исходных данных. Установлено, что фильтрация практически не зависит от типа приложения и лежит в пределах 10–20 % для автокодировщиков с тремя слоями, тогда как для пятислойных автокодировщиков предпочтение отдается кодировщикам с минимальной размерностью внутреннего слоя. Основная* ***новизна*** *работы заключается в разработке интегрального статистического показателя, который не только отражает степень сжатия данных мобильных приложений, но и учитывает сохранность исходной информационной структуры. В отличие от существующих подходов, данный показатель позволяет проводить систематическое сравнение различных архитектур автокодировщиков с учетом не только уменьшения размерности, но и качества восстановления исходной информации. Это создает основу для более объективной оценки эффективности многослойных автокодировщиков в конкретных прикладных условиях.* ***Практическая значимость****. Предложенная методология может быть полезна разработчикам и исследователям, работающим над оптимизацией систем сбора, хранения и обработки данных мобильных приложений. В условиях ограниченных вычислительных ресурсов, характерных для мобильных устройств и встроенных систем, использование многослойных автокодировщиков, настроенных на достижение заданного баланса между сжатием и сохранением информации, обеспечивает существенное сокращение объема передаваемых данных. Результаты исследования могут быть внедрены в существующие аналитические платформы, системы мониторинга и классификации мобильных приложений.*

Ключевые слова: *нейронные сети, классификация, приложения, атрибуты, фильтрация, статистические характеристики*

Ссылка для цитирования: Шелухин О.И., Маторин Ф.А. Снижение размерности массивов данных с помощью многослойных автокодировщиков в задаче классификации мобильных приложений // Труды учебных заведений связи. 2024. Т. 10. № 6. С. 111–120. DOI:10.31854/1813-324X-2024-10-6-111-120. EDN:TOPDUA

Original research
https://doi.org/10.31854/1813-324X-2024-10-6-111-120

Reducing the Dimensionality of Data Arrays Using Multi-Layer Autoencoders in the Task of Classifying Mobile Applications

Oleg I. Sheluhin, sheluhin@mail.ru
Fedor A. Matorin✉, f.matorin@mail.ru

Moscow Technical University of Communications and Informatics,
Moscow, 111024, Russian Federation

Annotation

The problem of reducing the dimension of the initial data arrays to improve the efficiency of mobile application traffic processing is considered. ***The relevance*** *of the study is due to the need to optimize the volume of transmitted and stored data when working in conditions of limited computing resources, as well as to increase the speed and quality of analytical operations. To solve this problem, multi-layer autoencoders are used, capable of forming compact representations of the source data with minimal losses in their informativeness. The approach is based on the idea of training neural network models that extract the most significant features from the source arrays and are able to restore them with a given level of accuracy.* ***Methods used.*** *During the experiments, various architectures of multilayer autocoders were used, differing in the number of layers and dimensions of hidden representations. The research was conducted on real data sets collected from mobile applications with a wide range of functionality. The analysis was carried out by varying the internal parameters of the networks and evaluating the results through an integral statistical indicator reflecting the degree of compression. This indicator allows you to identify how much the spread of attributes changes when passing data through the autoencoder.*

Results. *To evaluate the filtering properties of multilayer autoencoders, an integral compression indicator is proposed that characterizes the change in the spread of attributes of mobile applications when passing them through an autoencoder of a given structure. The indicator is calculated as the ratio of the standard deviation of the attributes at the input and at the output, which allows you to assess the degree of data compression and the degree of information preservation after processing. It is shown that an increase in the integral compression index indicates a more significant compression of the initial data. It was found that filtering is practically independent of the type of application and lies within 10–20 % for three-layer autoencoders, whereas for five-layer auto-encoders, preference is given to encoders with a minimum dimension of the inner layer. The main* ***novelty*** *of the work lies in the development of an integral statistical indicator that not only reflects the degree of compression of mobile application data, but also takes into account the preservation of the original information structure. Unlike existing approaches, this indicator allows for a systematic comparison of various architectures of autoencoders, taking into account not only the reduction in dimension, but also the quality of recovery of the original information. This creates the basis for a more objective assessment of the effectiveness of multilayer autoencoders in specific application conditions.*

Practical significance. *The proposed methodology may be useful for developers and researchers working on optimizing systems for collecting, storing and processing mobile application data. In conditions of limited computing resources, which are typical for mobile devices and embedded systems, the use of multilayer autoencoders aimed at achieving a given balance between compression and preservation of information provides a significant reduction in the volume of transmitted data. The results of the study can be implemented into existing analytical platforms, monitoring systems and classification of mobile applications.*

Keywords: *neural networks, classification, applications, attributes, filtering, statistical characteristics*

For citation: Sheluhin O.I., Matorin F.A. Reducing the Dimensionality of Data Arrays Using Multi-Layer Autoencoders in the Task of Classifying Mobile Applications. *Proceedings of Telecommunication Universities.* 2024;10(6):111–120. (in Russ.) DOI:10.31854/1813-324X-2024-10-6-111-120. EDN:TOPDUA

Постановка задачи

Снижение размерности данных играет ключевую роль в задачах анализа больших массивов информации, особенно в контексте обработки данных мобильных приложений. Эти приложения генерируют значительные объемы сетевого трафика, который часто содержит избыточную информацию. Эффективная фильтрация и сжатие таких данных позволяют уменьшить объем обрабатываемой информации, снижая нагрузку на сеть и требования к вычислительным ресурсам. Это особенно актуально в условиях ограниченной полосы пропускания и низких мощностей мобильных устройств. Для снижения размерности данных мо-

гут использоваться различные методы, основанные на искусственных нейронных сетях.

Глубокие сверточные нейронные сети [1], эффективно обрабатывающие данные с пространственной структурой, что делает их отличным выбором для работы с изображениями. Однако использование глубоких сверточных нейронных сетей требует значительного объема размеченных данных, что не всегда возможно при работе с неструктурированными данными, такими как сетевой трафик мобильных приложений.

Сети глубокого доверия [2], позволяющие поэтапно обучать модель, снижая размерность данных. Однако подобные сети требуют тщательной настройки большого количества гиперпараметров, что может затруднять их использование в условиях ограниченных вычислительных ресурсов [3].

Ограниченные машины Больцмана [4] являются эффективным инструментом для выявления скрытых зависимостей в данных, однако их обучение является вычислительно затратным при увеличении размерности данных и сложности структуры.

Многослойные автокодировщики [1, 5–7] позволяют использовать неконтролируемое обучение, что делает их особенно подходящими для работы с неразмеченными данными. Они обеспечивают баланс между эффективностью сжатия и сохранением значимой информации, что особенно важно для задач обработки мобильных приложений.

Опираясь на рассмотренные методы и работы, можно сделать вывод, что многослойные автокодировщики (АК) являются наиболее подходящим выбором для задач, связанных с обработкой трафика мобильных приложений, благодаря их способности к неконтролируемому обучению и высокой эффективности при работе с неразмеченными данными. В работах [8, 9] была доказана эффективность применения АК в задачах классификации нежелательных мобильных приложений, что подтверждает их целесообразность использования в данной области. Выбор многослойных АК для обработки данных мобильных приложений позволяет найти баланс между сжатием и сохранением значимой информации, а также позволяют работать с неразмеченными данными, что снижает требования к предварительной подготовке данных и улучшает эффективность обработки.

Целью работы является исследование влияния многослойных АК на фильтрацию и снижение размерности обрабатываемых данных мобильных приложений с целью улучшения эффективности их классификации и обработки. Достижение этой цели позволит сократить объем данных, которые необходимо передавать и хранить, что приведет к снижению нагрузки на вычислительные ресурсы и сети передачи данных, а также увеличит точность классификации приложений, что особенно важно в задачах обеспечения кибербезопасности, автоматического контроля и оптимизации работы мобильных сервисов.

Модели многослойных автокодировщиков

Основой для построения всех моделей многослойных АК является модель простого трехслойного АК. Это сеть прямого распространения с входным и выходным слоями, содержащими одинаковое число нейронов, и единственным внутренним (горловым) слоем, содержащим меньшее число нейронов, чем входной и выходной слои.

Будем считать $X_1, X_2, \dots X_M \in R^N$ векторами входных данных, характеризующими анализируемые мобильные приложения. Тогда матрицу входных данных можно представить в виде:

$$X = [X_1, X_2, \dots X_M]^T,$$

где каждая строка представляет собой вектор обрабатываемых признаков (атрибутов) M анализируемых приложений, а число столбцов N характеризует размерность пространства признаков.

В результате матрица X представляет собой матрицу размера $N \times M$:

$$X = \begin{vmatrix} X_{11} & X_{12} & \cdots & X_{1M} \\ X_{21} & X_{22} & \cdots & X_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ X_{N1} & X_{N2} & \cdots & X_{NM} \end{vmatrix}, X \in R^{N \times M}.$$

Рассмотрим структуру предназначенного для сокращения размерности больших массивов данных, подлежащих обработке, многослойного АК [4, 10], который представляет собой специальный вид сети прямого распространения (МАК-сеть) – многослойный симметричный персептрон, содержащий несколько внутренних слоев уменьшающегося размера и слой «бутылочная горловина» в середине сети. МАК-сеть производит тождественное преобразование входного слоя на выходной. В результате ее работы в горловом слое появляется вектор, компонентами которого являются «признаки» – обобщенные характеристики входного массива данных, извлеченные из исходных данных и содержащие дополнительную существенную и не избыточную информацию, определяющую входной массив данных в пространстве меньшей размерности в так называемом скрытом пространстве.

Задачей скрытого пространства является выделение важных признаков (атрибутов), которые будут использоваться для восстановления исходных данных при максимально малой размерности слоя. Структура простейшего трехслойного АК представлена на рисунке 1.

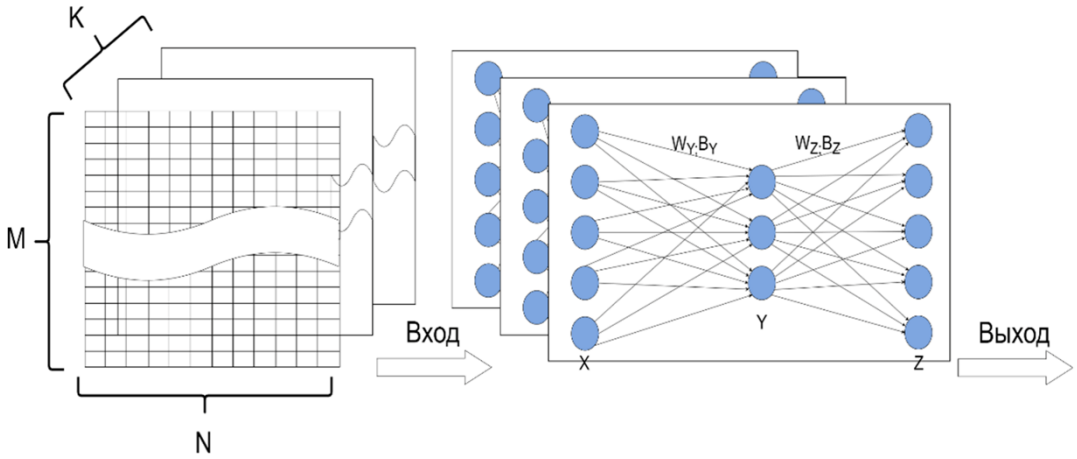

Рис. 1. Структура трехслойного АК

Fig. 1. Structure of a Three-Layer Autoencoder (AE)

Как показано на рисунке 1, наиболее простой АК представляет собой многослойный персептрон, который имеет один скрытый слой и один выходной с двумя ограничениями: матрица весов выходного слоя является транспонированной матрицей весов скрытого слоя $\widehat{W}_Y = \widehat{W}_Z^T = \widehat{W}$ (т. е. веса фиксированы) и количество выходных нейронов равно количеству входных.

Значения нейронов скрытого слоя, называемые кодированием, вычисляются по выражению:

$$Y = G_\theta(X) = F\left(\widehat{W}_Y X + B_Y\right),\ \theta = \{W_Y, B_Y\}, \qquad (1)$$

где X – входной вектор; F – функция активации нейронов сети; B_Y – вектор скрытых нейронных смещений; W_Y – матрица скрытых весов.

Задача функции кодирования $Y = F(X, \widehat{W}_Y, B_Y)$ заключается в сжатии входного вектора в соответствии с уравнениями.

Операция декодирования характеризуется функцией декодирования $Z = F(Y, \widehat{W}_Z, B_Z)$ и заключается в восстановлении входного «сжатого» вектора:

$$Z = G_{\hat{\theta}}(Y) = F\left(\widehat{W}_Z Y + B_Z\right),\ \hat{\theta} = \{W_Z, B_Z\}. \qquad (2)$$

В формулах (1) и (2) $\widehat{W}_Y$ и $\widehat{W}_Z$ – матрицы сетевых связей (матрицы весовых коэффициентов) кодировщика и декодировщика АК; весовые коэффициенты B_Y и B_Z – векторы смещения (определяют важность каждого входного сигнала для вычисления выходных значений слоя); θ и $\hat{\theta}$ – наборы параметров отображения.

Каждый нейрон имеет свое собственное смещение, не зависящее от входных данных, и настраивается в процессе обучения модели вместе с весами. Количество весов определяется количеством нейронов на предыдущем слое, а количество смещений – количеством нейронов на текущем. Общее количество параметров определяется соотношением:

$$(input_{demention} + 1) * cur_{dence_{demention}},$$

где $input_{demention}$, $cur_{dence_{demention}}$ – размерность предыдущего (размер выборки) и текущего слоя, соответственно, или значение веса и смещения.

Размерности скрытых слоев зависят от желаемой степени сжатия входных данных, количества признаков выборки и целевого значения размерности скрытого пространства – параметра, который влияет на способность модели к обучению и реконструкции. Слой Y содержит меньшее количество информативных параметров обрабатываемого массива данных, извлеченных в процессе работы АК. Меньшая размерность скрытого слоя может привести к более эффективному сжатию и выделению значимых признаков. Однако при этом увеличивается риск потери информации, и наоборот.

Целью обучения АК является минимизация разницы между входными X и выходными Z данными.

Типичная функция потерь представляет собой среднеквадратическую ошибку (СКО):

$$L(X, Y) = \|X - Z\|^2. \qquad (3)$$

Используя (1) и (2), выражение (3) может быть преобразовано к следующему виду:

$$L(X, Y) = \left\| X - F\left(\widehat{W}_Z\left(F\left(\widehat{W}_Y X + B_Y\right)\right) + B_Z\right)\right\|^2. \qquad (4)$$

Функция потерь $L(X, Y)$ определяет качество реконструкции оригинала, так что выходная реконструкция должна быть как можно ближе к исходному входному вектору. Отсюда основной задачей является минимизация значений функции потерь и обновление ее параметров для повышения точности реконструкции.

Наиболее распространенными функциями потерь являются СКО (MSE) и корень из СКО (RMSE).

Настройка многослойных АК осуществляется путем минимизации функции потерь:

$$L(X,Y) = cost(X,Y),$$

которая может быть произведена различными способами, например методом градиентного спуска, и позволяет обновлять параметры для повышения точности.

Основная цель обучения многослойных АК состоит в том, чтобы найти оптимальные параметры (θ и $\hat{\theta}$), которые могут эффективно минимизировать разницу между входными и восстановленными выходными данными по всему обучающему набору:

$$\theta = \{W, B\} = \arg_{\theta}\min L(X,Y). \quad (5)$$

Работу многослойной нейронной сети прямого распространения можно интерпретировать как вычисление композиции многомерных отображений многослойного АК, содержащего несколько внутренних слоев, которые обладают большими возможностями по сравнению с простыми трехслойными АК. В качестве примера на рисунке 2 изображена структура пятислойного АК.

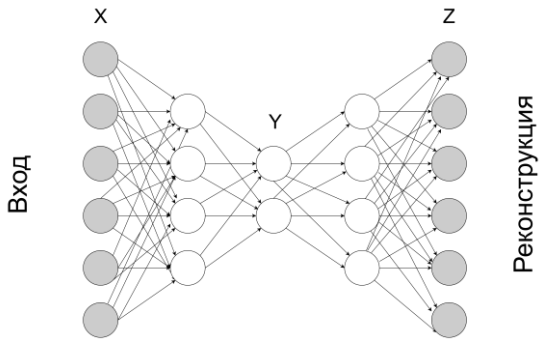

Рис. 2. Структура пятислойного АК

Fig. 2. Structure of a Five-Layer AE

Для многослойного АК можно записать:

$$Z = G\left(F(X;\theta), \theta\left(\{W_{jk}\}, \{V_k\}\right)\right),$$

где значения θ, обеспечивающих наилучшую аппроксимацию композиции функций, находятся путем обучения АК.

Вектор состояния слоя j; $Z^{(j)} \in R^{Lj}$, преобразуется в вектор состояния слоя j+1; $Z^{(j+1)} \in R^{j+1}$:

$$Z^{(j+1)} = \widehat{W}^{(i)} F\left(Z^{(j)}\right) + B^{(j)}, \quad (6)$$

где $\widehat{W}^{(i)}$ – веса $(L_j \times L_{j+1})$ матрицы связей слоев j и j+1; F – функция активации нейронной сети.

В структуре АК могут использоваться различные функции активации нейронов сети [11]. Нелинейность функции активации позволяет извлекать из исходных данных более существенные обобщенные характеристик, устраняя как линейные, так и нелинейные корреляции.

Реализация АК подразумевает под собой конфигурирование слоев кодера и декодера, указание функций активации для них, выбор гиперпараметров и оптимизацию параметров АК. Кроме перечисленных параметров при использовании многослойных АК, в задачах классификации и прогнозирования необходимо задаться гиперпараметрами. Для моделей многослойных АК гиперпараметрами являются веса и смещения внутри каждого слоя кодировщика и декодировщика. Эти параметры определяют, как модель сжимает входные данные и восстанавливает их обратно.

Оценка фильтрующих свойств многослойных автокодировщиков

Рассмотрим оценку фильтрующих свойств многослойных АК на примере экспериментальных данных мобильных приложений, приведенных в работах [12, 13]. Для формирования обучающей и тестовой выборок на мобильных устройствах под управлением ОС Android осуществлялся сбор необработанных данных сетевого трафика в виде IP-пакетов. Обработка данных (в том числе фильтрация пакетов, содержащих данные протокола TCP, группировка пакетов в TCP-сеансы и вычисление их атрибутов, характеризующих особенности анализируемых приложений) осуществлялась на сервере всякий раз, когда поступал IP-пакет. С применением разработанного программного комплекса был собран трафик различных типов мобильных приложений, из которых в дальнейшем будем использовать M = 6 мобильных приложений (*Skype*, *Booking*, *Instagram* (Деятельность Meta Platform Inc. по реализации продуктов – социальных сетей Facebook и Instagram на территории РФ запрещена по основаниям осуществления экстремистской деятельности), *Mail*, *SberMobile*). Каждое из них описывается набором из N = 21 атрибута, характеризующих то или иное приложение. Общее число экспериментально измеренных потоков каждого приложения составляло K = 5000 измерений.

Фильтрующую способность многослойных АК будем характеризовать динамическим диапазоном изменения разброса численных значений атрибутов исследуемых приложений до и после обработки данных с помощью АК. Эффект фильтрации можно проиллюстрировать гистограммами распределения одного из атрибутов приложения *Mail* до и после АК, представленными на рисунке 3. Как видно из рисунка, специфика структуры и обработки в АК приводит к уменьшению динамического диапазона изменения численных значений атрибутов на выходе АК, что иллюстрирует эффект сжатия (фильтрации). Исследовались структуры многослойных АК с тремя и пятью слоями и сигмоидальной функцией активации.

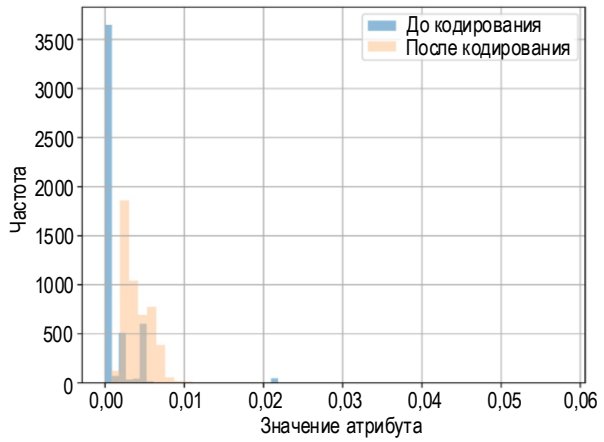

Рис. 3. Гистограмма атрибута *i* = 18 приложения *j* = 5 (*Mail*) до и после АК: $\sigma_{18,5\ вх} = 0{,}0126$; $\sigma_{18,5вых} = 0{,}0047$

Fig. 3. Histogram of Attribute i = 18 of Application j = 5 (Mail) before and after AE: $\sigma_{18,5\ вх} = 0{,}0126$; $\sigma_{18,5вых} = 0{,}0047$

Для формализации параметра, характеризующего фильтрующую способность многослойных АК введем в рассмотрение следующие обозначения:

$\sigma^2_{ij\text{вх}} = \frac{1}{K}\sum_{K=1}^{K}(a^{\text{вх}}_{ijk} - ma^{\text{вх}}_{ijk})^2$ – дисперсия *i*-го атрибута *j*-го приложения на входе многослойного АК;

$ma^{\text{вх}}_{ijk} = \frac{1}{K}\sum_{K=1}^{K} a^{\text{вх}}_{ijk}$ – среднее значение *i*-го атрибута *j*-го приложения, посчитанное по *K* измерениям;

$\sigma_{ij\text{вх}} = \sqrt{\sigma^2_{ij\text{вх}}}$ – СКО *i*-го атрибута *j*-го приложения, на входе многослойного АК;

$\sigma^2_{ij\text{вых}} = \frac{1}{K}\sum_{K=1}^{K}(a^{\text{вых}}_{ijk} - ma^{\text{вых}}_{ijk})^2$ – дисперсия *i*-го атрибута *j*-го приложения на выходе многослойного АК;

$ma^{\text{вых}}_{ijk} = \frac{1}{K}\sum_{K=1}^{K} a^{\text{вых}}_{ijk}$ – среднее значение *i*-го атрибута *j*-го приложения на выходе многослойного АК, посчитанное по *K* измерениям;

$\sigma_{ij\text{вых}} = \sqrt{\sigma^2_{ij\text{вых}}}$ – СКО *i*-го атрибута *j*-го приложения на выходе многослойного АК;

$\Delta\sigma_{ij} = \sigma_{ij\text{вх}} - \sigma_{ij\text{вых}}$ – абсолютное изменение СКО *i*-го атрибута *j*-го приложения на выходе многослойного АК по сравнению со входом;

$\delta_{ij} = \frac{\Delta\sigma_{ij}}{\sigma_{ij\text{вх}}} * 100\ \%$ – относительное изменение СКО *i*-го атрибута *j*-го приложения на выходе многослойного АК по сравнению со входом;

$\frac{1}{N}\sum_{i=1}^{N}\sigma_{ij\text{вх}}$ – усредненное по всем атрибутам среднее значение СКО *j*-го приложения на входе многослойного АК;

$\frac{1}{N}\sum_{i=1}^{N}\sigma_{ij\text{вых}}$ – усредненное по всем атрибутам среднее значение СКО *j*-го приложения на выходе многослойного АК;

$\frac{1}{N}\sum_{i=1}^{N}\Delta\sigma_{ij}$ – усредненное по всем атрибутам среднее значение абсолютного изменения СКО *j*-го приложения на выходе многослойного АК по сравнению со входом;

$\frac{1}{N}\sum_{i=1}^{N}\delta_{ij}\ \%$ – усредненное по всем атрибутам среднее *относительное* изменение СКО *j*-го приложения на выходе многослойного АК по сравнению со входом.

Введенную в рассмотрение величину, которую можно описать следующим выражением:

$$\text{ИПС}j = \frac{1}{N}\sum_{i=1}^{N}\delta_{ij}\ \% = \frac{1}{N}\sum_{i=1}^{N}\frac{\Delta\sigma_{ij}}{\sigma_{ij\text{вх}}} * 100\ \% = \\ = \frac{1}{N}\sum_{i=1}^{N}\frac{\sigma_{ij\text{вх}} - \sigma_{ij\text{вых}}}{\sigma_{ij\text{вх}}} * 100\ \%, \quad (7)$$

будем называть интегральным статистическим показателем (ИПС, *от англ.* Integral Statistic – *IS*) сжатия многослойного АК *j*-го приложения и использовать его для оценки фильтрующей способности многослойного АК заданной структуры. Данная величина является интегральным показателем, характеризующим изменение разброса атрибутов рассматриваемых приложений при пропускании через многослойный АК заданной структуры. Чем больше величина ИПС, тем значительней СКО входного показателя больше СКО выходного показателя. В качестве примера в таблице 1 приведены промежуточные результаты оценки введенных метрик для многослойного АК с тремя слоями.

ТАБЛИЦА 1. Метрики для АК с 3 слоями и структурой 21-5-21

TABLE 1. Metrics for AE with 3 Layers and Structure 21-5-21

j	Приложения	$\frac{1}{N}\sum_{i=1}^{N}\sigma_{ij\text{вх}}$	$\frac{1}{N}\sum_{i=1}^{N}\sigma_{ij\text{вых}}$	$\frac{1}{N}\sum_{i=1}^{N}\Delta\sigma_{ij}$	ИПС, %
1	*Chrome*	0,1496	0,1406	0,0090	10,2823
2	*Yandex*	0,0948	0,0820	0,0129	30,6003
3	*Booking*	0,0915	0,0728	0,0188	38,0363
4	*ISG**	0,0993	0,0873	0,0119	21,4529
5	*Mail*	0,0898	0,0770	0,0128	32,9408
6	*SberMobile*	0,1496	0,1406	0,0090	10,2823

Значения ИПС для последовательности каждого из 21-го атрибута приложения при использовании АК с 3-мя слоями и структурой представлены на рисунке 4. Анализ представленных зависимостей показывает, что у АК с тремя слоями наблюдается выигрыш в фильтрующей способности обусловленный уменьшением СКО процесса на выходе кодировщика. Зависимости уменьшения разброса выходных данных АК от анализируемых атрибутов, представленные на рисунке 4, показывают, что выигрыш слабо зависит от типа приложения и лежит в среднем в пределах 10…20 % за исключением атрибутов № 2, 3, 13, 18, у которых выигрыш достигает 60…100 %. Сравнительный анализ эффективности АК с тремя слоями оцениваемый по-

** Деятельность Meta Platform Inc. по реализации продуктов – социальных сетей Facebook и Instagram на территории РФ запрещена из-за экстремистской деятельности*

казателем ИПС иллюстрируется гистограммами, приведенными на рисунке 5.

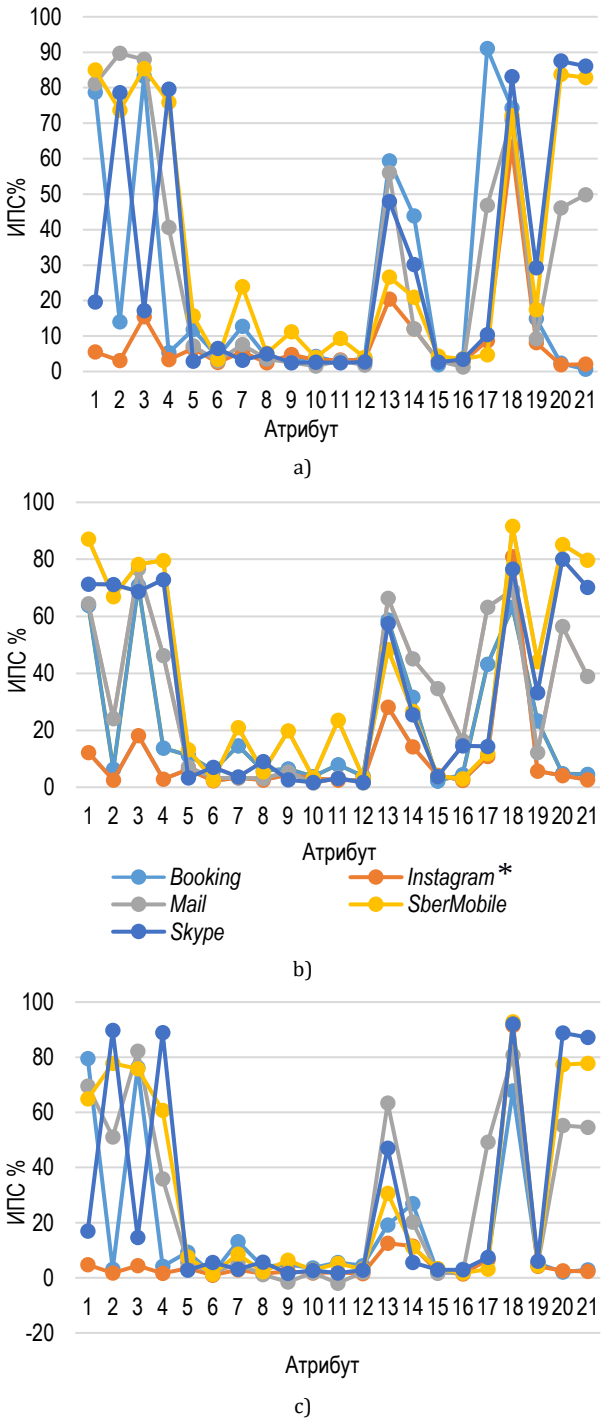

Рис. 4. Значение ИПС для последовательности атрибутов приложения, при использовании АК с 3-мя слоями и структурой: a) 21-7-21; b) 21-5-21; c) 21-9-21

Fig. 4. IS Value for a Sequence of Application Attributes, Using an AE with 3 Layers and Structure: a) 21-7-21; b) 21-5-21; c) 21-9-21

Гистограммы показывают, что в среднем уменьшение разброса обрабатываемых данных, оцениваемое величиной СКО, наиболее предпочтительно для АК с тремя слоями и структурой 21-5-21. Для этой структуры выигрыш достигает 20…25 % независимо от типа приложения.

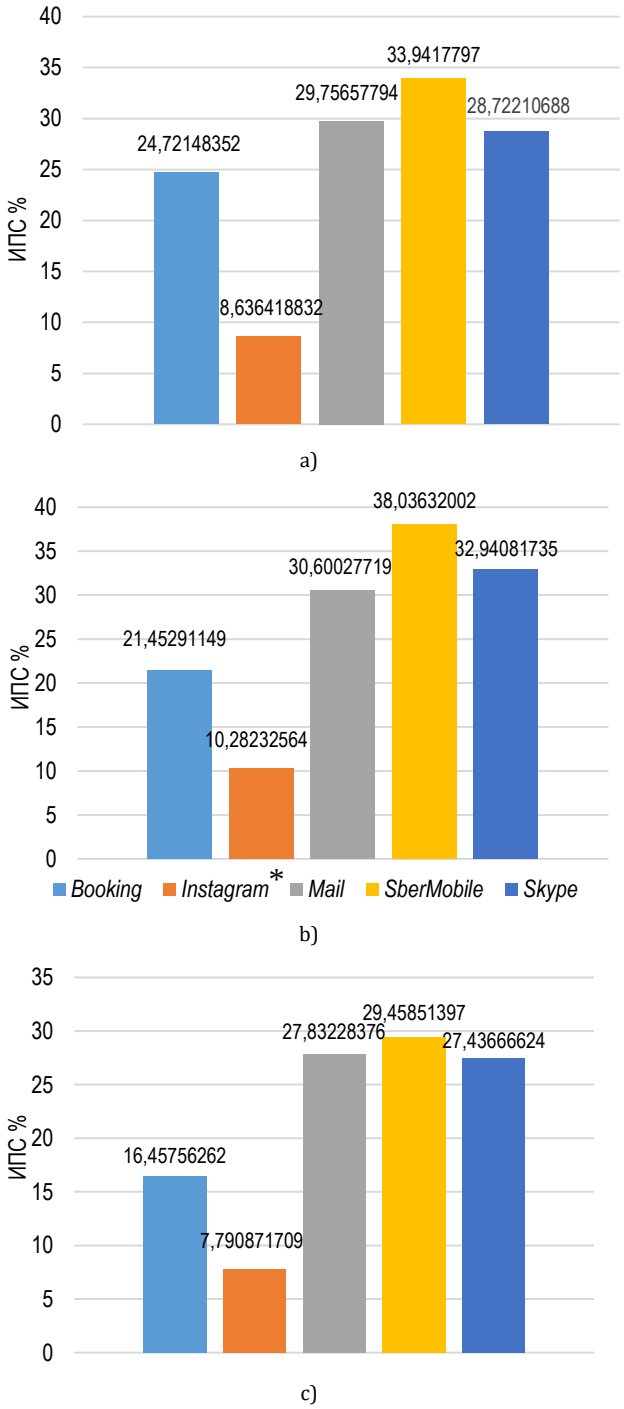

Рис. 5. Гистограммы среднего значения ИПС в процентах от приложения, при использовании АК с 3-мя слоями и структурой: a) 21-7-21; b) 21-5-21; c) 21-9-21

Fig. 5. Histograms of the Average IS Value in Percent of the Application, Using AE with 3 Layers and Structure: a) 21-7-21; b) 21-5-21; c) 21-9-21;

Для структуры 21-7-21 выигрыш скромнее и достигает в среднем 15…20 %, а для структуры 21-9-21 – не превышает 15 %.

* Деятельность Meta Platform Inc. по реализации продуктов – социальных сетей Facebook и Instagram на территории РФ запрещена из-за экстремистской деятельности

Таким образом, для анализа эффективности АК в задаче классификации нежелательных приложений целесообразно ограничиться структурой с наименьшим размером внутреннего слоя 21-5-21.

Анализ фильтрующих свойств многослойных АК для количества слоев более трех проводился для числа слоев равного 5. Численные значения для АК с 5 слоями и структурой 21-14-5-14-21 для рассмотренных выше метрик представлены в таблице 2. На рисунке 6 представлены зависимости ИПС при использовании АК с 5-ю слоями и различной структурой слоев. Анализ многослойных АК с пятью слоями и структурами 21-14-5-14-21 (см. рисунки 6c, 6d) и 21-14-7-14-21 (см. рисунки 6a, 6b) показывает, что зависимости от вида атрибутов сохраняются, как и для АК с тремя слоями. Выигрыш в среднем не превышает 10 % за исключением атрибутов с № 13, 17, в которых выигрыш может достигать 80…100 %.

ТАБЛИЦА 2. Метрики для АК с 5-ю слоями и структурой 21-14-5-14-21

TABLE 2. Metrics for AE with 5 Layers and Structure 21-14-5-14-14-21

j	Приложения	$\frac{1}{N}\sum_{i=1}^{N}\sigma_{ij\text{вх}}$	$\frac{1}{N}\sum_{i=1}^{N}\sigma_{ij\text{вых}}$	$\frac{1}{N}\sum_{i=1}^{N}\Delta\sigma_{ij}$	ИПС, %
1	*ISG**	0,1496	0,1446	0,0050	7,6304
2	*Mail*	0,0948	0,0896	0,0052	15,7605
3	*SberMobile*	0,0915	0,0872	0,0044	11,1490
4	*Booking*	0,0993	0,0898	0,0095	19,0898
5	*Chrome*	0,0898	0,0834	0,0064	21,9972
6	*Yandex*	0,1496	0,1446	0,0050	7,6304

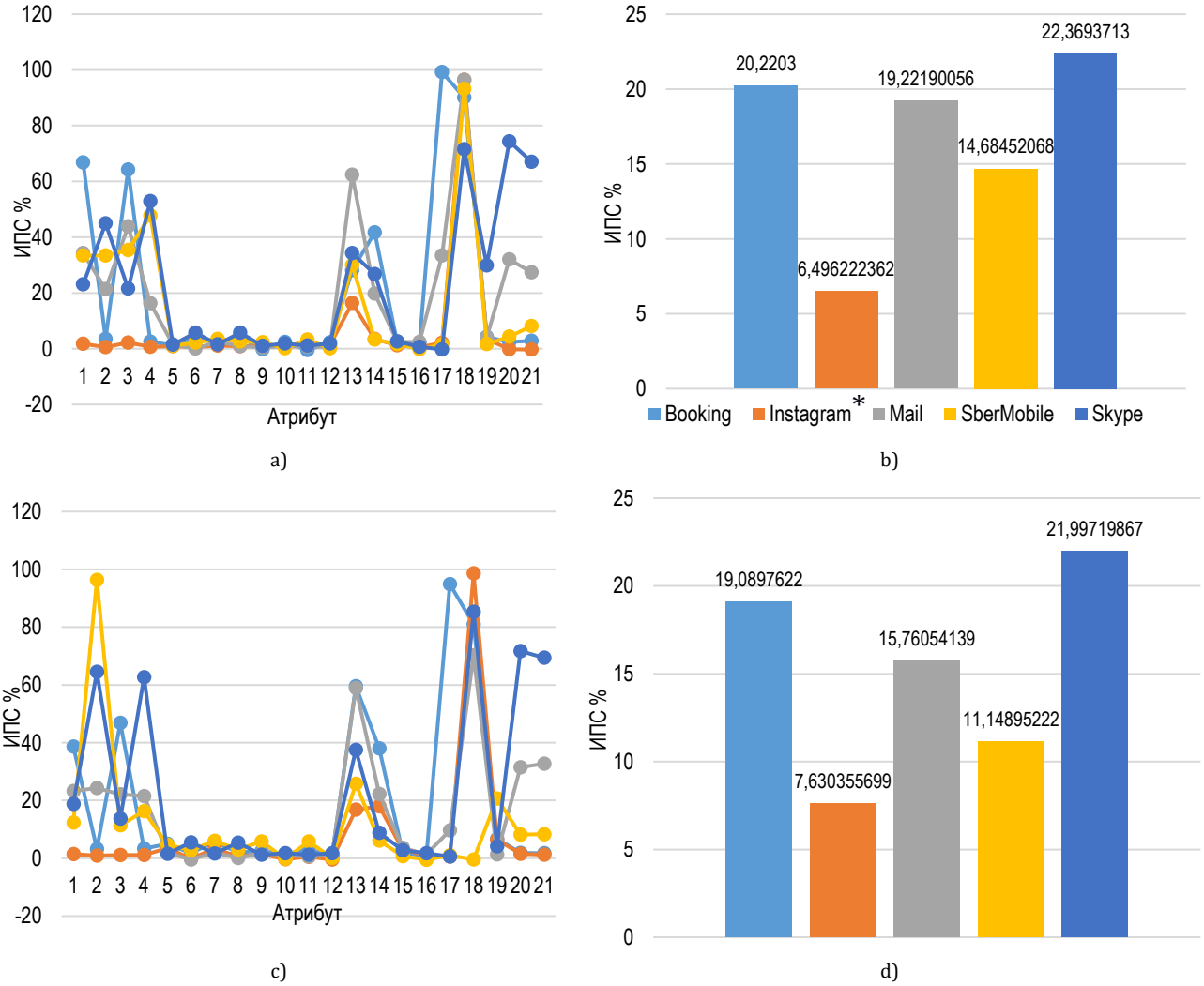

Рис. 6. Зависимости ИПС при использовании АК с 5-ю слоями от типа атрибута для рассматриваемых приложений (слева) и от гистограммы распределения ИПС для различных приложений (справа)

Fig. 6. Dependencies of IS Using AE with 5 Layers on the Attribute Type for the Considered Applications (Left) and from the Histogram of IS Distribution for Different Applications (Right)

* Деятельность Meta Platform Inc. по реализации продуктов – социальных сетей Facebook и Instagram на территории РФ запрещена из-за экстремистской деятельности

Что касается зависимости среднего ИПС от типа приложения, при использовании многослойного АК с 5-ю слоями со структурой 21-14-5-14-21, то она в среднем составляет 15 %, а для структуры 21-14-7-14-21 – около 6 %. Это показывает, что при использовании многослойных АК с 5-ю слоями предпочтение имеет кодировщик, размерность внутреннего слоя которого минимальна.

Сравнение многослойного АК с тремя слоями и структурой 21-5-21 и пятью слоями и структурой 21-14-5-14-21 показывает, что предпочтение следует отдать АК с тремя слоями, в которой выигрыш в среднем составляет 23 %.

Заключение

В работе были исследованы фильтрующие свойства многослойных автокодировщиков в задачах снижения размерности данных мобильных приложений. Основным научным результатом работы является разработка интегрального статистического показателя сжатия, который позволяет количественно оценить изменение разброса атрибутов мобильных приложений после обработки автокодировщиком заданной структуры.

Введенный показатель позволяет оценить эффективность сжатия данных и степень сохранения важной информации. Показано, что чем больше его величина, тем значительней среднеквадратическая ошибка входного показателя выше величины среднеквадратической ошибки выходного показателя, и тем лучше осуществляется сжатие входных данных. Зависимости уменьшения разброса выходных данных многослойных АК от анализируемых атрибутов (см. рисунок 1) показывают, что выигрыш слабо зависит от типа приложения и лежит в среднем в пределах 10…20 % за исключением отдельных атрибутов.

В результате экспериментов было установлено, что трехслойные автокодировщики со структурой 21-5-21 обеспечивают наилучший баланс между сжатием и сохранением информации, достигая уменьшения разброса данных на 20…25 %. Для пятислойных автокодировщиков предпочтение отдается моделям с минимальной размерностью внутреннего слоя, так как они обеспечивают наименьшие потери информации.

Применение разработанного статистического показателя дает возможность сравнивать разные конфигурации автокодировщиков не только по степени уменьшения объема данных, но и по качеству восстановления исходных признаков. Таким образом, проведенное исследование расширяет арсенал методологических средств, доступных специалистам в области анализа данных мобильных приложений, и формирует предпосылки для более осмысленного выбора архитектурных параметров нейросетевых моделей.

Практическая значимость полученных результатов проявляется в возможности адаптации разработанного подхода к реальным приложениям, где ограниченные ресурсы и требования к скорости обработки данных играют ключевую роль. Использование предложенной методологии помогает снижать затраты на хранение и передачу данных, ускорять аналитические операции и, в конечном счете, повышать общую эффективность мобильных сервисов, делая их более производительными и надежными.

Список источников

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. The MIT Press, 2016. 800 p.
2. Hinton G.E., Osindero S., Teh Y.W. A Fast Learning Algorithm for Deep Belief Nets // Neural Computation. 2006. Vol. 18. Iss. 7. PP. 1527–1554. DOI:10.1162/neco.2006.18.7.1527
3. Salakhutdinov R., Hinton G.E. Deep Boltzmann Machines // Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (Clearwater Beach, USA). Proceedings of Machine Learning Research. 2009. Vol. 5. PP. 448–455.
4. Кузьмина М.Г. Многослойные сети-автоэнкодеры в задачах анализа и обработки гиперспектральных изображений // Препринты ИПМ им. М. В. Келдыша. 2021. № 28. 21 с. DOI:10.20948/prepr-2021-28
5. Kramer M.A. Nonlinear principal component analysis using autoassociative neural networks // AIChE Journal. 1991. Vol. 37. Iss. 2. PP. 233–243. DOI:10.1002/aic.690370209
6. Bengio Y., Lamblin P., Popovici D., Larochelle H. Greedy Layer-Wise Training of Deep Networks // In: Advances in Neural Information Processing Systems (B. Schölkopf, J. Platt, T. Hoffman (eds.). Cambridge, 2007. PP. 153–160.
7. Windrim L., Ramakrishnan R., Melkumyan A., Murphy R.J., Chlingaryan A. Unsupervised feature-learning for hyperspectral data with autoencoders // Remote Sensing. 2019. Vol. 11. Iss. 7. P. 864. DOI:10.3390/rs11070864
8. Шелухин О.И., Барков В.В., Симонян А.Г. Обнаружение дрейфа концепта при классификации мобильных приложений с использованием автокодировщиков // Наукоемкие технологии в космических исследованиях Земли. 2023. Т. 15. № 3. С. 20–29. DOI:10.36724/2409-5419-2023-15-3-20-29. EDN:KBWOOG
9. Шелухин О.И., Барков В.В., Маторин Ф.А. Повышение эффективности классификации противоправных и нежелательных приложений в условиях фонового трафика с помощью автокодировщиков // Вестник Санкт-Петербургского государственного университета технологии и дизайна: Серия 1. Естественные и технические науки. 2023. № 3. С. 159–165. DOI:10.46418/2079-8199_2023_3_25. EDN:RLBDBM
10. Ososkov G., Goncharov P. Shallow and deep learning for image classification // Optical Memory and Neural Networks. 2017. Vol. 26. Iss. 4. PP. 221–248. DOI:10.3103/S1060992X1704004X

11. Шелухин О.И., Зегжда Д.П., Раковский Д.И., Самарин Н.Н., Александрова Е.Б. Интеллектуальные технологии информационной безопасности. М.: Горячая линия – Телеком, 2023. 384 с.
12. Шелухин О.И., Ерохин С.Д., Барков В.В. Создание базы данных сетевого трафика для автоматизации классификации мобильных приложений под управлением операционной системы Android // Нейрокомпьютеры: разработка, применение. 2019. № 1. С. 40–51. DOI:10.18127/j19998554-201901-06. EDN:BDDXDT
13. Шелухин О.И., Барков В.В. Экспериментальные исследования и создание базы данных сетевого трафика мобильных устройств под управлением операционной системы Android // Фундаментальные проблемы радиоэлектронного приборостроения. 2018. Т. 18. № 4. С. 1011–1017. EDN:ZABZMT

References

1. Goodfellow I., Bengio Y., Courville A. *Deep Learning*. The MIT Press, 2016. 800 p.
2. Hinton G.E., Osindero S., Teh Y.W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*. 2006;18(7): 1527–1554. DOI:10.1162/neco.2006.18.7.1527
3. Salakhutdinov R., Hinton G.E. Deep Boltzmann Machines. *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (Clearwater Beach, USA). Proceedings of Machine Learning Research, vol.5*. 2009. p.448–455.
4. Kuzmina M.G. Multilayered autoencoders in problems of hyperspectral image analysis and processing. *Preprint M.V. Keldysh IAM*. 2021;28:21. DOI:10.20948/prepr-2021-28
5. Kramer M.A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*. 1991;37(2) 233–243. DOI:10.1002/aic.690370209
6. Bengio Y., Lamblin P., Popovici D., Larochelle H. Greedy Layer-Wise Training of Deep Networks. *In: Advances in Neural Information Processing Systems (B. Schölkopf, J. Platt, T. Hoffman (eds.)*. Cambridge; 2007. p.153–160.
7. Windrim L., Ramakrishnan R., Melkumyan A., Murphy R.J., Chlingaryan A. Unsupervised feature-learning for hyperspectral data with autoencoders. *Remote Sensing*. 2019;11(7):864. DOI:10.3390/rs11070864
8. Sheluhin O.I. Barkov V.V. Simonyan A.G. Concept Drift Detection in Mobile Applications Classification Using Autoencoders. *H&ES Research*. 2023;15(3):20–29. (in Russ.) DOI:10.36724/2409-5419-2023-15-3-20-29. EDN:KBWOOG
9. Sheluhin O.I. Barkov V.V. Matorin F.A. Improving the classification of illegal and unwanted applications under background traffic conditions using autoencoders. *Bulletin of the St. Petersburg State University of Technology and Design: Series 1. Natural and technical sciences*. 2023;3:159–165 (in Russ.) DOI:10.46418/2079-8199_2023_3_25. EDN:RLBDBM
10. Ososkov G., Goncharov P. Shallow and deep learning for image classification. *Optical Memory and Neural Networks*. 2017;26(4):221–248. DOI:10.3103/S1060992X1704004X
11. Sheluhin O.I., Zegzhda D.P., Rakovsk, D.I., Samari, N.N., Aleksandrova E.B. *Intelligent Technologies of Information Security*. Moscow: Goryachaya Liniya – Telecom Publ.; 2023. 384 p. (in Russ.)
12. Sheluhin O.I., Erokhin S.D., Barkov V.V. Creation of a Network Traffic Database for Automating the Classification of Mobile Applications under the Android Operating System. *Neurocomputers: Development, Application*. 2019;1:40–51. (in Russ.) DOI:10.18127/j19998554-201901-06. EDN:BDDXDT
13. Sheluhin O.I., Barkov V.V. Experimental Studies and Creation of a Network Traffic Database of Mobile Devices under the Android Operating System. *Fundamental Problems of Radio Electronic Instrument Engineering*ю 2018;18(4):1011–1017. (in Russ.) EDN:ZABZMT

Статья поступила в редакцию 30.10.2024; одобрена после рецензирования 25.11.2024; принята к публикации 12.12.2024.
The article was submitted 30.10.2024; approved after reviewing 25.11.2024; accepted for publication 12.12.2024.

Информация об авторах:

ШЕЛУХИН Олег Иванович — доктор технических наук, профессор, заведующий кафедрой «Информационная безопасность» Московского технического университета связи и информатики
https://orcid.org/0000-0001-7564-6744

МАТОРИН Фёдор Андреевич — аспирант кафедры «Информационная безопасность» Московского технического университета связи и информатики
https://orcid.org/0009-0002-4897-2338

Авторы сообщают об отсутствии конфликтов интересов.
The authors declare no conflicts of interests