

*Прокофьев Виктор Александрович*¹,
студент;
*Хохловский Владимир Николаевич*²,
доцент, канд. техн. наук, доцент

МЕТОДЫ И АЛГОРИТМЫ ПРЕДИКТИВНОЙ АНАЛИТИКИ КОМПОНЕНТОВ ЦИФРОВОЙ ПОДСТАНЦИИ

^{1,2} Россия, Санкт-Петербург,
Санкт-Петербургский политехнический университет Петра Великого,
¹ prokofiev2.va@edu.spbstu.ru, ² hohlovskij_vn@spbstu.ru

Аннотация. В работе исследуются возможности классификации данных, полученных при помощи цифровой модели трансформаторной линии подстанции, а также возможность предсказывать возникновение аварийных ситуаций при помощи нейросетевых моделей. В работе используются различные наборы данных, отличающиеся как представленными в них режимами работы трансформаторной линии, так и степенью приближенности моделируемых характеристик к реальным условиям. Далее рассматривается алгоритм приведения данных к доступному для обучения нейросети виду. В ходе опытов составляется и настраивается нейросетевой классификатор, позволяющий добиться 96 % точности в задаче многокатегориальной оценки рассматриваемых данных, а также предиктивную нейросетевую модель с точностью 97 % при заданном числе доступных предыдущих состояний.

Ключевые слова: энергетика, подстанции, цифровая модель, трансформаторы, нейросети, классификация, предиктивная аналитика, анализ данных, интеллектуальная аналитика.

*Viktor A. Prokofiev*¹,
Student;
*Vladimir N. Khokhlovskiy*²,
Associate Professor, Candidate of Technical Sciences

METHODS AND ALGORITHMS FOR PREDICTIVE ANALYTICS OF DIGITAL SUBSTATION COMPONENTS

^{1,2} Peter the Great St.Petersburg Polytechnic University, St. Petersburg, Russia,
¹ prokofiev2.va@edu.spbstu.ru, ² hohlovskij_vn@spbstu.ru

Abstract. This paper investigates the possibility of classifying the data obtained using a digital model of a substation transformer line, as well as the possibility of predicting the occurrence of emergency situations using neural network models. Different data sets are used in this work, differing both in the transformer line operating modes represented in them and in the degree of closeness of the modeled characteristics to real conditions. Next, we consider

the algorithm of data reduction to the form available for neural network training. In the course of experiments, a neural network classifier is compiled and tuned to achieve 96 % accuracy in the task of multi-category evaluation of the considered data, as well as a predictive neural network model with 97 % accuracy at a given number of available previous states.

Keywords: energy, substations, digital model, transformers, neural networks, classification, predictive analytics, data analysis, intelligent analytics.

Введение

На сегодняшний день оптимизация и модернизация энергетической отрасли страны являются особенно актуальными вопросами. В связи с этим особую значимость приобретают новые методы предотвращения аварийных ситуаций на объектах инфраструктуры энергетической отрасли. Данное исследование ставит целью повысить безопасность и износостойкость электрических подстанций, увеличить надежность контроля и скорость реагирования систем защиты от короткого замыкания на трансформаторах при помощи введения современных нейросетевых технологий. Подобные технологии способны работать совместно с традиционными релейными системами защиты, повышая точность работы, а также обеспечивая возможность предварительного реагирования на возникающие предаварийные ситуации.

1. Постановка задачи

Целью работы является исследование возможности классификации данных, полученных при помощи цифровой модели трансформаторной линии подстанции, а также предсказания возникновения аварийных ситуаций при работе оборудования. В работе будут рассмотрены возможности двухкритериальной и многокритериальной классификации и предсказания, а также опытным путем проанализировано влияние погрешностей измерительных приборов на эффективность работы нейросетевого классификатора и определена наибольшая возможная точность нейронной сети при помощи настройки ее архитектуры и параметров.

1.1. Описание предметной области

Линия-трансформатор с выключателем (схема ЗН) – это распространенная схема соединения на электрических подстанциях, используемая на напряжениях 35–220 кВ. Она применяется для тупиковых или ответвительных двухтрансформаторных подстанций [2].

Она включает в себя следующие основные элементы:

- 1) линия – подводит электроэнергию от вышестоящей сети к подстанции;
- 2) выключатель – коммутационный аппарат, предназначенный для включения и отключения линии или трансформатора;
- 3) трансформатор – повышает или понижает напряжение электроэнергии.

В связи с тем, что для продолжения работы необходимо было получить большие объемы данных, воспользуемся цифровой моделью подстанции.

В модели реализован блок линия-трансформатор с выключателем (см. рис. 1). К обмотке НН трансформатора подключена трехфазная симметричная нагрузка. Напряжение высокой стороны – 35 кВ, напряжение низкой стороны – 0,4 кВ. Мощность трансформатора – 2500 кВА.

1.2. Определение проблемы

Короткое замыкание в трансформаторе электрической подстанции – это аварийный режим работы, при котором происходит непреднамеренное соединение между двумя проводниками, находящимися под разным напряжением. Это приводит к резкому скачку тока, который может в десятки, а то и в сотни раз превышать номинальный ток трансформатора. Это серьезная авария, которая может привести к значительным материальным ущербам и перебоям в электроснабжении.

В данной работе рассматриваются следующие группы коротких замыканий, определенные по месту возникновения:

1) на стороне высокого напряжения (далее ВН) вне зоны действия дифференциальной защиты (точка K1);

2) на стороне ВН в зоне действия дифференциальной защиты (точка K2);

3) на стороне низкого напряжения (далее НН) в зоне действия дифференциальной защиты (точка K3);

4) на стороне НН вне зоны действия дифференциальной защиты (точка K4).

Токи и напряжения по сторонам ВН и НН измеряются соответственно в блоках ТА-35 ТН-35 и ТА-0,4 ТН-0,4.

2. Сбор данных с модели

2.1. Влияние помех на качество данных

При обучении нейросетевой модели были использованы несколько наборов данных, отличающихся местом возникновения короткого замыкания (далее КЗ) – вне зоны дифференциальной защиты, в зоне дифференци-

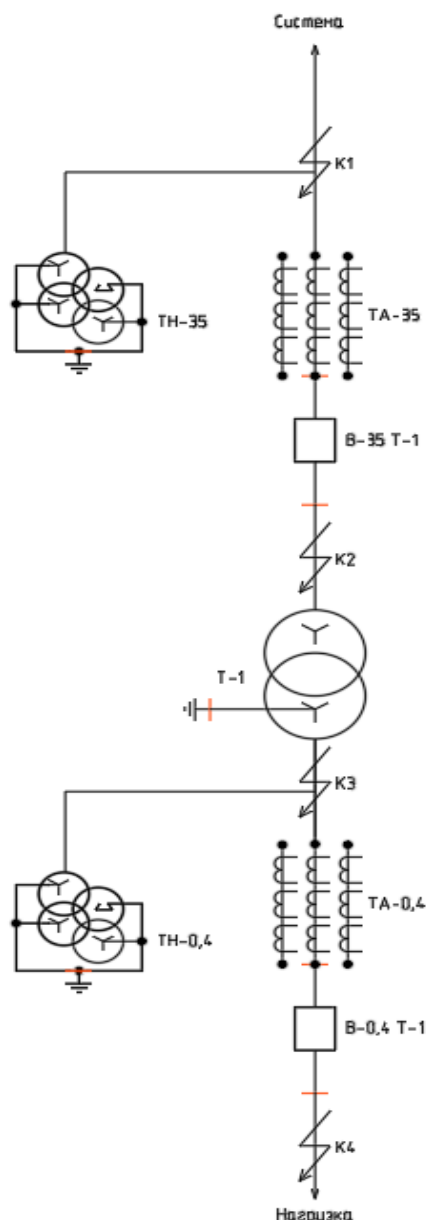


Рис. 1. Однолинейная схема моделируемой подстанции

альной защиты, на стороне высокого напряжения, на стороне низкого напряжения; характером КЗ – замыкание на землю, замыкание между фазами.

Также использовались данные с различной степенью приближенности к реальным условиям, где наиболее «простые» данные отображали только нормальный режим и резкое изменение токов и напряжений при возникновении КЗ (см. рис. 2), а данные, более приближенные к реальным, также включали в себя симуляцию включения на холостой ход, насыщения трансформаторов тока и наложение небольших случайных шумов на показания измерительных приборов (см. рис. 3).



Рис. 2. Графики токов на стороне высокого напряжения упрощенной модели

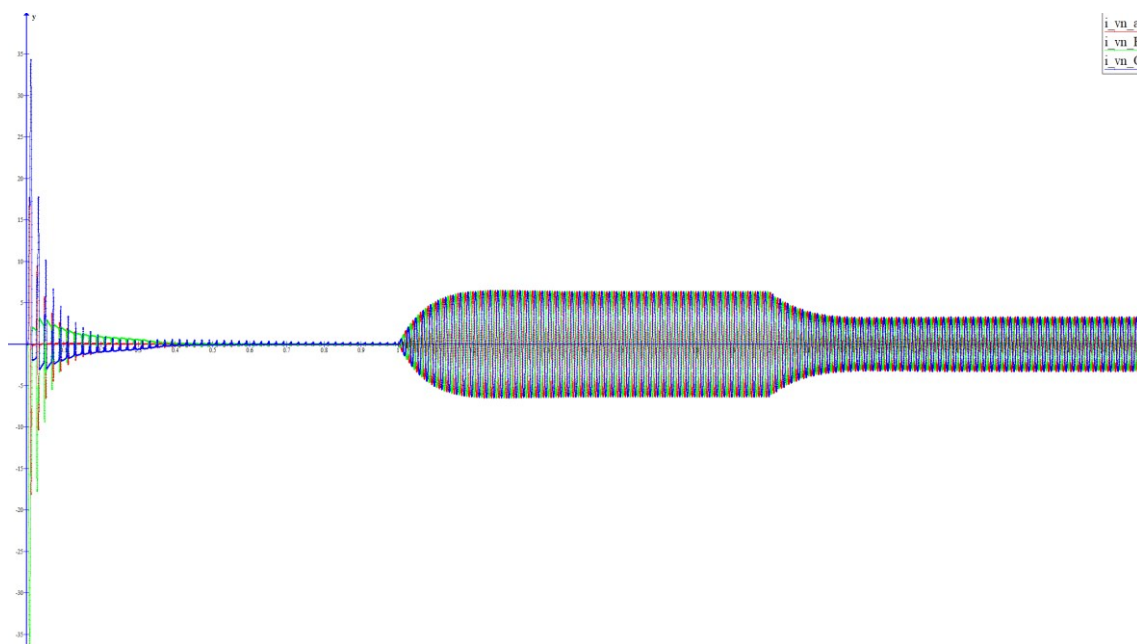


Рис. 3. Графики токов на стороне высокого напряжения модели, приближенной к реальной

Данные с упрощенной модели использовались для первичного обучения модели, в ходе которого опытным путем была подтверждена возможность нейронных сетей с удовлетворительной точностью выявлять нормальный режим работы и режим короткого замыкания, а также различать КЗ по месту возникновения, характеру возникновения и фазам, на которых произошло КЗ.

Данные с модели, приближенной к условиям работы реального оборудования, использовались при дальнейшем обучении нейросетевого классификатора и предсказателя. При помощи этих данных исследовалось влияние помех измерительных приборов на точность работы программ, а также способность нейросетевой модели корректно предсказывать переходные процессы в трансформаторах.

2.2. Обработка данных

Данные, полученные с модели трансформаторной линии подстанции, представим в виде таблице, содержащей значения высоких и низких напряжений и токов по трем фазам (см. табл. 1). Также при обучении нейросети-классификатора добавим столбец, в котором обозначим дискретными значениями текущее состояние системы (нормальный режим – 0, КЗ на фазе А – 1 и т. д.).

Таблица 1

Пример записи данных

i_vn_a	i_vn_B	i_vn_C	v_vn_A	v_vn_B	v_vn_C	tag
-1,753	49,307	-47,563	1483,292	23972,81	-25456,108	0
0,449	48,224	-48,681	2602,537	23343,70	-25946,243	0

Это позволит далее использовать алгоритмы обучения с учителем. Использование только численных значений хоть и затрудняет чтение данных человеком, упрощает дальнейшую их обработку для дальнейшей классификации, так как большая часть алгоритмов машинного обучения работают с численными данными [3].

Сначала нормализуем численные данные. Будем использовать Min-Max метод нормализации [5].

`MinMaxScaler.fit()` – вычисляет среднее и стандартное отклонение (или другие параметры масштабирования), необходимые для выполнения операции масштабирования.

`MinMaxScaler.transform()` – преобразует набор данных после нормализации. Преобразование обычно масштабирует признаки до среднего значения 0 и стандартного отклонения 1 (стандартизация) или масштабирует их до определенного диапазона (например, [0, 1] при масштабировании Min-Max) в зависимости от используемого принципа масштабирования.

Следующим шагом является деление набора данных на обучающие и тестовые множества для проведения обучения с учителем [4]. При этом

необходимо, чтобы в обоих множествах присутствовали точки, характеризующие разные состояния системы. Для контроля обобщающей способности сети, на основе которой строится классификатор, полезно использовать тестовое множество, формируемое из случайно отбираемых примеров обучающего набора данных. Примеры тестового множества не участвуют в процессе обучения сети (т. е. не влияют на подстройку её весов), а просто подаются на её вход вместе с обучающими примерами.

Для этого воспользуемся функцией `train_test_split` библиотеки `scikit-learn` [9]. Она позволяет разделить общее множество значений входных и выходных параметров на обучающее и тестовое подмножества в заданных пропорциях.

3. Составление нейросетевых моделей

3.1. Нейросеть-классификатор

Ключом к созданию эффективного классификатора является выбор оптимального размера сети, то есть количества соединений между нейронами. Эти соединения настраиваются во время обучения и обрабатывают входные данные при работе сети.

Слишком маленькая сеть не сможет реализовать сложные функции разделения классов, что приведет к неточному результату.

С другой стороны, увеличение числа связей повышает информационную емкость модели, что теоретически может улучшить скорость обучения и работы. Однако, это также может привести к переобучению, когда модель становится не способна эффективно работать с новыми данными.

Инициализацию модели проведем при помощи функции `Sequential` библиотеки `Keras` [10], которая позволяет последовательно создавать слои нейронной сети, добавляя каждый новый слой поверх предыдущего.

После этого добавляем слои в сеть в соответствии с числом категорий, на которые необходимо обучиться разделять входные данные (см. рис. 4). Выполняется это при помощи функции `add`, в параметрах которой задается тип слоя, функцию активации нейронов слоя и форму входных данных слоя. При этом крайний, т. е. выходной слой должен иметь число нейронов, равное числу возможных категорий данных.

```
basic_model.add(Dense(units = 500 , activation = 'relu', input_shape = (12,)))
basic_model.add(Dense(200, activation = 'relu'))
basic_model.add(Dense(3))
```

Рис. 4. Слои нейросети-классификатора

После этого необходимо определить тип оптимизации весов. Будем использовать оптимизатор Адама [6] со скоростью обучения 0,001.

После этого необходимо скомпилировать модель и начать обучение при помощи команды `fit`, позволяющей задать обучающие входные и выходные множества, а также число циклов обучения. Для оценки точности

модели используем метод evaluate [7]. Он вычисляет потери (часто та же функция потерь, которая использовалась во время обучения) и любые заданные метрики (например, точность) для предсказаний модели на тестовых данных. Если сеть показывает высокую точность как на обучающем, так и на тестовом множествах, то можно говорить о том, что сеть приобрела обобщающую способность.

Одним из способов улучшения обобщающей способности сети является увеличение числа обучающих примеров или сокращение числа связей [1]. Первое не всегда возможно из-за ограниченного объема набора данных и возрастания вычислительных затрат. Сокращение же числа связей приводит к ухудшению точности сети. Поэтому выбор размера модели часто оказывается достаточно сложной задачей, требующей многократных экспериментов.

В ходе серии экспериментов и настройки нейросетевой модели получится добиться точности определения 3-х категорий состояния оборудования (нормальный режим, КЗ на землю, КЗ на другую фазу) в 96 %, после чего результаты работы были представлены в виде графика при помощи метода ConfusionMatrixDisplay (см. рис. 5).

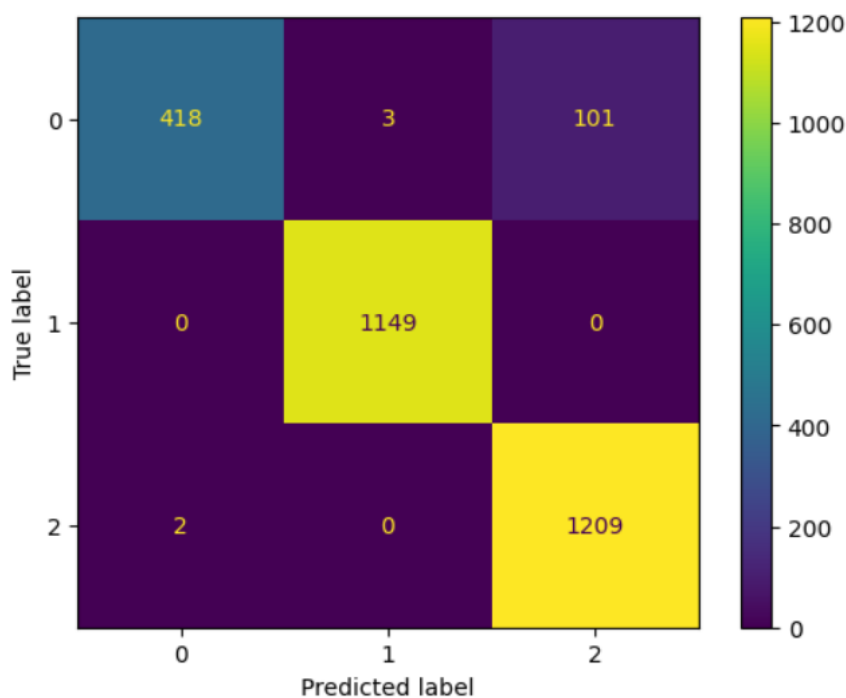


Рис. 5. Результат работы классификатора

3.2. Нейросеть-предсказатель

Для построения предиктивной нейросети используем архитектуру LSTM. В основе LSTM архитектуры лежит ячейка памяти, оснащенная тремя воротами: забывающим, входным и выходным. Эти ворота контролируют поток информации через ячейку, позволяя сети запоминать важные сведения и отфильтровывать несущественные.

При формировании как входных данных, так и самой LSTM-нейросети важно учитывать количество предшествующих состояний системы n , используемых для предсказания состояния системы $n+1$. Исходя из этого формируется трехмерный массив входных данных.

Архитектура сети включает в себя LSTM-слой, предназначенный для обработки входных последовательностей с формой, определяемой количеством признаков (столбцов) в обучающих данных. После установлен слой Dropout для предотвращения переоценки (см. рис. 6).

```
basic_model = Sequential()
basic_model.add(LSTM(200, return_sequences=True, input_shape=(30, 12)))
basic_model.add(Dropout(0.2))
basic_model.add(Dense(1))
```

Рис. 6. Слои предиктивной нейросети

После проведения нескольких экспериментов были установлены оптимальные параметры нейросети: число элементов входного LSTM слоя, коэффициент Dropout слоя, число предшествующих состояний. Нейросеть была протестирована на приближенных к реальным данным, полученных с цифровой модели подстанции (см. рис. 7).

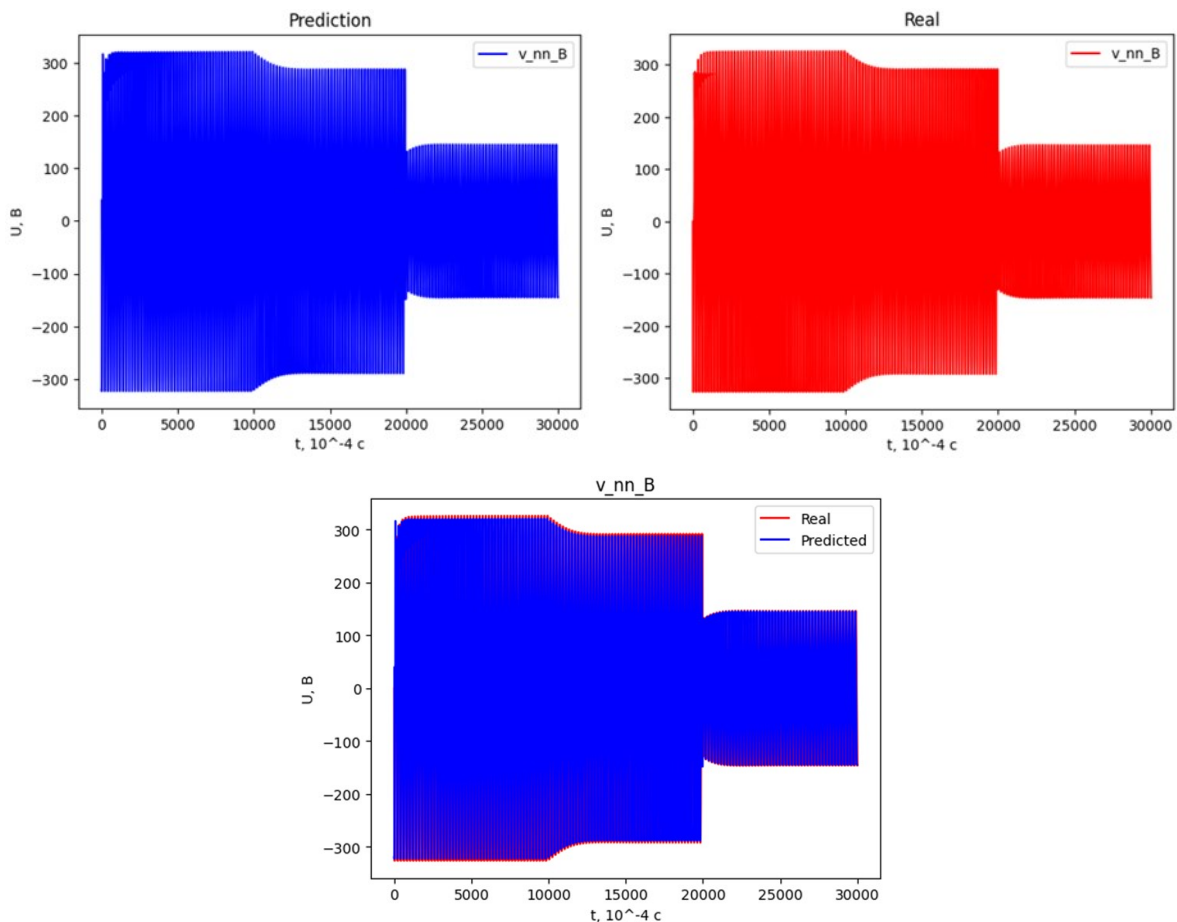


Рис. 7. Результат работы предиктивной нейросети

Заключение

В работе приведены сведения об устройстве трансформаторной блок-линии подстанции, проанализированы ее возможные режимы работы, а также способность особенностей и погрешностей измерительных приборов влиять на поступающие данные о состоянии трансформаторной линии.

К полученным при помощи цифровой подстанции данным были применены методы первичной обработки с целью приведения их к форме, доступной для обработки нейросетевыми моделями. В ходе эксперимента была проведена компоновка многослойной нейронной сети на языке python с применением библиотеки Keras, настройка числа связей нейронов и активационных функций нейронов с целью достижения наиболее высокой точности распознавания категорий данных нейросетью. В результате была достигнута точность в 96 % при многокатегориальной классификации. Аналогичная компоновка и настройка были проведены также для предиктивной нейросети на основе LSTM слоя. Данная модель показала точность в 97 % при предсказании изменений на основе 30 точек, характеризующих предыдущие состояния системы (3 мс в реальном времени).

Список литературы

1. Галушкин А. И. Теория нейронных сетей. Серия «Нейрокомпьютеры и их применение». – М.: ИПРЖР, 2000. – Кн. 1. – 456 с.
2. Кокин С. Е., Дмитриев С. А., Хальясмаа А. И. Схемы электрических соединений подстанций. – Екатеринбург: Изд-во Урал. ун-та, 2015. – 100 с. – ISBN 978-5-7996-1457-7.
3. Обучение нейросети: методы и алгоритмы [Электронный ресурс] // Training data: <https://trainingdata.ru> [сайт]. – 2024. – URL: <https://trainingdata.ru/metodic/tpost/vjrhfibe51-obuchenie-neiroseti-metodi-i-algoritmi> (дата обращения: 10.04.2024).
4. Созыкин А. В. Обзор методов обучения глубоких нейронных сетей // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. – 2017. – № 3.
5. Умная нормализация данных [Электронный ресурс] // Хабр. – 2024. – URL: <https://habr.com/ru/articles/527334/> (дата обращения: 10.04.2024).
6. Kingma D. P., Ba J. Adam: a method for stochastic optimization // ICLR 2015. – URL: <https://arxiv.org/abs/1412.6980> дата обращения: 10.04.2024).
7. Model training APIs [Electronic resource] // Keras: <https://keras.io> [website]. – 2024. – URL: https://keras.io/api/models/model_training_apis/ (access date: 18.04.2024).
8. sklearn.preprocessing.MinMaxScaler [Electronic resource] // Scikit Learn: <https://scikit-learn.org/> [website]. – 2024. – URL: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html> (access date: 15.04.2024).
9. sklearn.model_selection.train_test_split [Electronic resource] // Scikit Learn: <https://scikit-learn.org/> [website]. – 2024. – URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (access date: 15.04.2024).
10. The Sequential class [Electronic resource] // Keras: <https://keras.io/> [website]. – 2024. – URL: <https://keras.io/api/models/sequential/> (access date: 17.04.2024).