ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОБНАРУЖЕНИЯ ВЕБ-АТАК

Лапина М. А.[1], Мовзалевская В. В.[2], Токмакова М. Е.[3], Бабенко М. Г.[4], Саджид М.[5]

DOI: 10.21681/2311-3456-2024-4-92-103

Цель исследования: *исследование применимости методов машинного обучения и их оценки в области обнаружения вторжений и атак в веб-среде.*

Методы исследования: *рассмотрены различные реализации алгоритмов машинного обучения для определения типа и атаки в веб-среде, в частности алгоритмы классификации и кластеризации. Для обнаружения атак были выбраны самые оптимальные алгоритмы машинного обучения, реализованные с помощью библиотеки Scikit-learn, после их рассмотрения и сравнительного анализа. В рамках этой работы параметрами оценки эффективности исследуемых алгоритмов являются показатели времени обучения, а также характеристики из Confusion matrix и Classification Report для алгоритмов классификации, и Homogeneity, Completeness, V-measure для алгоритмов кластеризации.*

Результат: *для рассматриваемой выборки данных был определен и реализован наиболее экономичный по времени и качеству алгоритм – метод деревьев решений. Наилучшие характеристики для решения поставленной задачи показали деревья решения точность при определении типа и подтипа атаки составляет 99.9662% и 99.9576% соответственно. Время обнаружение атаки в среднем равно 85.39 ms и 114.72 ms для типа и подтипа соответственно.*

Практическая ценность *состоит в том, что предлагается решение задачи для обнаружения и определения различных типов и под типов атаки в веб среде которые позволяют разработать оптимальную стратегию защиты интернет ресурсов и минимизировать вероятность потери, кражи или искажения данных.*

Вклад авторов: *Лапина М. А., Бабенко М. Г., Саджид М. – выбор и постановка задачи исследования; Лапина М. А., Мовзалевская В. В., Токмакова М. Е. – выбор решений, программная реализация и проведение экспериментов; Лапина М. А., Мовзалевская В. В., Токмакова М. Е., Бабенко М. Г. – обсуждения результатов экспериментов, анализ полученных результатов.*

Ключевые слова: *веб-среда, алгоритмы классификации, алгоритмы кластеризации, искусственный интеллект, интернет-безопасность, методы обнаружения угроз, информационная безопасность.*

DETECTING WEB ATTACKS USING MACHINE LEARNING ALGORITHMS

Lapina M. A.[6], Movzalevskaya V. V.[7], Tokmakova M. E.[8], Babenko M. G.[9], Sajid M.[10]

The purpose of the study: *study the applicability of machine learning methods and their evaluation in the field of intrusion and attack detection in the web environment.*

Research methods: *various implementations of machine learning algorithms for determining the type and attack in the web environment are considered classification and clustering algorithms. To detect attacks, the most optimal machine learning algorithms implemented using the Scikit-learn library were selected after their consideration and comparative*

1 Лапина Мария Анатольевна, кандидат физико-математических наук, доцент, доцент кафедры информационной безопасности автоматизированных систем Северо-Кавказского федерального университета, Ставрополь, Россия. E-mail: mlapina@ncfu.ru, ORCID: 0000-0001-8117-9142.

2 Мовзалевская Виталия Валентиновна, студентка специальности информационная безопасность автоматизированных систем Северо-Кавказского федерального университета, Ставрополь, Россия. E-mail: vitaliya1306@gmail.com, ORCID: 0009-0007-7540-3110.

3 Токмакова Марина Евгеньевна, студентка специальности информационная безопасность автоматизированных систем Северо-Кавказского федерального университета, Ставрополь, Россия. E-mail: marinatokmakova175@mail.ru, ORCID: 0009-0000-2608-7712.

4 Бабенко Михаил Григорьевич, доктор физико-математических наук, доцент, заведующий кафедрой вычислительной математики и кибернетики Северо-Кавказского федерального университета, Ставрополь, Россия. E-mail: mgbabenko@ncfu, ORCID: 0000-0001-7066-0061.

5 Саджид Мохаммад, доктор наук, доцент кафедры компьютерных наук Мусульманского университета, Алигарха, Алигарх, Индия. E-mail: sajid.cst@gmail.com, ORCID: 0000-0001-8822-5332.

6 Maria A. Lapina, Ph.D., Associate Professor, Associate Professor of the Department of Information Security of Automated Systems, North Caucasus Federal University, Stavropol, Russia. E-mail: mlapina@ncfu.ru, ORCID: 0000-0001-8117-9142.

7 Vitaliya V. Movzalevskaya, student of the specialty information security of automated systems at the North Caucasus Federal University, Stavropol, Russia. E-mail: vitaliya1306@gmail.com, ORCID: 0009-0007-7540-3110.

8 Marina E. Tokmakova, student of the specialty information security of automated systems at the North Caucasus Federal University, Stavropol, Russia. E-mail: marinatokmakova175@mail.ru, ORCID: 0009-0000-2608-7712.

9 Mikhail G. Babenko, Dr.Sc., Associate Professor, Head of the Department of Computational Mathematics and Cybernetics, North Caucasus Federal University, Stavropol, Russia. E-mail: mgbabenko@ncfu, ORCID: 0000-0001-7066-0061.

10 Mohammad Sajid, Ph.D., Associate Professor, Department of Computer Science, Muslim University, Aligarh, Aligarh, India. E mail: sajid.cst@gmail.com, ORCID: 0000-0001-8822-5332.

analysis. In this work, the parameters for evaluating the effectiveness of the studied algorithms are training time indicators, as well as characteristics from the Confusion matrix and Classification Report for classification algorithms, and Homogeneity, Completeness, V-measure for clustering algorithms.

The results obtained: *for the considered data sample, the most time-efficient and quality-efficient algorithm was determined and implemented - the decision tree method. The best characteristics for solving the problem were shown by decision trees; the accuracy in determining the type and subtype of an attack is 99.9662% and 99.9576%, respectively. The average attack detection time is 85.39 ms and 114.72 ms for the type and subtype, respectively.*

The scientific novelty *is that it offers a solution to the problem of detecting and defining various types and subtypes of attacks in the web environment, which allows developing an optimal strategy for protecting Internet resources and minimizing the likelihood of loss, theft or corruption of data.*

Contribution of the authors: *Lapina M. A., Babenko M. G., Sajid M. – selection and formulation of the research problem; Lapina M. A., Movzalevskaya V. V., Tokmakova M. E. – selection of solutions, software implementation and experiments; Lapina M. A., Movzalevskaya V. V., Tokmakova M. E., Babenko M. G. – discussions of the experimental results, analysis of the obtained results.*

Keywords: *web environment, classification algorithms, clustering algorithms, artificial intelligence, Internet security, threat detection methods, information security.*

Введение

Внедрение умных устройств в жизнь людей предоставило злоумышленникам значительно большее количество ресурсов с низким уровнем защиты, что позволило им разработать новые сценарии для проведения кибератак с использованием ботнет. Ботнет состоят из тысяч зараженных вредоносным программным обеспечения умных устройств, которые одновременно непрерывно отправляют огромные объемы данных для нанесения огромного вреда отдельным пользователям, компаниям с использованием DDoS атак [1].

Безопасность веб-сервисов является сложной задачей. В целом DDoS-атаки стали серьезной угрозой для веб-сервисов. Для выполнения DoS/DDoS-атак могут использоваться различные подходы, включая сетевые подходы, такие как лавинная рассылка через пакеты TCP SYN, ICMP или UDP, а также подходы на основе хостов, когда один или несколько хостов нацелены на определенные приложения для использования структуры своей памяти, протокола аутентификации или определенного алгоритма [2]. По данным Information Technology Intelligence Consulting, час простоя ИТ-услуг может стоить компаниям от 300 000 до 1 000 000 долларов. Учитывая эту цифру, размер понесенного финансового ущерба невообразим, когда в октябре 2020 года на тысячи IP-адресов Google была обрушена DDoS-атака. Атака была совершена тремя китайскими интернет-провайдерами и длилась шесть месяцев, достигнув ошеломляющего уровня 2,5. Тбит/с [1].

Алгоритмы машинного обучения позволяют обнаруживать и предотвращать атаки, что значительно повышает эффективность защиты сайта. По своей сути, машинное обучение можно представить как процесс вывода алгоритмов прогнозирования неизвестных данных, с использованием ранее собранной информации.

1. Постановка задачи

Задачей данного исследования является создание прогностической модели, способной различать «плохие» сетевые соединения (вторжения или атаки) и «хорошие» (обычные) соединения, а также определять конкретный тип атак для защиты компьютерной сети от неавторизованных пользователей, включая, возможно, инсайдеров. Используемая в исследовании база данных содержит стандартный набор данных для аудита, который включает в себя широкий спектр вторжений и атак, имитируемых в сетевой среде. Все, используемые в работе, алгоритмы машинного обучения были реализованы с использованием библиотеки Scikit-learn, также для метода градиентного бустинга приведена реализация CatBoost. Для моделирования использовался датасет KDD Cup 1999.

2. Обзор литературы

В табл. 1, 2 приведены свойства всех реализованных в исследовании алгоритмов машинного обучения:

- ❖ возможность работы с категориальными данными;
- ❖ сложность модели алгоритма, связанная с количеством параметров в модели;
- ❖ интерпретируемость, чем она выше у модели, тем легче понять, почему были приняты определенные решения или прогнозы;
- ❖ необходимость масштабирования данных;
- ❖ временная сложность.

Учитывая данные свойства методов, самыми оптимальными являются логическая регрессия (возможность работы с категориальными данными, низкая сложность модели, высокий уровень интерпретируемости, не требуется масштабируемость данных) и деревья решений (возможность работы с категориальными данными, низкая сложность модели,

Таблица 1.

Свойства алгоритмов машинного обучения

Методы	Высокая			
	Категориальные данные	Сложность модели	Уровень интерпретируемость	Масштабируемость данных
Алгоритмы классификации				
Случайный лес [3]	–	Высокая	Средний	Не требуется
GB (Scikit-learn) [4]	+	Высокая	Высокий	Не требуется
GB (CatBoost) [5]	+	Высокая	Средний	Не требуется
Logit model [6]	+	Низкая	Высокий	Не требуется
Наивный Байес [7]	+	Высокая	Низкий	Требуется
Деревья решений [8]	+	Низкая	Высокий	Не требуется
SVMs [9]	+	Высокая	Низкий	Не требуется
Метод k-NN [10]	+	Низкая	Высокий	Требуется
Алгоритмы кластеризации				
Метод k-средних [11]	–	Высокая	Низкий	Требуется
HCA [12]	+	Низкая	Средний	Требуется

Таблица 2.

Свойства алгоритмов машинного обучения

Методы	Временная сложность
Алгоритмы классификации	
Случайный лес	O(n)
GB	O(knm)
Logit model	O(n)
Наивный Байес	O(dk) + O(dkn)
Деревья решений	$O(n_{features} \times n^2_{samples} \log(n_{samples}))$
SVMs	$O(n_{features} \times n^2_{samples})$
Метод k-NN	O(log n)
Алгоритмы кластеризации	
Метод k-средних	O(knT)
HCA	O(n3)

высокий уровень интерпретируемости, не требуется масштабируемость данных).

3. Моделирование

В данном разделе приведены реализация, обучение и тестирование вышеописанных алгоритмов машинного обучения. Моделирование проводилось на Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz, DDR3 SDRAM 256 Gb под управлением операционной системы Ubuntu 18.04.5 на языке программирования Python 3.8.8 с использованием jupyter core 4.7.1, jupyter-notebook 6.3.0, qtconsole 5.0.3, ipython 7.22.0, ipykernel 5.3.4, jupyter client 6.1.12, jupyter lab 3.0.14, nbconvert 6.0.7, ipywidgets 7.6.3, nbformat 5.1.3, traitlets 5.0.5, sklearn 0.24.1, catboost 1.2.3 на языке программирования Python.

В качестве набора данных используются данные из необработанного трафика, перехваченного утилитой tcpdump в локальной сети. Набор данных содержит 494 023 записи, из них 330 995 тренировочных и 163 028 тестовых. Набор данных взят KDD Cup 1999. Задача обученной модели – определить атаку по ее свойствам.

В машинном обучении существует много различных метрик, которые позволяют определить точность и эффективность работы обученной модели. В рамках данного исследования параметрами оценки эффективности исследуемых алгоритмов являются показатели времени обучения, а также характеристики из Confusion matrix и Classification Report для алгоритмов классификации, и Homogeneity, Completeness, V-measure для алгоритмов кластеризации.

3.1. Алгоритмы классификации

Для оценки точности работы алгоритмов классификации используются показатели времени работы модели, Confusion matrix и Classification Report, представленные в таблицах в соответствующих подразделах, а также в разделе 5.

В работе для обнаружения атак в веб-среде выбранные алгоритмы машинного обучения реализованы с помощью библиотеки Scikit-learn. Однако, для алгоритма градиентного бустинга приведена реализация CatBoost, т. к. она обеспечивает высокую производительность и предотвращение переобучения, также данная реализация заявлена как самая быстрая и оптимизированная [15], поэтому в исследовании приведено ее сравнение с Scikit-learn, самой часто реализуемой.

Для моделей классификации Случайный лес, GB (Scikit-learn), GB (CatBoost), Logit model, Наивный Байес, Деревья решений, SVMs, Метод k-NN, представлены в табл. 3. Как видно из приведенных данных в табл. 3.A (Precision), табл. 3.B (Recall) и табл. 3.C (F1-Score) алгоритм определяет все классы, причем с минимальным, по отношению к общему числу атак текущего класса, количеством ошибок. Самый маленький класс, состоящий из 10 атак, верно определился только в половине случаев. Из данных представленных в Таблицах 3.A, 3.B и 3.C наилучшие параметры precision, recall и F1-Score для классов benign, dos, probe, r2l и u2r близки к своим лучшем значениям для метода случайный лес. Точность предсказания данного практически достигает наилучших результатов. Достигается за счет использование ансамбля деревьев решений [13, 14], которые позволяют выявить ключевые факторы и более точно решить задачу классификации по сравнению с имеющимися аналогами. Однако, стоит отметить, что вычислительная сложность случайного леса больше вычислительная сложность деревья решений и требует больше вычислительных ресурсов для реализации приложений в реальном времени, чем деревья решений.

4. Алгоритмы кластеризации

Для оценки точности работы алгоритмов кластеризации используются показатели времени работы модели, Completeness, Homogeneity, V-measure, представленные в таблицах в соответствующих подразделах, в разделе 5, а также на рисунках, показывающих распределение классов в изначальных данных и по окончании работы алгоритмов.

4.1. Метод k-средних

Сначала приведены данные для определения типа атаки (5 классов), далее описано работа с определением подтипа атаки (23 класса).

Изначальное распределение классов в тренировочных данных, полученное с помощью Principal component analysis (PCA), представлено на рис. 1 (определение типа атаки).

Распределение кластеров, полученное в результате работы модели, обученной с помощью алгоритма K-средних, представлено на рис. 2 (определение типа атаки).

Ниже приведены основные метрики для оценки результатов работы модели при определении типа атаки:

- Completeness = 0.40571;
- Homogeneity = 0.8058;
- V-measure = 0.53969.

Не очень высокое значение Completeness показывает, что члены одного класса далеко не всегда относятся моделью к одному кластеру. Это наглядно представлено на рис. 3, слева фрагмент с изначальным распределением классов, а справа – фрагмент с результатом работы модели.

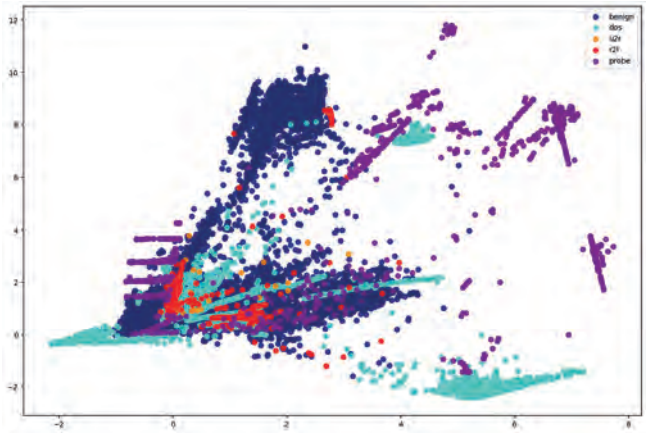

Рис. 1. Изначальное распределение кластеров в наборе данных при определении типа атаки

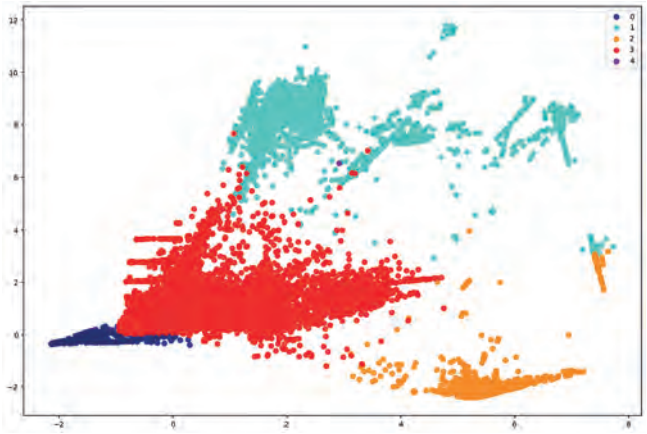

Рис. 2. Кластеры алгоритма K-means при определении типа атаки

Таблица 3.

Результат классификации атак на веб-сервисы с помощью моделей классификации Случайный лес, GB (Scikit-learn), GB (CatBoost), Logit model, Наивный Байес, Деревья решений, SVMs, Метод k-NN

A) Precision

	Метод								
	Случайный лес	GB (Scikit-learn)	GB (CatBoost)	Logit model	Наивный Байес	Деревья решений	SVMs	Метод k-NN	Support
Классы (тип атаки)									
benign	**1.00**	**1.00**	**1.00**	**1.00**	0.98	**1.00**	**1.00**	**1.00**	32041
dos	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	129287
probe	**1.00**	0.98	0.99	0.98	0.04	0.99	0.99	0.99	1320
r2l	**0.99**	0.75	0.96	0.93	0.23	0.98	0.98	0.96	369
u2r	**1.00**	0.00	0.50	0.75	0.00	0.38	**1.00**	0.50	10
Параметры									
macro avg	**1.00**	0.75	0.89	0.93	0.45	0.87	0.99	0.89	163027
weighted avg	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	**1.00**	**1.00**	163027

B) Recall

	Метод								
	Случайный лес	GB (Scikit-learn)	GB (CatBoost)	Logit model	Наивный Байес	Деревья решений	SVMs	Метод k-NN	Support
Классы (тип атаки)									
benign	**1.00**	0.99	**1.00**	**1.00**	0.91	**1.00**	**1.00**	**1.00**	32041
dos	**1.00**	**1.00**	**1.00**	**1.00**	0.73	**1.00**	**1.00**	**1.00**	129287
probe	**1.00**	0.98	0.99	0.97	0.97	**1.00**	0.99	0.99	1320
r2l	**0.97**	0.77	0.96	0.91	0.48	**0.97**	0.93	0.96	369
u2r	0.50	0.00	0.30	0.60	**0.80**	0.30	0.40	0.30	10
Параметры									
macro avg	**0.89**	0.75	0.85	**0.89**	0.78	0.85	0.86	0.85	163027
weighted avg	**1.00**	**1.00**	**1.00**	**1.00**	0.76	**1.00**	**1.00**	**1.00**	163027

C) F1-Score

	Метод								
	Случайный лес	GB (Scikit-learn)	GB (CatBoost)	Logit model	Наивный Байес	Деревья решений	SVMs	Метод k-NN	Support
Классы (тип атаки)									
benign	**1.00**	0.99	**1.00**	**1.00**	0.94	**1.00**	**1.00**	**1.00**	32041
dos	**1.00**	**1.00**	**1.00**	**1.00**	0.84	**1.00**	**1.00**	**1.00**	129287
probe	**1.00**	0.98	0.99	0.98	0.07	0.99	0.99	0.99	1320
r2l	**0.98**	0.76	0.96	0.92	0.31	**0.98**	0.96	0.96	369
u2r	**0.67**	0.00	0.37	**0.67**	0.01	0.33	0.57	0.37	10
Параметры									
accuracy	**1.00**	**1.00**	**1.00**	**1.00**	0.76	**1.00**	**1.00**	**1.00**	163027
macro avg	**0.93**	0.75	0.86	0.91	0.43	0.86	0.90	0.86	163027
weighted avg	**1.00**	**1.00**	**1.00**	**1.00**	0.85	**1.00**	**1.00**	**1.00**	163027

Рис. 3. Сравнение исходных данных и результатов работы модели

Показатель Homogeneity принимает довольно высокое значение, это свидетельствует о том, что каждый кластер преимущественно состоит только из членов одного класса.

Стоит отметить, что модель плохо определила классы с небольшим количеством членов, а также некоторые классы поделила между несколькими кластерами.

Оценка времени работы модели, полученная в ходе проведения экспериментов на одних и тех же данных, представлена в табл. 4 (раздел 5). Среднее время работы модели на тестовых данных составляет 0.30534 секунд.

Далее представлена работы с подтипами атак (23 класса). Изначальное распределение классов в тренировочных данных, полученное с помощью Principal component analysis (PCA), представлено на рис. 4 (определение подтипа атаки).

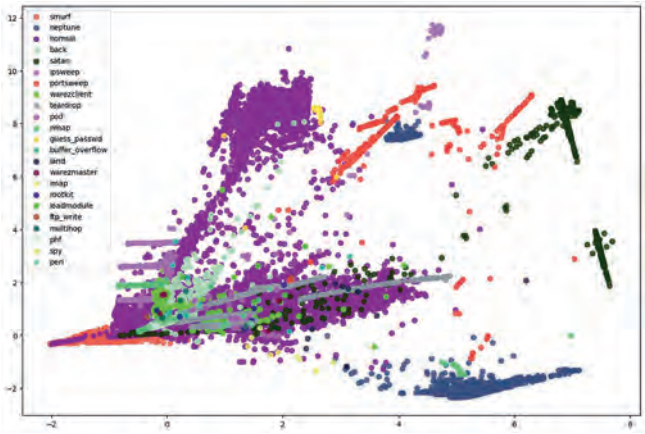

Рис. 4. Изначальное распределение кластеров в наборе данных при определении подтипа атаки

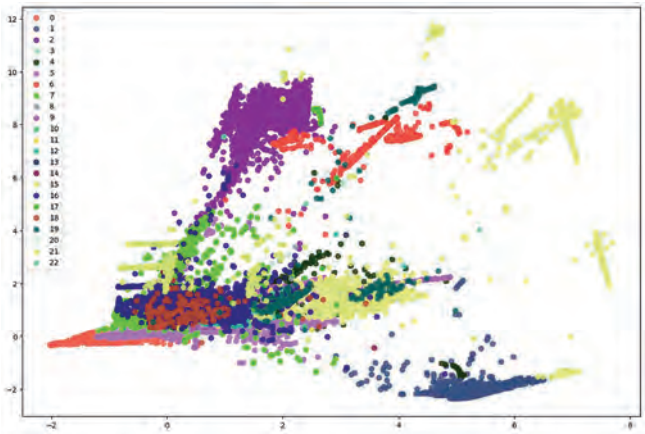

Рис. 5. Кластеры алгоритма K-means при определении подтипа атаки

Распределение кластеров, полученное в результате работы модели, обученной с помощью алгоритма K-средних, представлено на рис. 5 (определение подтипа атаки)

Ниже приведены основные метрики для оценки результатов работы модели при определении подтипа атаки:

- Completeness = 0.72072;
- Homogeneity = 0.94527;
- V-measure = 0.81786.

Достаточно высокое значение Completeness показывает, что члены одного класса практически всегда относятся моделью к одному кластеру.

Показатель Homogeneity принимает очень высокое значение, это свидетельствует о том, что каждый кластер преимущественно состоит только из членов одного класса.

Оценка времени работы модели, полученная в ходе проведения экспериментов на одних и тех же данных, представлена в табл. 5 (раздел 5). Среднее время работы модели на тестовых данных составляет 0.24645 секунд.

4.2. Иерархическая кластеризация

Во время тестирования алгоритма иерархической кластеризации было принято решение использовать только 5% от исходного набора тренировочных и тестовых данных, так как данный алгоритм требует большого количества памяти для корректной работы. Сначала приведены данные для определения типа атаки (5 классов), далее описано работа с определением подтипа атаки (23 класса).

Изначальное распределение классов в тренировочных данных, полученное с помощью Principal component analysis (PCA), представлено на рис. 6 (определение типа атаки).

Ниже приведены основные метрики для оценки результатов работы модели при определении типа атаки:

- Completeness = 0.40278;
- Homogeneity = 0.79463;
- V-measure = 0.53459.

Не высокое значение Completeness показывает, что члены одного класса далеко не всегда отнесены моделью к одному кластеру.

Показатель Homogeneity принимает довольно высокое значение, это свидетельствует о том, что каждый кластер преимущественно состоит только из членов одного класса.

Стоит отметить, что модель плохо определила классы с небольшим количеством членов, а также некоторые классы поделила между несколькими кластерами.

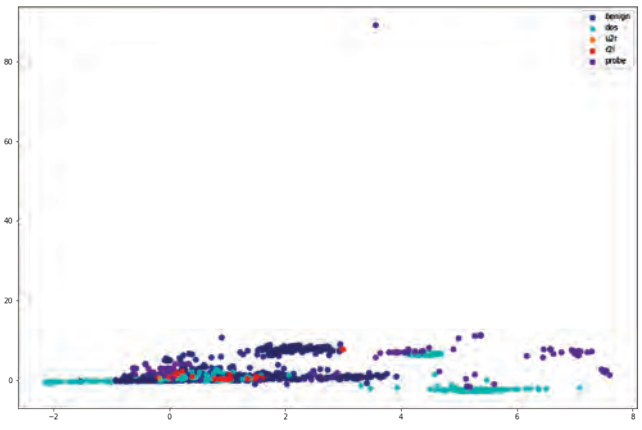

Рис. 6. Изначальное распределение кластеров в наборе данных при определении типа атаки

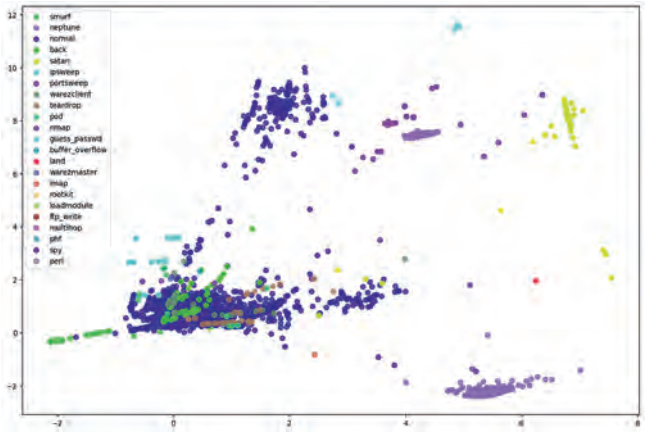

Рис. 8. Изначальное распределение кластеров в наборе данных при определении подтипа атаки

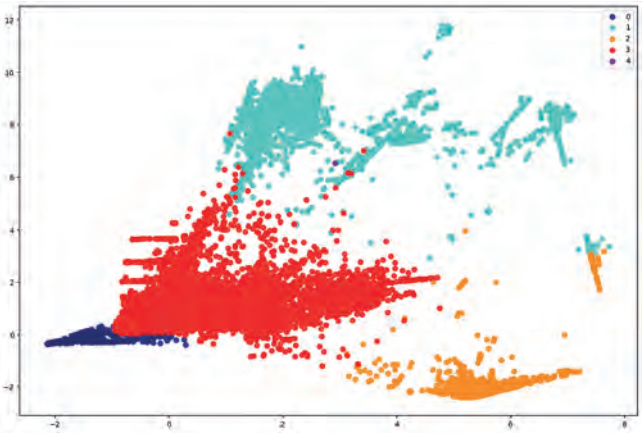

Рис. 7. Кластеры алгоритма Иерархической кластеризации при определении типа атаки

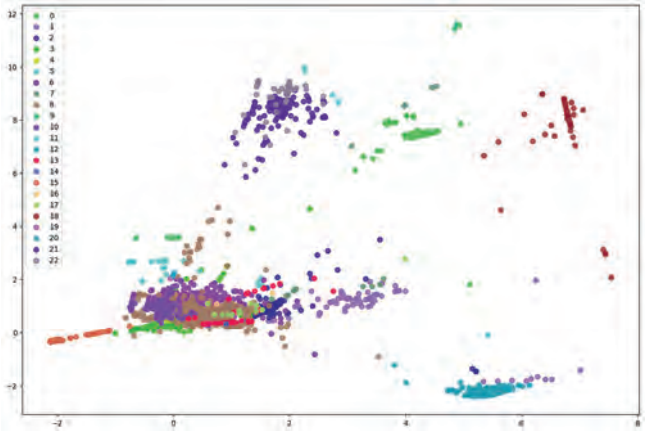

Рис. 9. Кластеры алгоритма иерархической кластеризации при определении подтипа атаки

Оценка времени работы модели, полученная в ходе проведения экспериментов на одних и тех же данных, представлена в табл. 4 (раздел 5). Среднее время работы модели на тестовых данных составляет 122.050 секунд.

Далее представлена работы с подтипами атак (23 класса). Изначальное распределение классов в тренировочных данных, полученное с помощью Principal component analysis (PCA), представлено на рис. 8 (определение подтипа атаки).

Распределение кластеров, полученное в результате работы модели, обученной с помощью алгоритма иерархической кластеризации, представлено на рис. 9. (определение подтипа атаки)

Ниже приведены основные метрики для оценки результатов работы модели при определении подтипа атаки:

- Completeness = 0.69550;
- Homogeneity = 0.95251;
- V-measure = 0.80397.

Не высокое значение Completeness показывает, что члены одного класса не всегда относятся моделью к одному кластеру.

Показатель Homogeneity принимает очень высокое значение, это свидетельствует о том, что каждый кластер преимущественно состоит только из членов одного класса.

Стоит отметить, что модель плохо определила классы с небольшим количеством членов, а также некоторые классы поделила между несколькими кластерами.

Оценка времени работы модели, полученная в ходе проведения экспериментов на одних и тех же данных, представлена в табл. 5 (раздел 5). Среднее время работы модели на тестовых данных составляет 102.074 секунд.

5. Анализ полученных данных

Проведён сравнительный анализ алгоритмов машинного обучения применительно к выборке атак и вторжений в веб-среде. Исследование продемонстрировало разную степень эффективности моделей

Таблица 4.

Время работы алгоритмов для определения типа атаки (5 классов)

Методы	Average, sec	T_{max}, sec	T_{min}, sec	σ
Алгоритмы классификации				
Случайный лес	2.82505	7.20053	2.01484	1.05204
Градиентный бустинг (Scikit-learn)	1.37482	4.23323	0.87714	0.72573
Градиентный бустинг (CatBoost)	0.88187	1.47155	0.74535	0.21805
Логическая регрессия	0.26269	0.62514	0.19481	0.08247
Наивный байесовский классификатор	0.85927	1.30261	0.80680	0.06827
Деревья решений	0.08539	0.11705	0.07151	0.01232
Метод опорных векторов	38.3194	38.3773	38.2732	0.02738
Метод k-ближайших соседей	670.8163	871.81656	650.8959	1.23631
Алгоритмы кластеризации				
Метод k-средних	0.30534	0.51908	0.23136	0.07874
Иерархическая кластеризация	122.050	283.725	96.0285	1.30705

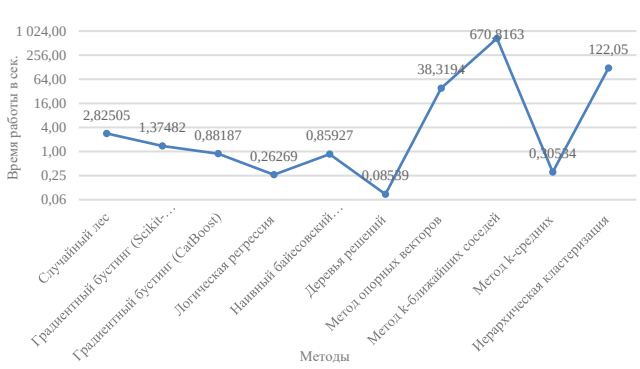

Рис.10. Время работы алгоритмов для определения типа атаки (5 классов)

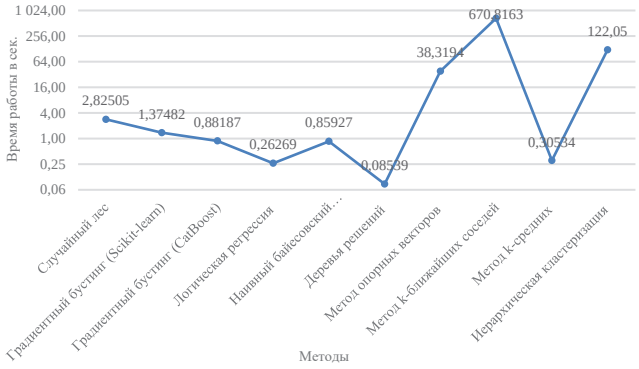

Рис.11. Время работы алгоритмов для определения подтипа атаки (23 классов)

Таблица 5.

Время работы алгоритмов для определения подтипа атаки (23 класса)

Методы	Average, sec	T_{max}, sec	T_{min}, sec	σ
Алгоритмы классификации				
Случайный лес	3.92719	4.50832	3.63689	0.2723
Градиентный бустинг (Scikit-learn)	3.24662	6.97630	2.70011	0.81519
Градиентный бустинг (CatBoost)	2.44120	5.76815	1.78189	0.87678
Логическая регрессия	0.32962	0.51410	0.27861	0.06833
Наивный байесовский классификатор	4.32658	4.89500	4.04916	0.22773
Деревья решений	0.11472	0.14774	0.10419	0.01176
Метод опорных векторов	48.88156	70.00742	43.25102	0.40573
Метод k-ближайших соседей	28.46818	31.26908	27.85873	0.35328
Алгоритмы кластеризации				
Метод k-средних	0.24645	0.34519	0.21609	0.03729
Иерархическая кластеризация	102.074	123.239	90.7761	0.53563

и скорости их работы в решении конкретной задачи определения типа и подтипа атак.

Ниже приведены показатели времени обучения каждой модели при определении типа атаки (Табл. 4, Рис. 10), а также подтипа атаки (Табл. 5, Рис. 11).

Исходя из показателей времени (Табл. 4) обучения моделей, их можно разделить на те, которые работают быстрее (случайны лес (2.82505), градиентный бустинг (Scikit-learn) (1.37482), градиентный бустинг (CatBoost) (0.88187), логическая регрессия (0.26269), наивный байесовский классификатор (0.85927), деревья решений (0.08539), метод k-средних (0.30534)) и те, которые работают медленнее (метод опорных векторов (38.3194), метод k-ближайших соседей (670.8163), иерархическая кластеризация (122.050)).

Исходя из показателей времени обучения моделей (Табл. 5), их можно разделить на те, которые работают быстрее (логическая регрессия (0.32962), деревья решений (0.11472), метод k-средних (0.24645) и те, которые работают медленнее (случайный лес (3.92719), градиентный бустинг (Scikit-learn) (3.24662), градиентный бустинг (CatBoost) (2.44120), наивный байесовский классификатор (4.32658), метод опорных векторов (48.88156), метод k-ближайших соседей (28.46818), иерархическая кластеризация (102.074)).

Также приведены показатели из матрицы несоответствия для алгоритмов классификации при определении типа атаки (Табл. 6). Из данных, представленных в табл. 6 можно сделать вывод, что алгоритм основанный на технологии случайный лес позволяет наилучшим образом определить тип атаки.

Данные точности определения подтипа атаки для каждого из алгоритмов представлены в табл. 7, опираясь на них можно сделать выводы, что случайный лес позволяет получить наибольшую точность определения подтипа атаки, но при этом не позволяет определить следующие подтипы атаки: land, spy, loadmodule, phf. Градиентный бустинг Scikit-learn не позволяет определить следующие подтипы атаки: warezmaster, land, guess_passwd, imap, ftp_write multihop, loadmodule, perl, rootkit, phf. Градиентный бустинг CatBoost не позволяет определить следующие подтипы атаки: multihop, loadmodule, rootkit, satan, phf. Деревья решений не позволяют определить следующие подтипы атаки: land, imap, spy, multihop, rootkit, phf.

Данные точности определения подтипа атаки для каждого из алгоритмов представлены в табл. 8, опираясь на них можно сделать выводы, что логическая регрессия не позволяет определить следующие подтипы атаки: spy, multihop, loadmodule, perl, rootkit, phf. Наивный байесовский классификатор не позволяет определить следующие подтипы атаки: multihop, loadmodule, rootkit, phf. Метод опорных векторов не позволяет определить следующие подтипы атаки: land, spy, multihop, perl, rootkit, phf. Метод k-ближайших соседей не позволяет определить следующие подтипы атаки: land, spy, rootkit, phf.

6. Вывод

Учитывая временные показатели, в качестве основной модели был выбран метод деревьев решений, время его работы оказалось минимальным и при определении типа атаки (0.08539), и при определении подтипа (0.11472).

При этом на рассматриваемых данных метрики качества метода деревьев решений принимают следующие значения: при определении типа атаки

Таблица 6.

Матрица несоответствия для определения типа атаки (5 классов)

Методы	benign		dos		probe		r2l	
Случайный лес	True	False	True	False	True	False	True	False
Градиентный бустинг (Scikit-learn)	32037	4	129283	4	1317	3	358	11
Градиентный бустинг (CatBoost)	31777	264	129215	72	1291	29	283	86
Логическая регрессия	32003	38	129277	10	1304	16	353	16
Наивный байесовский классификатор	31984	57	129241	46	1276	44	335	34
Деревья решений	29287	2754	93943	35344	1278	42	176	193
Метод опорных векторов	32018	23	129277	10	1315	5	359	10
Метод k-ближайших соседей	32019	22	129277	10	1303	17	345	24

Таблица 7.

Точность определения подтипа атаки (23 класса) – часть 1

Классы	Random Forest	Градиентный бустинг Scikit-learn	Градиентный бустинг CatBoost	Деревья решений
back	723/723	721/723	723/723	723/723
warezmaster	4/7	0/7	2/7	3/7
land	0/6	0/6	1/6	0/6
guess_passwd	19/20	0/20	20/20	19/20
imap	1/3	0/3	2/3	0/3
ipsweep	387/391	356/391	23/391	390/391
ftp_write	4/6	0/6	3/6	3/6
spy	0/1	1/1	1/1	0/1
multihop	1/3	0/3	0/3	0/3
neptune	35268/35270	34633/35270	1454/35270	35265/35270
nmap	73/75	44/75	75/75	75/75
normal	32038/32041	27229/32041	23043/32041	32017/32041
loadmodule	0/1	0/1	0/1	1/1
perl	1/1	0/1	1/1	1/1
pod	81/84	50/84	84/84	81/84
warezclient	327/329	313/329	249/329	324/329
rootkit	1/1	0/1	0/1	0/1
satan	523/525	474/525	0/525	521/525
smurf	92868/92868	92690/92868	92787/92868	92866/92868
phf	0/1	0/1	0/1	0/1
portsweep	336/336	322/336	336/336	336/336
teardrop	325/328	166/328	115/328	326/328
buffer_overflow	7/7	0/7	7/7	7/7

Таблица 8.

Точность определения подтипа атаки (23 класса) – часть 2

Классы	Логическая регрессия	Наивный байесовский классификатор	Метод опорных векторов	Метод k-ближайших соседей
back	722/723	722/723	723/723	721/723
warezmaster	4/7	2/7	3/7	7/7
land	1/6	1/6	0/6	0/6
guess_passwd	20/20	20/20	19/20	19/20
imap	1/3	2/3	1/3	1/3
ipsweep	382/391	23/391	382/391	389/391
ftp_write	4/6	3/6	3/6	5/6
spy	0/1	1/1	0/1	0/1
multihop	0/3	0/3	0/3	1/3
neptune	3526/35270	3526/35270	1454/35270	35266/35270
nmap	70/75	75/75	70/75	72/75
normal	31988/32041	23043/32041	32020/32041	32013/32041
loadmodule	0/1	0/1	1/1	1/1
perl	0/1	1/1	0/1	1/1
pod	81/84	84/84	81/84	81/84
warezclient	325/329	249/329	324/329	326/329
rootkit	0/1	0/1	0/1	0/1
satan	510/525	488/525	519/525	519/525
smurf	92867/92868	92867/92868	92787/92868	92867/92868
phf	0/1	0/1	0/1	0/1
portsweep	335/336	336/336	334/336	334/336
teardrop	305/328	115/328	316/328	325/328
buffer_overflow	7/7	7/7	7/7	7/7

(benign = 32018/23, dos = 129277/10, probe = 1315/5, r2l = 359/10, u2r = 3/7), при определении подтипа атаки (back = 723/723, warezmaster = 3/7, land = 0/6, guess_passwd = 19/20, imap = 0/3, ipsweep = 390/391, ftp_write = 3/6, spy = 0/1, multihop = 0/3, neptune = 35265/35270, nmap = 75/75, normal = 32017/32041, loadmodule = 1/1, perl = 1/1, pod = 81/84, warezclient = 324/329, rootkit = 0/1, satan = 521/525, smurf = 92866/92868, phf = 0/1, portsweep = 336/336, teardrop = 326/328, buffer_overflow = 7/7).

В обоих случаях данный алгоритм является одним их наиболее точных и безошибочных (для 5 классов параметр accuracy составляет 0.999662, для 23 классов – 0.999576). Наиболее близко к методу деревьев решений подошел только алгоритм случайного леса (для 5 классов параметр accuracy составляет 0.999834, для 23 классов – 0.999748), однако, последний довольно сильно уступает в скорости. Модель плохо определяет только классы с очень маленьким количеством членов, что видно при определении типа атаки, где плохо идентифицировался класс u2r (3 из 7 верно определенных члена), и при определении подтипа атаки, где классы land, imap, spy, multihop, rootkit и phf не идентифицировались (содержат от 1 до 6 членов). Но все остальные классы идентифицируются моделью либо без ошибок, либо с их минимальным количеством, что подтверждается в табл. 6 и 7.

Время работы данной модели оказалось минимальным, поэтому можно предположить, что это оптимальная модель для обработки рассматриваемой выборки данных.

Литература

1. *Singh A., Gupta B. B. Distributed denial-of-service (DDoS) attacks and defense mechanisms in various web-enabled computing platforms: issues, challenges, and future research directions // International Journal on Semantic Web and Information Systems (IJSWIS). – 2022. – Т. 18. – №. 1. – С. 1–43. DOI: 10.4018/IJSWIS.297143*
2. *Eliyan L. F., Di Pietro R. DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges // Future Generation Computer Systems. – 2021. – Т. 122. – С. 149–171. DOI: 10.1016/j.future.2021.03.011*
3. *Hu Q. et al. A rotating machinery fault diagnosis method based on multi-scale dimensionless indicators and random forests // Mechanical systems and signal processing. – 2020. – Т. 139. – С. 106609. DOI: 10.1016/j.ymssp.2019.106609*
4. *Nhat-Duc H., Van-Duc T. Comparison of histogram-based gradient boosting classification machine, random Forest, and deep convolutional neural network for pavement raveling severity classification // Automation in Construction. – 2023. – Т. 148. – С. 104767. DOI: 10.1016/j.autcon.2023.104767*
5. *Hancock J. T., Khoshgoftaar T. M. CatBoost for big data: an interdisciplinary review //Journal of big data. – 2020. – Т. 7. – №. 1. – С. 94. DOI: 10.1186/s40537-020-00369-8*
6. *Schober P., Vetter T. R. Logistic regression in medical research //Anesthesia & Analgesia. – 2021. – Т. 132. – №. 2. – С. 365–366. DOI: 10.1213/ANE.0000000000005247*
7. *Wickramasinghe I., Kalutarage H. Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation //Soft Computing. – 2021. – Т. 25. – №. 3. – С. 2277–2293. DOI: 10.1007/s00500-020-05297-6*
8. *Priyanka, Kumar D. Decision tree classifier: a detailed survey // International Journal of Information and Decision Sciences. – 2020. – Т. 12. – №. 3. – С. 246–269. DOI: 10.1504/IJIDS.2020.108141*
9. *Pisner D. A., Schnyer D. M. Support vector machine // Machine learning. – Academic Press, 2020. – С. 101–121. DOI: 10.1016/B978-0-12-815739-8.00006-7*
10. *Sinaga K. P., Yang M. S. Unsupervised K-means clustering algorithm // IEEE access. – 2020. – Т. 8. – С. 80716–80727. DOI: 10.1109/ACCESS.2020.2988796*
11. *Oyewole G. J., Thopil G. A. Data clustering: application and trends //Artificial Intelligence Review. – 2023. – Т. 56. – №. 7. – С. 6439-6475. DOI: 10.1007/s10462-022-10325-y*
12. *Ren Y. et al. Deep clustering: A comprehensive survey //IEEE Transactions on Neural Networks and Learning Systems. – 2024. DOI: 10.1109/TNNLS.2024.3403155*
13. *Antoniadis A., Lambert-Lacroix S., Poggi J. M. Random forests for global sensitivity analysis: A selective review // Reliability Engineering & System Safety. – 2021. – Т. 206. – С. 107312. DOI: 10.1016/j.ress.2020.107312*
14. *Aria M., Cuccurullo C., Gnasso A. A comparison among interpretative proposals for Random Forests // Machine Learning with Applications. – 2021. – Т. 6. – С. 100094. DOI: 10.1016/j.mlwa.2021.100094*
15. *Bo Y. et al. Real-time hard-rock tunnel prediction model for rock mass classification using CatBoost integrated with Sequential Model-Based Optimization // Tunnelling and underground space technology. – 2022. – Т. 124. – С. 104448. DOI: 10.1016/j.tust.2022.104448*

References

1. *Singh A., Gupta B. B. Distributed denial-of-service (DDoS) attacks and defense mechanisms in various web-enabled computing platforms: issues, challenges, and future research directions //International Journal on Semantic Web and Information Systems (IJSWIS). – 2022. – Т. 18. – №. 1. – С. 1–43. DOI: 10.4018/IJSWIS.297143*
2. *Eliyan L. F., Di Pietro R. DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges // Future Generation Computer Systems. – 2021. – Т. 122. – С. 149–171. DOI: 10.1016/j.future.2021.03.011*
3. *Hu Q. et al. A rotating machinery fault diagnosis method based on multi-scale dimensionless indicators and random forests // Mechanical systems and signal processing. – 2020. – Т. 139. – С. 106609. DOI: 10.1016/j.ymssp.2019.106609*

4. Nhat-Duc H., Van-Duc T. Comparison of histogram-based gradient boosting classification machine, random Forest, and deep convolutional neural network for pavement raveling severity classification //Automation in Construction. – 2023. – Т. 148. – С. 104767. DOI: 10.1016/j.autcon.2023.104767
5. Hancock J. T., Khoshgoftaar T. M. CatBoost for big data: an interdisciplinary review //Journal of big data. – 2020. – Т. 7. – №. 1. – С. 94. DOI: 10.1186/s40537-020-00369-8
6. Schober P., Vetter T. R. Logistic regression in medical research //Anesthesia & Analgesia. – 2021. – Т. 132. – №. 2. – С. 365-366. DOI: 10.1213/ANE.0000000000005247
7. Wickramasinghe I., Kalutarage H. Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation //Soft Computing. – 2021. – Т. 25. – №. 3. – С. 2277–2293. DOI: 10.1007/s00500-020-05297-6
8. Priyanka, Kumar D. Decision tree classifier: a detailed survey //International Journal of Information and Decision Sciences. – 2020. – Т. 12. – №. 3. – С. 246–269. DOI: 10.1504/IJIDS.2020.108141
9. Pisner D. A., Schnyer D. M. Support vector machine //Machine learning. – Academic Press, 2020. – С. 101-121. DOI: 10.1016/B978-0-12-815739-8.00006-7
10. Sinaga K. P., Yang M. S. Unsupervised K-means clustering algorithm //IEEE access. – 2020. – Т. 8. – С. 80716-80727. DOI: 10.1109/ACCESS.2020.2988796
11. Oyewole G. J., Thopil G. A. Data clustering: application and trends //Artificial Intelligence Review. – 2023. – Т. 56. – №. 7. – С. 6439–6475. DOI: 10.1007/s10462-022-10325-y
12. Ren Y. et al. Deep clustering: A comprehensive survey //IEEE Transactions on Neural Networks and Learning Systems. – 2024. DOI: 10.1109/TNNLS.2024.3403155
13. Antoniadis A., Lambert-Lacroix S., Poggi J. M. Random forests for global sensitivity analysis: A selective review //Reliability Engineering & System Safety. – 2021. – Т. 206. – С. 107312. DOI: 10.1016/j.ress.2020.107312
14. Aria M., Cuccurullo C., Gnasso A. A comparison among interpretative proposals for Random Forests //Machine Learning with Applications. – 2021. – Т. 6. – С. 100094. DOI: 10.1016/j.mlwa.2021.100094
15. Bo Y. et al. Real-time hard-rock tunnel prediction model for rock mass classification using CatBoost integrated with Sequential Model-Based Optimization //Tunnelling and underground space technology. – 2022. – Т. 124. – С. 104448. DOI: 10.1016/j.tust.2022.104448