УДК 004.032.26:004.67:004.62:004.056.55 DOI: 10.18698/0236-3933-2021-4-109-121

# ОРГАНИЗАЦИЯ ЗАЩИТЫ КОНФИДЕНЦИАЛЬНЫХ СВЕДЕНИЙ, СОДЕРЖАЩИХСЯ В НАБОРАХ ДАННЫХ, ПРИ ОБУЧЕНИИ НЕЙРОННЫХ СЕТЕЙ НА УДАЛЕННЫХ ЭВМ И В ОБЛАЧНЫХ СЕРВИСАХ

**С.С. Тарасенко** dor7la96@mail.ru
**С.В. Морковин** msw-c@ya.ru

**Академия ФСО России, г. Орел, Российская Федерация**

**Аннотация**

Различные компании и исследователи вкладывают значительные материальные и временные ресурсы в сбор необходимых наборов данных и, как следствие, желают сохранить их в тайне. Для анализа собранных наборов данных и их дальнейшего применения при обучении нейронных сетей и выполнении конкретных задач необходимо соответствующее аппаратное обеспечение, приобрести которое имеют возможность не все исследователи в области машинного обучения. Для решения этой проблемы многие IT-корпорации, например Amazon или Google, предоставляют доступ к своей мощной аппаратной инфраструктуре (на возмездной и безвозмездной основе) с высокой вычислительной мощностью для обучения нейронных сетей. Семантика наборов данных, на которых будет происходить обучение, открыта. Поэтому возникает необходимость защиты семантики данных, на которых будет происходить обучение нейронных сетей в облачных сервисах или на удаленных ЭВМ

**Ключевые слова**

*Нейронные сети, защита данных, облачные вычисления, шифрование, генеративные состязательные сети*

Поступила 08.02.2021
Принята 04.03.2021


**Введение.** В настоящее время алгоритмы машинного обучения и нейронные сети используются для решения многих задач в разных сферах. Частные компании, различные лаборатории, разработчики программного обеспечения используют (или хотели бы использовать) в своей работе нейронные сети для анализа большого объема информации. Однако обучение даже простой нейронной сети на компьютере с центральным процессором (central processing unit, CPU) может занять много времени, а покупку качественных графических процессоров (graphics processing unit, GPU), на которых процесс обучения нейронных сетей будет приемлемым

по времени, могут позволить далеко не все субъекты-исследователи. В настоящее время у этой проблемы есть решение — компании, лидирующие в сфере информационных технологий, предоставляют доступ к своим удаленным графическим процессорам за денежную плату или в ограниченном варианте безвозмездно, например, компания Google с ее проектом Google Colaboratory [1]. Это открывает большие возможности для исследователей технологий машинного обучения и искусственного интеллекта, а также отлично подходит для образовательных целей. Но, если какая-либо лаборатория будет стоять на пороге нового открытия или же частная компания будет разрабатывать модель нейронной сети, которая будет обучаться на персональных данных ее клиентов, то эти данные не должны разглашаться. Возникает необходимость организовать защиту данных при обучении в облачных сервисах.

*Цель работы* — создать способ защиты семантики данных при обучении нейронных сетей в облачных сервисах и на удаленных ЭВМ. Для достижения поставленной цели необходимо проанализировать существующие способы защиты данных при облачных вычислениях; разработать способ, обеспечивающий достижение поставленной цели; выбрать операционную систему (ОС) для разработки программного обеспечения при реализации разработанного способа; выбрать среду разработки и язык программирования для создания комплекса программного обеспечения; практически реализовать комплекс программного обеспечения для разработанного способа; разработать предложения по практическому применению предлагаемого способа.

**Основная часть.** Одним из известных способов защиты данных при облачных вычислениях является гомогенное шифрование [2]. Поскольку оно основано на асимметричном шифровании [3], то криптографическая стойкость такой системы основана на математической сложности процесса дешифрования. Теоретическая вероятность расшифровки подобных систем возможна, а с темпами развития технологий, которые имеются сегодня (квантовые компьютеры, вычислительная мощность которых в 10 000 раз больше, чем у самого мощного классического суперкомпьютера [4]), в скором будущем асимметричное шифрование может вовсе потерять актуальность, так как его дешифрование станет возможным за разумный период времени. Необходим поиск альтернативы защиты данных в облачных вычислениях с помощью гомогенного шифрования.

Основная идея предлагаемого альтернативного способа защиты семантики данных — это замена реального набора данных ложным и обучение целевой нейронной сети на нем в облачном сервисе. Ложный набор данных

создается генератором генеративной состязательной сети (generative adversarial networks, GAN) [5], обученным на реальном наборе данных на локальной ЭВМ.

Для наиболее наглядной демонстрации способа защищаемыми данными для обучения выбраны изображения из набора данных рукописных цифр Modified National Institute and Technology (MNIST) [6] (по 2000 изображений каждого класса для обучения, по 1000 изображений каждого класса для тестирования и по 1000 изображений каждого класса для верификации). Экземпляры классов изображений приведены на рис. 1.

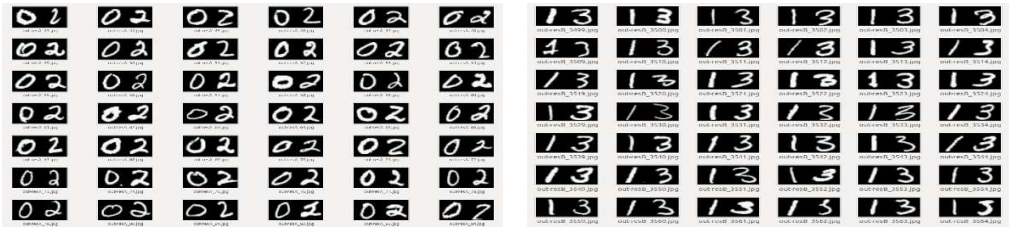


**Рис. 1.** Часть набора данных MNIST

Для программной разработки комплекса, обеспечивающего функционирование предлагаемого способа, в качестве целевой ОС, на которой будет вестись разработка, выбрана ОС Ubuntu 20.04 LTS [7], являющаяся бесплатной и свободной для распространения. Для более удобной реализации алгоритмов машинного обучения и построения нейронных сетей в качестве языка программирования выбран Python 3.6, так как на нем уже реализовано множество библиотек в области обработки больших наборов данных и машинного обучения. Для реализации программ использованы бесплатные и свободные для распространения библиотеки scikit-learn [8] и Keras [9]. В качестве облачной платформы для обучения нейронных сетей выбрана платформа Google Colaboratory от компании Google, которая предоставляет посредством этой платформы бесплатный доступ к удаленному GPU, и интерактивная вычислительная среда Jupyter Notebook.

Наиболее подходящим видом сети GAN для решения данной задачи является GAN Pix2Pix [10].

Полная система, реализующая предлагаемый способ, будет состоять из нескольких частей. Первая — это генератор GAN, обученный на локальной машине (рис. 2), задачей которого является подменять реальные изображения ложными, но при этом оставляя связь между ними. Пример: реальными данными являются два класса изображений: изображения «0» и изображения «1». Генератор переводит все возможные изображения «0» в изображения «2», а все изображения «1» в изображения «3». Однако по-

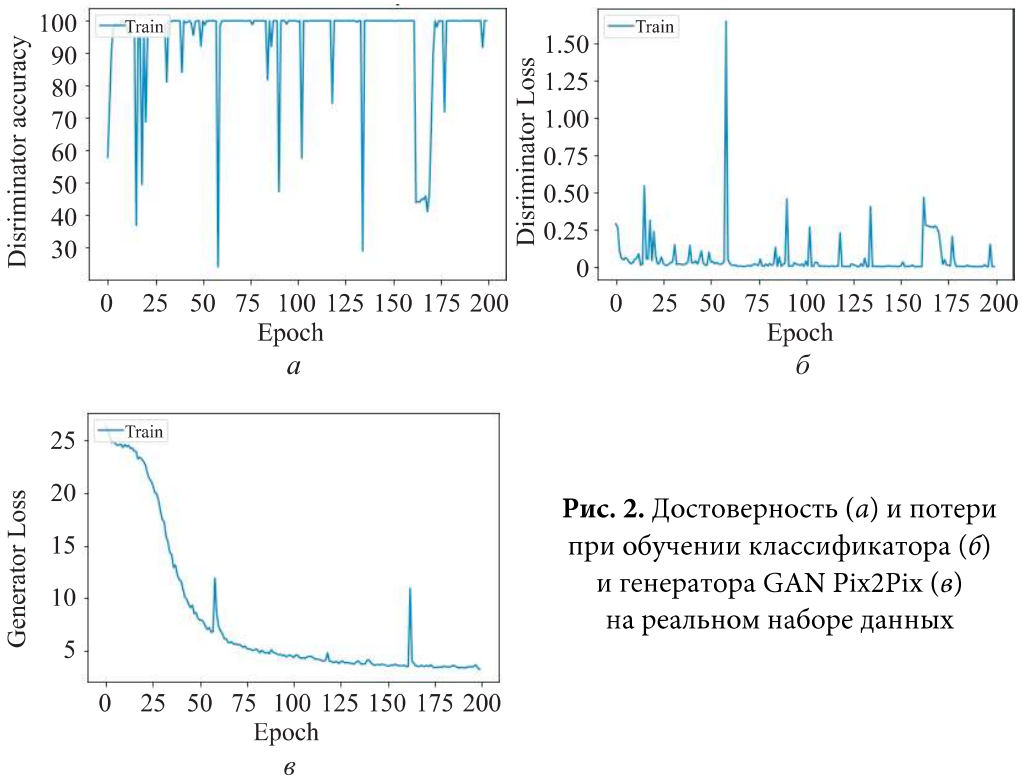



**Рис. 2.** Достоверность (*а*) и потери при обучении классификатора (*б*) и генератора GAN Pix2Pix (*в*) на реальном наборе данных

лучить исходное изображение невозможно математическими методами (только полным перебором всех возможных вариантов функции, обратной генератору). Учитывая сказанное, генератор GAN — это функция, хранимая в секрете, без знания которой нельзя однозначно построить функцию, обратную этому генератору.

Следующим шагом после обучения генератора является генерация им из реального набора данных ложного набора для дальнейшего обучения на нем целевой нейронной сети в облачном сервисе (рис. 3).

**Рис. 3.** Пример набора ложных данных MNIST, созданных генератором GAN Pix2Pix

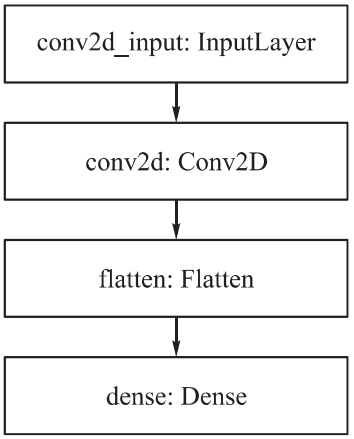



**Рис. 4.** Архитектура выбранного классификатора

В качестве примера целевой нейронной сети, обучающейся в облачном сервисе, рассматривается нейронная сеть-классификатор на основе сверточной нейронной сети [11] (рис. 4), состоящей из входного сверточного 2D-слоя размером $M \times N \times C$, где $M$ и $N$ — высота и ширина входного изображения, а $C$ — число цветовых каналов в изображении. Нейроны в данном слое имеют функцию активации ReLU. Ядро свертки имеет размер $5 \times 5$, число фильтров 32. Далее следует плоский слой нейронной сети и далее выходной слой нейронной сети — полносвязный слой с числом нейронов, равным числу вариантов результирующих классов, и функцией активации Softmax (наиболее часто используемой в задачах классификации в качестве функции активации выходного слоя нейронной сети). Оптимизатор обучения сети — Adam, функция потерь — Categorical_crossentropy. Работа сети заключается в предсказании класса входного изображения из ложного набора данных (изображения «2» или «3»).

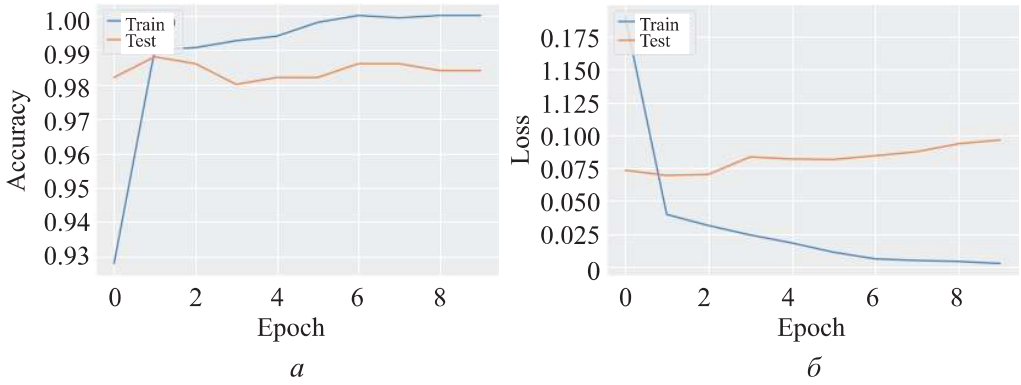
Ложный набор данных загружается в Google Colaboratory и на нем осуществляется обучение классификатора.

Сверточная нейронная сеть обучается на ложном наборе данных. Результаты ее обучения оцениваются на тестовой выборке ложных изображений, которая не была задействована при обучении нейронной сети: точность предсказания составляет 0,99, показатель функции потерь равен 0,0381. Это свидетельствует о том, что нейронная сеть выявила необходимые закономерности для корректной классификации изображений «2» и «3», а не запомнила все варианты входных и выходных данных тренировочной выборки (рис. 5).

Поскольку при обучении на тренировочном наборе данных и оценке работы нейронной сети на тестовой выборке число изображений для каждого класса одинаковое, то в качестве метрики оценки качества обучения используют

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{1}$$

где $TP$ (True Positive) и $TN$ (True Negative) — истинно позитивные и негативные предсказания (варианты верных предсказаний); $FP$ (False Positive)




*а* *б*

```
16/16 [==============================] - 0s 3ms/step - loss: 0.0381 - accuracy: 0.9900
Evaluate on fake "2" and "3" --> Loss, Accuracy: [0.03814179077744484, 0.9900000095367432]
```

*в*

**Рис. 5.** Достоверность (*а*), потери (*б*) и оценка (*в*) обучения классификатора на ложном наборе данных

и *FN* (False Negative) — ложнопозитивные и ложнонегативные предсказания (варианты ложных предсказаний).

Далее обученная в облачном сервисе модель вместе с весами модели загружается в локальную машину. Полный цикл обработки изображения для предсказания класса на локальной машине следующий: реальное изображение (неизвестного для системы класса) передается на обработку генератору GAN; генератор формирует ложное изображение и передает на обработку классификатору, обученному в облачном сервисе (рис. 6). Предсказанный результат будет коррелировать с классом исходного изображения (рис. 7 и 8).

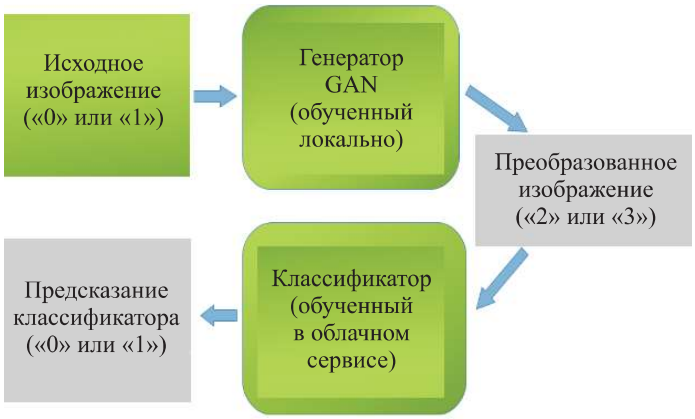



**Рис. 6.** Полный цикл обработки изображения

```
47/47 [==============================] - 0s 3ms/step - loss: 0.3592 - accuracy: 0.9440
Evaluate on real "2" and "3" --> Loss, Accuracy: [0.35924527049064636, 0.9440000057220459]
```

**Рис. 7.** Результат оценки классификатором на реальных данных MNIST для обучения

```
1/1 [==============================] - 0s 4ms/step
Our predict on (0_1.png). Исходное изображение ("0_1.jpg") =  0
(1, 32, 32, 3)
1/1 [==============================] - 0s 2ms/step
Our predict on (0_2.png). Исходное изображение ("0_2.jpg") =  0
(1, 32, 32, 3)
1/1 [==============================] - 0s 2ms/step
Our predict on (0_3.png). Исходное изображение ("0_3.jpg") =  0
(1, 32, 32, 3)
1/1 [==============================] - 0s 2ms/step
Our predict on (1_1.png). Исходное изображение ("1_1.jpg") =  1
(1, 32, 32, 3)
1/1 [==============================] - 0s 3ms/step
Our predict on (1_2.png). Исходное изображение ("1_2.jpg") =  1
(1, 32, 32, 3)
1/1 [==============================] - 0s 1ms/step
Our predict on (1_3.png). Исходное изображение ("1_3.jpg") =  1
```

**Рис. 8.** Предсказанные системой результаты (6 из 6 верно)

У предлагаемого способа есть недостаток — это потери при обработке генератором реального изображения (см. рис. 7 и 8). Но для многих изображений данный уровень потерь является допустимым. Также при условии абсолютной случайности величин гиперпараметров генератора GAN предлагаемый способ можно назвать теоретически недешифруемым (ТНДШ) системой в области нейронных сетей и машинного обучения. Поскольку ложное изображение могло быть получено из любого изображения, то получить исходное из ложного возможно, только подобрав функцию, обратную генератору. Добиться этого без знания генератора можно только перебором всех возможных значений генераторов (с неизвестными гиперпараметрами: число и вид слоев, число нейронов в каждом слое, различные коэффициенты обучения и т. д.). В итоге после перебора получится полный набор всех возможных изображений, что дает право, с определенной долей уверенности, говорить о том, что данный способ, по сути, является ТНДШ системой в мире нейронных сетей и машинного обучения.

Проекция теоремы Шеннона о совершенном шифре [12], выполнение условий которой допускает именование способа шифрования как ТНДШ системы, принимает в рассматриваемой абстракции следующий вид: условие равновероятности ключа заключается в том, что знание зашифрованного сообщения, в рассматриваемом примере это изображение ложной «2» или «3», не дает никакой информации об исходном изображении, «0» или «1» из которого было получено ложное. Теоретически данное условие можно считать выполненным, если удастся с помощью ключа такой же

размерности, как и тот, на котором были получены ложные изображения, получить такие же ложные данные. Поскольку скрываемая информация — не конкретная реализация какого-либо изображения, а класс изображения («0» или «1»), то достаточно получить из другого набора данных ложное изображение такого же класса (в рассматриваемом примере из «1» получить «2» или из «0» получить «3»). Для доказательства выполнения этого условия на практике реализована GAN такой же архитектуры и размерности, как исходная, получающая на вход изображения «0» и «1» и переводящая их в изображения ложных «3» и «2» (рис. 9).

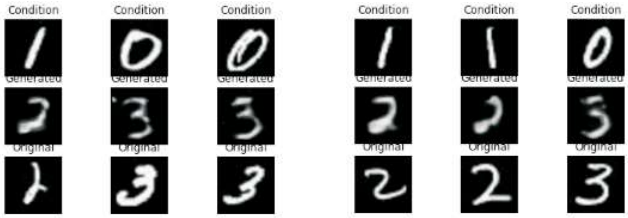



**Рис. 9.** Пример работы новой сети GAN

Вторым условием теоремы Шеннона о совершенном шифре является следующее выражение: число вариантов ключа (в рассматриваемом примере 41 843 331 вариант сети генератора с определенным числом гиперпараметров, рис. 10) должно быть больше либо равно числу вариантов возможных входных сообщений (к примеру, GrayScale изображений размером 256 × 256 пикселей — увеличенное в размере классическое изображение из набора данных MNIST).

```
Total params: 41,843,331
Trainable params: 41,834,499
Non-trainable params: 8,832
_________________________________________________________________
Generator_GAN_summary:/n None
```

**Рис. 10.** Число параметров сети GAN

Математическое доказательство выполнения второго условия можно представить в следующем виде: теоретически гиперпараметры могут принимать любые значения (зависит от выбранных функций активации нейронов сети). Для определенности выбрано допустимое значение гиперпараметров сети в пределах от нуля до единицы с точностью float64 (для хранения одного числа в памяти компьютера отводится 8 байт) [13], следовательно, значение гиперпараметра может иметь 16 цифр после запятой, причем каждая цифра может принимать 10 значений (от 0 до 9), т. е. $10^{16}$. Отсюда следует, что теоретически возможное число вариантов

рассматриваемой в данной статье сети генератора равняется $N_{\text{сетей}} = 41\,843\,331^{10^{16}} = 10^{7,62^{10^{16}}}$, а возможное число вариантов изображений равно $N_{\text{изображений}} = (256 \cdot 256)^{256} = 10,5 \cdot 10^{10^3}$. Поскольку $N_{\text{сетей}} \gg N_{\text{изображений}}$, то второе условие можно считать выполненным. Однако на практике все гиперпараметры взаимосвязаны. Рассчитанное число $N_{\text{сетей}}$ является теоретическим. Это дает право выдвинуть гипотезу о том, что предложенный способ является ТНДШ системой, для подтверждения которой необходимо практически доказать второе условие теоремы Шеннона и рассчитать число практически возможных вариантов сети. Данный вопрос требует более детальной проработки и является предметом дальнейших исследований.

Отметим, что представленный способ может быть альтернативой гомогенному шифрованию, особенно в долгосрочной перспективе.

**Выводы.** Таким образом, в результате проведения исследования все поставленные задачи решены в полном объеме, цель работы достигнута — разработан способ защиты семантики данных при обучении на них нейронных сетей в облачных сервисах.

Основными достоинствами разработанного способа являются его простота и, гипотетически, абсолютная криптографическая стойкость ложного набора данных и возможность получения исходных изображений из ложных лишь путем полного перебора всех возможных функций, обратных генератору GAN, что в результате даст полный набор изображений, которые теоретически возможно получить из ложного набора данных.

Недостатками данного метода являются: необходимость обучения генератора GAN на локальной машине (что также требует больших временных затрат) и потери при обучении генератора (зависят от качества подобранных гиперпараметров и времени обучения генератора).

Резюмируя сказанное, можно сделать вывод о том, что предложенный способ целесообразно применять, когда времени на обучение целевой нейронной сети (в рассматриваемом примере это сеть классификатора) будет затрачено больше, чем на обучение генератора GAN. Рассмотренный в статье классификатор к таким сетям не относится, однако он наглядно демонстрирует основную идею предлагаемого способа. На практике же предложенный способ рекомендуется использовать при поиске наилучших гиперпараметров нейронных сетей, где с одним и тем же набором данных необходимо проводить большое число опытов, или, например, при обучении рекуррентных нейронных сетей (или их различных вариаций, таких как управляемые рекуррентные блоки (gated recurrent unit, GRU) [14] или

долгой краткосрочной памяти (long short-term memory, LSTM) [15], в которых имеют значение временные ряды анализируемых данных, на обучение которых требуется много времени.

Полученный в результате выполнения научной работы программный комплекс дает возможность на практике обучать нейронные сети на защищаемых наборах данных на удаленных ЭВМ и в облачных сервисах, не раскрывая семантики этих данных, тем самым расширяя арсенал возможностей многих исследователей в области машинного обучения и искусственного интеллекта. Принимая во внимание, что данный способ гипотетически имеет абсолютную криптографическую стойкость, его можно в будущем рассматривать как достойную альтернативу способу защиты данных при облачных вычислениях, основанному на гомогенном шифровании.

## ЛИТЕРАТУРА

[1] Google colaboratory: *веб-сайт*. URL: https://colab.research.google.com (дата обращения: 15.09.2021).

[2] Gentry C. Fully homomorphic encryption using ideal lattices. *Proc. STOC*, 2009, pp. 169–178. DOI: https://doi.org/10.1145/1536414.1536440

[3] Саломаа А. Криптография с открытым ключом. М., Мир, 1995.

[4] Arute F., Arya K., Babbush R., et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 2019, vol. 574, pp. 505–510.
DOI: https://doi.org/10.1038/s41586-019-1666-5

[5] Goodfellow I., Pouget-Abadie J., Mirza M., et al. Generative adversarial networks. *Proc. NIPS*, 2014, pp. 2672–2680.

[6] LeCun Y., Cortez C., Burges C.J.C. The MNIST database of handwritten digits. *yann.lecun.com: веб-сайт*. URL: http://yann.lecun.com/exdb/mnist (дата обращения: 15.09.2021).

[7] Официальный сайт дистрибутива Ubuntu 20.04 LTS.
URL: https://releases.ubuntu.com/20.04 (дата обращения: 15.09.2021).

[8] Официальный сайт библиотеки scikit-learn.
URL: https://scikit-learn.org/stable (дата обращения: 15.09.2021).

[9] Официальный сайт библиотеки Keras. URL: https://keras.io (дата обращения: 15.09.2021).

[10] Isola P., Zhu J.Y., Zhou T., et al. Image-to-image translation with conditional adversarial networks. URL: https://arxiv.org/abs/1611.07004 (дата обращения: 15.09.2021).

[11] Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks. *Proc. NIPS*, 2012, vol. 1, pp. 1097–1105.

[12] Шеннон К. Теория связи в секретных системах. В: Работы по теории информации и кибернетике. М., ИЛ, 1963, с. 243–322.

[13] Standard ECMA-262. ECMAScript language specification. Geneva, Ecma International, 2011.

[14] Cho K., van Merrienboer B., Gulcehre C., et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation.
URL: https://arxiv.org/abs/1406.1078 (дата обращения: 15.09.2021).

[15] Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Comput.*, 1997, vol. 9, no. 8, pp. 1735–1780. DOI: https://doi.org/10.1162/neco.1997.9.8.1735

**Тарасенко Сергей Сергеевич —** сотрудник Академии ФСО России (Российская Федерация, 302015, г. Орел, ул. Приборостроительная, д. 35).

**Морковин Сергей Владимирович —** сотрудник Академии ФСО России (Российская Федерация, 302015, г. Орел, ул. Приборостроительная, д. 35).

**Просьба ссылаться на эту статью следующим образом:**

Тарасенко С.С., Морковин С.В. Организация защиты конфиденциальных сведений, содержащихся в наборах данных, при обучении нейронных сетей на удаленных ЭВМ и в облачных сервисах. *Вестник МГТУ им. Н.Э. Баумана. Сер. Приборостроение*, 2021, № 4 (137), с. 109–121.

DOI: https://doi.org/10.18698/0236-3933-2021-4-109-121

## ORGANIZING THE PROTECTION OF CONFIDENTIAL INFORMATION CONTAINED IN DATASETS WHEN TRAINING NEURAL NETWORKS ON REMOTE ELECTRONIC COMPUTERS AND CLOUD SERVICES

**S.S. Tarasenko** dor7la96@mail.ru
**S.V. Morkovin** msw-c@ya.ru

**FSS Academy of Russia, Orel, Russian Federation**

**Abstract**

Various companies and researchers invest significant material and time resources to collect the necessary datasets, and as a consequence, wish to keep them secret. To analyze the collected datasets and further apply them in the training of neural networks for specific tasks requires the appropriate hardware, which not all machine learning researchers are able to acquire. To solve this problem, many IT corporations, such as Amazon or Google, provide access to their powerful hardware infra-

**Keywords**

*Neural networks, data protection, cloud computing, encryption, generative adversarial networks*

structure (on a reimbursable and non-reimbursable basis) with high computing power for training neural networks. The semantics of the datasets on which the training of neural networks will take place is open. Therefore, there is a need to protect the semantics of the data on which neural networks will be trained in the cloud services or on remote computers

Received 08.02.2021
Accepted 04.03.2021


## REFERENCES

[1] Google colaboratory: *website*. Available at: https://colab.research.google.com (accessed: 15.09.2021).

[2] Gentry C. Fully homomorphic encryption using ideal lattices. *Proc. STOC*, 2009, pp. 169–178. DOI: https://doi.org/10.1145/1536414.1536440

[3] Salomaa A. Public-key cryptography. New York, Springer, 1990.

[4] Arute F., Arya K., Babbush R., et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 2019, vol. 574, pp. 505–510.

DOI: https://doi.org/10.1038/s41586-019-1666-5

[5] Goodfellow I., Pouget-Abadie J., Mirza M., et al. Generative adversarial networks. *Proc. NIPS*, 2014, pp. 2672–2680.

[6] LeCun Y., Cortez C., Burges C.J.C. The MNIST database of handwritten digits. *yann.lecun.com: website*. Available at: http://yann.lecun.com/exdb/mnist (accessed: 15.09.2021).

[7] Ofitsial'nyy sayt distributiva Ubuntu 20.04 LTS [Official website of Ubuntu 20.04 LTS]. Available at: https://releases.ubuntu.com/20.04 (accessed: 15.09.2021).

[8] Ofitsial'nyy sayt biblioteki scikit-learn [Official website of Scikit-Learn Library]. Available at: https://scikit-learn.org/stable (accessed: 15.09.2021).

[9] Ofitsial'nyy sayt biblioteki Keras [Official website of Keras Library].
Available at: https://keras.io (accessed: 15.09.2021).

[10] Isola P., Zhu J.Y., Zhou T., et al. Image-to-image translation with conditional adversarial networks. Available at: https://arxiv.org/abs/1611.07004 (accessed: 15.09.2021).

[11] Krizhevsky A., Sutskever I., Hinton G.E. Imagenet classification with deep convolutional neural networks. *Proc. NIPS*, 2012, vol. 1, pp. 1097–1105.

[12] Shannon C.E. Communicftion theory of secrecy systems. *BSTJ*, 1949, vol. 28, no. 4, pp. 656–715.

[13] Standard ECMA-262. ECMAScript language specification. Geneva, Ecma International, 2011.

[14] Cho K., van Merrienboer B., Gulcehre C., et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation.
Available at: https://arxiv.org/abs/1406.1078 (accessed: 15.09.2021).

[15] Hochreiter S., Schmidhuber J. Long short-term memory. *Neural Comput.*, 1997, vol. 9, no. 8, pp. 1735–1780. DOI: https://doi.org/10.1162/neco.1997.9.8.1735

**Tarasenko S.S.** — Employee, FSS Academy of Russia (Priborostroitelnaya ul. 35, Orel, 302015 Russian Federation).

**Morkovin S.V.** — Employee, FSS Academy of Russia (Priborostroitelnaya ul. 35, Orel, 302015 Russian Federation).

**Please cite this article in English as:**

Tarasenko S.S., Morkovin S.V. Organizing the protection of confidential information contained in datasets when training neural networks on remote electronic computers and cloud services. *Herald of the Bauman Moscow State Technical University, Series Instrument Engineering*, 2021, no. 4 (137), pp. 109–121 (in Russ.).
DOI: https://doi.org/10.18698/0236-3933-2021-4-109-121