УДК 004.056

Н. Р. Воронина[1]✉, А. А. Киселев[2], В. М. Белов[1]

Нейросетевая модель классификатора объектов КИИ

[1]Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
[2]Новосибирский государственный технический университет,
г. Новосибирск, Российская Федерация
e-mail: nata.voronina.00@inbox.ru

Аннотация. В статье рассматривается способ применения нейросетевых технологий для классификации объектов критической информационной инфраструктуры (ОКИИ). Обеспечение безопасности ОКИИ является одним из важнейших элементов информационной безопасности Российской Федерации, что доказывает множество федеральных законов и нормативных правовых актов в данной сфере. От скорости и точности присвоения категории значимости ОКИИ зависит обеспечение его безопасности. В статье рассматриваются необходимый функционал классификатора – возможность обучения на основе новых данных, классификации файлов с описанием ОКИИ, вариант используемых библиотек для машинного обучения, таких как scikit-learn, PyPDF2 и docx, обработка текста, включающая приведение к нижнему регистру и удаление стоп-слов. Также рассматривается проблема, возникающая в процессе создания нейросетевой модели классификатора – нехватка данных для обучения модели. Способы решения этой проблемы – использование аугментации и взаимодействие с органом-регулятором в сфере КИИ.

Ключевые слова: искусственный интеллект, машинное обучение, объекты критической информационной инфраструктуры

N. R. Voronina[1]✉, A. A. Kiselyov[2], V. M. Belov[1]

Neural network model of the CII object classifier

[1]Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
[2]Novosibirsk State Technical University, Novosibirsk, Russian Federation
e-mail: nata.voronina.00@inbox.ru

Abstract. The article discusses the method of using neural network technologies in the classification of critical information infrastructure facilities (CIIF). Ensuring the security of CIIF is one of the most important elements of the information security of the Russian Federation, which is proved by many federal laws and regulations in this area. The safety of this facility depends on the speed and accuracy of assigning a category of significance to an object. The article discusses the necessary functionality of the classifier – the ability to learn from new data and classify files with descriptions of AI objects, a variant of used libraries for machine learning such as scikit-learn, PyPDF2 and docx, and text processing, including lowercase reduction and removal of stop words. The problem that arises in the process of creating a classifier neural network model is also considered – the lack of data for training the model, and the demand for solutions to this problem is the use of augmentation and interaction with the regulatory authority in the field of CII.

Keywords: artificial intelligence, machine learning, critical information infrastructure facilities

Введение

С момента появления в Доктрине информационной безопасности Российской Федерации [1] тезиса о повышении защищенности КИИ законодательство в сфере обеспечения безопасности ОКИИ продолжает активно дополняться, о чем свидетельствует Федеральный закон N 58-ФЗ «О внесении изменений в Федеральный закон «О безопасности критической информационной инфраструктуры Российской Федерации», принятый 07.04.2025 и вступающий в силу в сентябре 2025 года [2]. Это приводит к пониманию, что в настоящее время безопасность КИИ является одним из важнейших вопросов безопасности информации страны.

В Федеральном законе «О безопасности критической информационной инфраструктуры Российской Федерации» для ОКИИ определяются три категории значимости [3]. Присвоение категории значимости запускает процесс обеспечения безопасности объекта, поэтому скорость определение и присвоение категории, а также точность, напрямую влияют на то, насколько быстро и правильно будет обеспечена защищенность объекта от угроз информационной безопасности.

В данный момент не существует автоматизированного классификатора ОКИИ, который имел бы достаточный уровень точности и скорости для решения проблем категорирования. Классификатор на основе нейросетевой модели может стать одним из решений, поэтому целью статьи является обзор возможного варианта применения нейросетевых технологий для решений проблемы классификации ОКИИ.

Методы и материалы

Для ручной классификации ОКИИ используется информация об объектах, представленная текстом: критические процессы, архитектуры объектов, программно-аппаратные средства и прочее. Информация, необходимая для категорирования, определяется в Постановлении правительства №127 «Об утверждении Правил категорирования объектов критической информационной инфраструктуры Российской Федерации, а также перечня показателей критериев значимости объектов критической информационной инфраструктуры Российской Федерации и их значений» [4]. Из этого можно сделать вывод, что автоматизированный классификатор ОКИИ на основе нейросетевой модели должен обрабатывать текст как на этапе обучения, так и на этапе классификации нового объекта на основе его описания. Рассмотрим реализацию алгоритма классификации на языке Python [5]. В нем представлено множество библиотек для работы с нейронными сетями, что значительно облегчает создание нейросетевых классификаторов [6].

Для обучения нейронных сетей используются базы данных с уже размеченными данными по классам, следовательно, для корректной классификации нам необходимо иметь базу данных с описанием ОКИИ, которым уже присвоены категории значимости. На этом этапе мы сталкиваемся с проблемой нехватки информации, так как большая часть информации о КИИ имеет конфиденциальный

характер. В таких условиях методы глубокого обучения сразу отпадают, так как при обучении на небольшом количестве данных может возникнуть проблема переобучения. Следовательно, для работы лучше использовать классические алгоритмы: логистическая регрессия или деревья решений. Метод дерева решений склонен к переобучению и неустойчив к шуму, поэтому выбор пал на использование метода логистической регрессии [7]. В языке программирования Python использовать метод логистической регрессии позволяет библиотека scikit-learn.

Классификатор на основе нейросетевой модели должен иметь две функции: обучение на подготовленной базе данных и загрузку нового файла для классификации ОКИИ. Для обучения в программу будет загружаться файл CSV формата, так как это универсальный способ представления тектовых данных для обучения и поддерживаемый всеми библиотеками [8]. Возможность обучить классификатор на основе новых данных позволяет в дальнейшем обновлять его в соответствии с изменениями в законодательстве. Также в программе должна быть реализована возможность единоразового обучения, то есть полученая модель должна сохраняться для дальнейшего использования. Это облегчит использование классификатора и позволит сэкономить время. Для классификации будем загружать файлы двух самых распространенных форматов: DOC и PDF, поэтому используем библиотеки для обработки этих форматов – PyPDF2 и docx [9].

Процесс обучения нейросетевой модели можно разделить на несколько этапов: обработка текста из датасета и само обучение.

Обработка текста необходима для создания «чистого» датасета. Изначально датасет может содержать ошибки, которые могут повлиять на результаты обучения и в дальнейшем снизить эффективность классификации [10]. Обработка текста включает в себя: приведение к нижниму регистру, удаление стоп-слов, удаление строк с пустыми значениями, удаление выбросов и знаков препинания. Для большей части этих задач может быть использована стандартная библиотека Python для обработки текста – string, но для удаления стоп-слов можно использовать библиотеку nltk [11]. Датасет делится на тестовую и тренировочную части, где тестовая составляет 20 % от всего датасета.

Далее текст проходит процесс векторизации. В библиотеке scikit-learn есть несколько методов: CountVectorizer, TfidfVectorizer и HashingVectorizer. CountVectorizer не учитывает важность слов, т.е. все слова будут считаться равнозначными, а HashingVectorizer предназначен для больших массивов данных, поэтому стоит использовать TfidfVectorizer [12].

После обработки и векторизации текста, можно перейти к этапу обучения модели. Подготовленный датасет должен содержать список характерных черт ОКИИ, размеченный по категориям значимости. Модель обучается на тренировочной части и тестируется на тестовой, после чего модель сохраняется для дальнейшего использования.

Когда нейросетевая модель классификатора ОКИИ обучена, можно загрузить новый файл с описанием ОКИИ, также произвести обработку и векторизацию и произвести процесс классификации.

Результаты

В результате структура классификатора ОКИИ на основе нейросетевой модели будет следующей, позволяющей как обучить модель, используя подготовленные данные, так и производить классификацию нового файла (рис. 1).

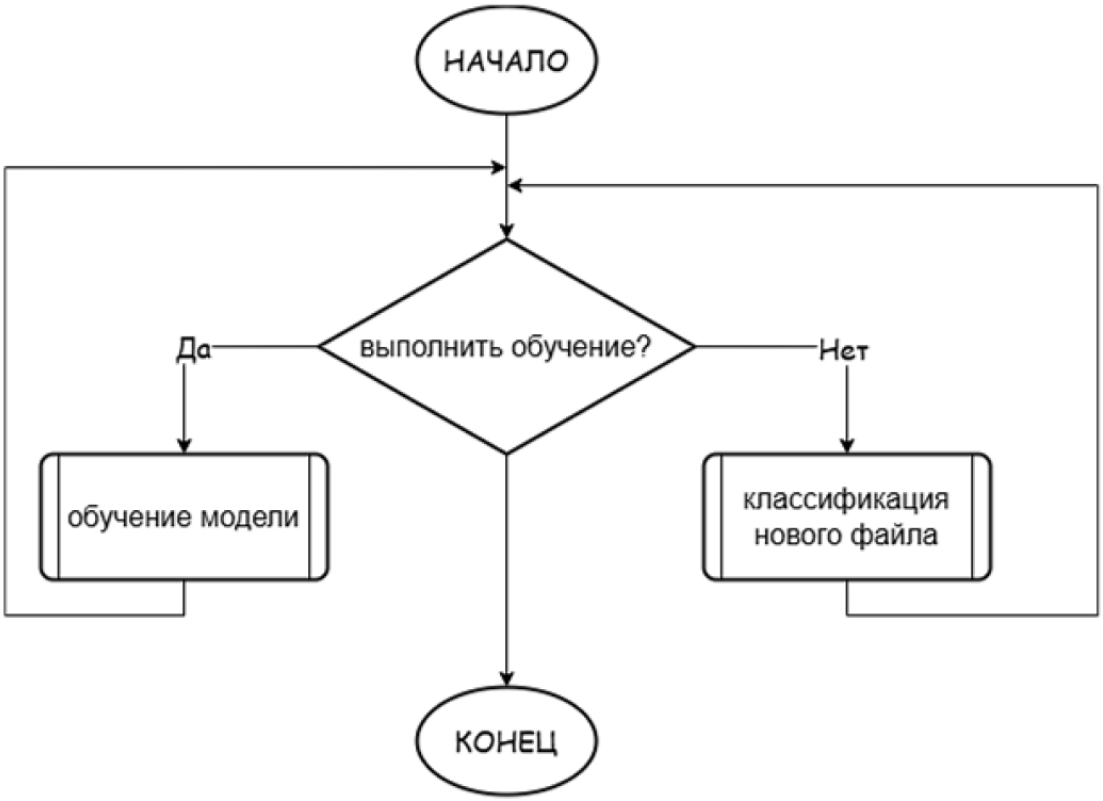

Рис. 1. Структура нейросетевой модели классификатора

Процесс обучения также делится на этапы. Обучение осуществляется с использованием логистической регрессии с помощью библиотеки scikit-learn. Для векторизации текста применяется метод TfidfVectorizer. Полученые модель и векторизатор сохраняются для дальнейшего использования. Этапы обучения представлены на блок-схеме (рис. 2).

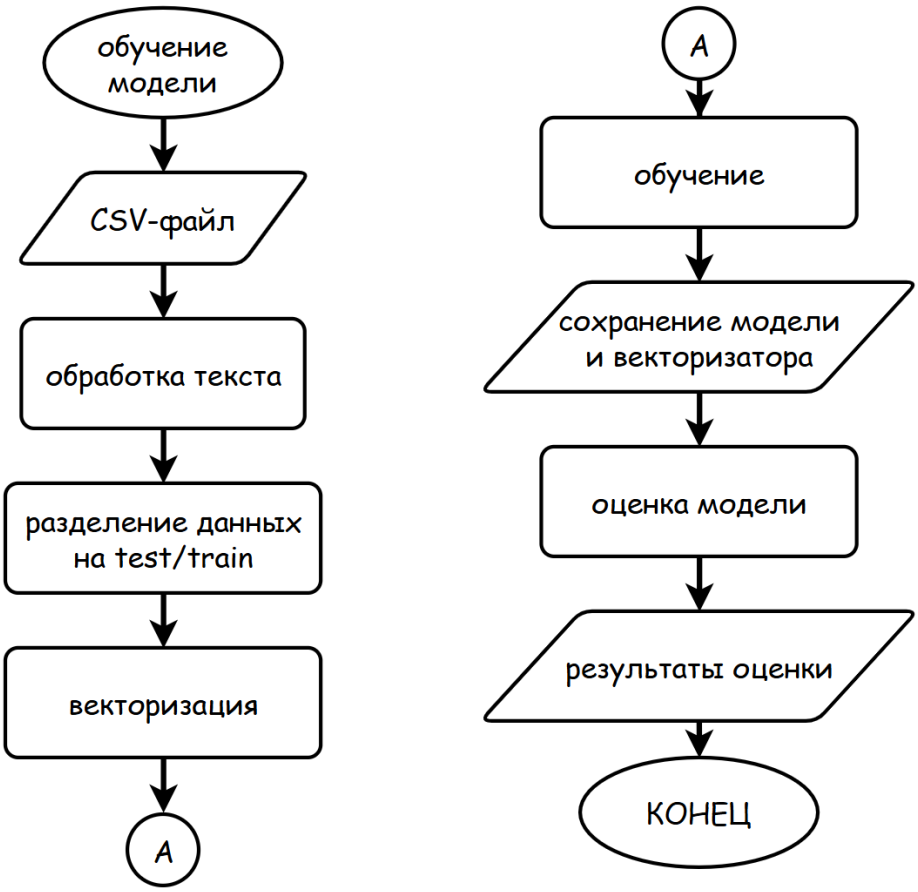

Рис. 2. Процесс обучения нейросетевой модели

Далее классификатор должен иметь функцию загрузки и обработки файлов DOC и PDF для классификации новых объектов КИИ. Это можно реализовать с помощью библиотек PyPDF2 и docx. Процесс классификации нового файла представлен на следующей блок-схеме (рис. 3).

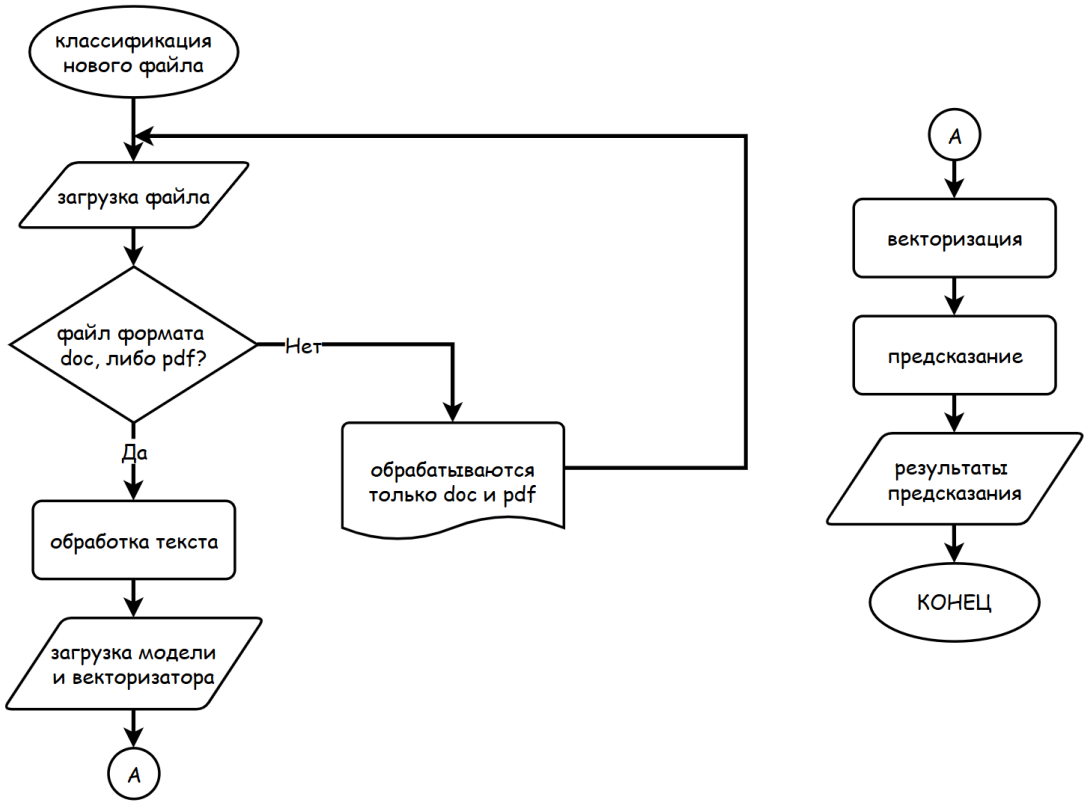

Рис. 3. Процесс классификации нового файла

Обсуждение

Главной проблемой создания классификатора на основе нейросетевой модели отстается нехватка информации об ОКИИ в открытом доступе. Это мешает создавать датасеты с достаточным количеством примеров для обучения. Одним из решений данной проблемы может быть использование методов аугментации для увеличения уже существующего датасета, но лучшим вариантом является взаимодействие с органом-регулятором в данной сфере – Федеральной службой по техническому и экспортному контролю, что позволит получить доступ к конфиденциальной информации.

Заключение

Использование нейросетевых моделей для решения задач классификации ОКИИ может значительно ускорить присвоение категорий значимости и снизить количество ошибок. В данном исследовании приведен один из способов реализации классификатора, определены архитектура и требования к функционалу. Также рассмотрены основные проблемы, с которыми может столкнуться разработчик.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Доктрина информационной безопасности Российской Федерации: утв. Указом Президента Рос. Федерации от 5 декабря 2016 г. № 646;

2. О внесении изменений в Федеральный закон "О безопасности критической информационной инфраструктуры Российской Федерации": федер. Закон Рос. Федерации от 7 апреля 2025 г. № 58-ФЗ: принят Гос. Думой Федер. Собр. Рос. Федерации 25 марта 2025 г.: одобр. Советом Федерации Федер. Собр. Рос. Федерации 2 апреля 2025 г. // КонсультантПлюс;

3. О безопасности критической информационной инфраструктуры: федер. Закон Рос. Федерации от 26 июня 2017 г. № 187-ФЗ: принят Гос. Думой Федер. Собр. Рос. Федерации 12 июля 2017 г.: одобр. Советом Федерации Федер. Собр. Рос. Федерации 19 июня 2017 г. // КонсультантПлюс;

4. Об утверждении Правил категорирования объектов критической информационной инфраструктуры Российской Федерации, а также перечня показателей критериев значимости объектов критической информационной инфраструктуры Российской Федерации и их значений: пост. Правительства Рос. Федерации от 8 фейраля 2018 г. № 127 с изм. от 24 декабря 2021 г.;

5. Базарбаев Х., Байрамов Т., Дадаева Л. Языки программирования для разработки искусственного интеллекта // Вестник науки. 2024. №10 (79);

6. Давыдов А. В., Жусупова А. К., Салыкова О. С. Сравнение различных языков программирования, применяемых в машинном обучении // Вестник науки. 2023. – №2 (59);

7. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. – 2-е изд. – М.: ДМК Пресс, 2015. – 401 с.;

8. Николенко С., Кадурин А., Архангельская Е Глубокое обучение. Погружение в мир нейронных сетей. – СПб.: Питер, 2018. – 401 с.;

9. Свэйгард Э. Автоматизация ручных процессов с помощью Python. Практическое руководство для начинающих. – М.: ООО "И.Д. Вильямс", 2017. – 592 с.;

10. Макаров А. В., Намиот Д. Е. Обзор методов очистки данных для машинного обучения // International Journal of Open Information Technologies. 2023. – №10.;

11. Макаров К. С., Халин А. А., Костенков Д. А., Муханов Э. Э. Сравнительный анализ библиотек для обработки естественного языка (NLP) // Auditorium. 2024. – №1 (41);

12. Полетаева Н. Г. Классификация систем машинного обучения // Вестник Балтийского федерального университета им. И. Канта. Серия: Физико-математические и технические науки. 2020. – №1.