ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
INFORMATION TECHNOLOGY AND TELECOMMUNICATIONS

УДК 004.85
DOI: 10.21822/2073-6185-2024-51-3-72-85 Оригинальная статья/Original article

Методика выявления аномалий в данных оценки кибератак с использованием Random Forest и градиентного бустинга в машинном обучении

А.С. Кечеджиев, О.Л. Цветкова, А.И. Дубровина
Донской государственный технический университет,
344002, г. Ростов-на-Дону, пл. Гагарина, 1, Россия

Резюме. Цель. Исследование направлено на обнаружение аномалий в данных с использованием моделей машинного обучения, в частности случайного леса и градиентного бустинга, для анализа активности сети и обнаружения кибератак. Тема исследования является актуальной, поскольку кибератаки становятся все более сложными и изощренными. Разработка эффективных методов обнаружения аномалий и защиты от киберугроз становится приоритетной задачей для организаций. **Метод.** Исследование основано на методах комплексного анализа, начиная с разработки алгоритма тестирования и загрузки данных для обучения и тестирования. Исследование проводится с помощью двух алгоритмов машинного обучения: Random Forest и градиентного Бустинга. Процесс включает в себя анализ важных признаков, визуализацию решений, оценку производительности моделей и анализ матриц ошибок для каждой категории атак. **Результат.** Модель Random Forest показала точность около 94% при использовании топ-10 важных признаков. Графическое представление решений позволяет понять, как модель принимает решения на основе признаков. Модель градиентного бустинга Xgboost достигла высокой точности и достоверности результатов. В классификационном отчете приводится подробное описание производительности моделей по каждой категории. **Вывод.** Проделанная работа представляет собой результат комплексного анализа модели машинного обучения, предназначенной для обнаружения кибератак. Она включает в себя несколько ключевых шагов и методов, позволяющих оценить эффективность модели, выделить наиболее важные признаки и проанализировать ее производительность для различных категорий атак.

Ключевые слова: аномалия в данных, машинное обучение, алгоритм Random Forest (случайный лес), модель градиентного бустинга

Для цитирования: А.С. Кечеджиев, О.Л. Цветкова, А.И. Дубровина. Методика выявления аномалий в данных оценки кибератак с использованием Random Forest и градиентного бустинга в машинном обучении. Вестник Дагестанского государственного технического университета. Технические науки. 2024; 51(3):72-85. DOI:10.21822/2073-6185-2024-51-3-72-85

Methodology for detecting anomalies in cyber attack assessment data using Random Forest and Gradient Boosting in machine learning

A.S. Kechedzhiev, O.L. Tsvetkova, A.I. Dubrovina
Don State Technical University,
1 Gagarina Square, Rostov-on-Don, 344002, Russia

Abstract. Objective. The research aims to detect anomalies in data using machine learning models, in particular random forest and gradient boosting, to analyze network activity and detect cyberattacks. The research topic is relevant as cyber attacks are becoming increasingly complex and sophisticated. Developing effective methods for detecting anomalies and protecting against cyber threats is becoming a priority for organizations. **Method.** The research is carried out using two machine learning algorithms: Random Forest and gradient boosting. The process includes analyzing important metrics, visualizing solutions, evaluating the performance of each model, and analyzing error matrices for attack categories. **Result.** The Random Forest model

showed an accuracy of about 94% when using the top 10 important features. The graph provides insight into how the model makes decisions based on features. The Xgboost gradient boosting model achieved high accuracy and reliability of results. The report provides a description of the model's performance for each category. **Conclusion.** The work done is the result of a comprehensive analysis of a machine learning model designed to detect cyberattacks. It includes several key steps and methods that allow us to evaluate the effectiveness of the model, identify important features, and analyze performance for various attacks.

Keywords: data anomaly, machine learning, Random Forest algorithm, gradient boosting model

For citation: A.S. Kechedzhiev, O.L. Tsvetkova, A.I. Dubrovina. Methodology for detecting anomalies in cyber attack assessment data using Random Forest and Gradient Boosting in machine learning. Herald of Daghestan State Technical University. Technical Sciences. 2024; 51(3):72-85. DOI:10.21822/2073-6185-2024-51-3-72-85

Введение. Аномалии в контексте машинного обучения представляют собой необычные или редкие события, отличающиеся от ожидаемого или типичного поведения данных. Они могут возникать из-за ошибок в данных, необычных ситуаций или проблем в процессе сбора данных. Аномалии могут проявляться в различных областях, таких как финансы (мошеннические транзакции), телекоммуникации (потеря сигнала), медицина (выявление заболеваний) и многие другие. Аномалии могут возникать по разным причинам, включая технические сбои, мошенничество, нештатные ситуации. Их обнаружение важно для обеспечения информационной безопасности, выявления проблем в системах, предотвращения финансовых потерь.

Использование методов машинного обучения для обнаружения аномалий позволяет автоматизировать процесс выявления необычных паттернов в данных [1]. Авторы предлагают модификацию алгоритма K-Means, которая обеспечивает повышение эффективности алгоритма при выявлении аномалий при обработке потоковых данных в реальном времени [2]. Следующая статья предлагает гибридное использование двух методов машинного обучения для выявления аномалий сетевого трафика [3]. Авторы утверждают, что данный алгоритм позволяет снизить количество ложноположительных срабатываний, повысить точность обнаружения и качество обслуживания. Известны научные работы, посвященные решению задачи обнаружения аномального поведения пользователей в системах на основе методов машинного обучения [4, 5]. В других работах авторы проводят исследование возможности обнаружения аномального поведения сетевого трафика на основе статистических методов при помощи машинного обучения и выполнения динамической аутентификации пользователей на основе анализа работы с компьютерной мышью [6, 7].

Постановка задачи. Random Forest и градиентный бустинг являются алгоритмами обучения с учителем, которые строят несколько деревьев решений и объединяют их для принятия окончательного решения. В контексте обнаружения аномалий Random Forest и градиентный бустинг могут использоваться для классификации точек данных как «нормальные» или «аномальные» на основе их признаков, что позволяет выявлять потенциальные аномалии в данных [8]. В настоящей статье приведены результаты исследования на примере двух алгоритмов машинного обучения.

Первый, рассматриваемый алгоритм машинного обучения — Random Forest (случайный лес), используется для решения задач классификации и регрессии. Он основан на идее комбинирования нескольких деревьев решений для получения более точных и стабильных прогнозов. Во время обучения создается множество деревьев решений и выполняется усреднение их результатов для получения более точного прогноза. Каждое дерево строится на основе подвыборки данных (bootstrap sample) и использует случайный набор признаков для разделения узлов дерева. Это способствует разнообразию деревьев в лесу, что помогает избежать переобучения и повышает обобщающую способность модели.

Второй алгоритм машинного обучения - модель градиентного бустинга, построен на идее комбинирования нескольких слабых моделей (например, деревьев решений) для

создания сильной ансамблевой модели. Основная идея заключается в последовательном добавлении новых моделей к ансамблю с фокусом на исправлении ошибок, сделанных предыдущими моделями [9]. Градиентный бустинг обучает новые модели таким образом, чтобы они исправляли недочеты или остатки предыдущих моделей. Он использует градиентный спуск для минимизации функции потерь, то есть для нахождения направления, в котором нужно изменить модель, чтобы улучшить ее прогнозы [10, 11].

Методы исследования. В рамках исследования будет использован набор данных, подходящий для задачи обнаружения аномалий. Сначала данные будут предварительно обработаны, включая процессы такие как очистка от выбросов, заполнение пропущенных значений и масштабирование признаков при необходимости. Затем данные будут поделены на обучающую и тестовую выборки. Далее каждый из алгоритмов будет обучен на обучающей выборке. Для Random Forest будет создано несколько деревьев решений на основе подвыборок данных, а для модели градиентного бустинга будут последовательно добавлены новые модели с фокусом на исправлении ошибок предыдущих моделей. После обучения каждая модель будет протестирована на тестовой выборке. Будут оценены и сравнены их производительность и качество предсказаний, основываясь на метриках, таких как точность, полнота, F1-мера.

Обсуждение результатов. 1. Разработка алгоритма проведения тестирования. В ходе исследования проведен анализ данных, используемых для обучения моделей машинного обучения с целью обнаружения кибератак. Данные представляют набор разнообразных параметров, связанных с сетевой активностью, таких как длительность соединения, тип протокола, состояние соединения, количество пакетов, переданных и полученных, размер переданных и полученных байтов, скорость передачи данных, параметры, связанные с конечной точкой соединения и другие. В данных присутствуют целевые метки, указывающие на наличие или отсутствие атаки в конкретном сценарии, они являются ключевыми для обучения модели машинного обучения на определение аномалий и выявление атак на сетевую инфраструктуру. Каждая строка в исходном файле обучения представляет отдельный экземпляр данных, содержащий значения параметров и метки, которые используются для обучения модели на основе машинного обучения. Данные будут проанализированы, обработаны и использованы для создания моделей, способных определять потенциальные кибератаки на основе предоставленных параметров сетевой активности. На рис. 1 представлен алгоритм действий, который был реализован в процессе работы над статьей.

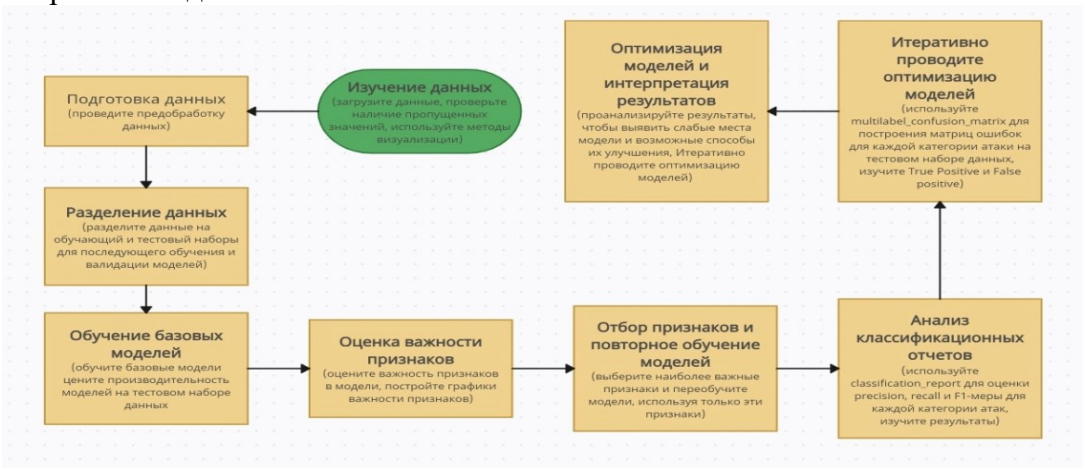

Рис. 1. Алгоритм действий при выполнении тестирования
Fig. 1. Algorithm of actions when performing testing

2. Загрузка данных для обучения и тестирования. Код представляет собой загрузку данных из файлов CSV в два различных набора данных: training и testing. При этом используется библиотека Pandas для чтения данных из указанных файлов.

После загрузки каждого набора данных код выводит информацию об их форме, то есть количество строк (наблюдений) и столбцов (признаков): Training-set содержит

82332 тысячи строк и 45 столбцов; Testing-set содержит 89838 тысячи строк и 45 столбцов.

Далее идет проверка на совпадение столбцов по названию и количеству. Когда все столбцы совпадают, нужно объединить тестовый и обучающийся наборы по оси строк создавая один data frame. Результат приведен на рис. 2 (фрагмент отображения data frame).

	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	.
0	0.000011	udp	-	INT	2	0	496	0	90909.0902	254	
1	0.000008	udp	-	INT	2	0	1762	0	125000.0003	254	
2	0.000005	udp	-	INT	2	0	1068	0	200000.0051	254	
3	0.000006	udp	-	INT	2	0	900	0	166666.6608	254	
4	0.000010	udp	-	INT	2	0	2126	0	100000.0025	254	

5 rows × 44 columns

Рис. 2. Фрагмент отображения data frame
Fig. 2. Data frame display fragment

Каждый столбец содержит определенное количество ненулевых значений (Non-Null Count). Некоторые столбцы имеют недостающие значения (например, dmean, trans_depth, attack_cat, label и другие), так как их Non-Null Count меньше общего количества записей (172169 вместо 172170), что указывает на наличие пропущенных данных.

Эти данные содержат различные числовые значения (с плавающей запятой и целочисленные) и категориальные данные (тип object). Это может быть важным для предварительной обработки данных, так как некоторые модели машинного обучения могут требовать числовые данные, а не категориальные. Также важно обратить внимание, что столбец label содержит числовые значения (float64), которые могут представлять целевую переменную для модели машинного обучения, а столбец attack_cat содержит категории атак. Возможно, что столбец attack_cat является целевой переменной для задачи классификации. Далее необходимо получить все уникальные виды атак.

Получим массив всех видов атак в столбце attack_cat. Массив атак: array (['Normal', 'Reconnaissance', 'Backdoor', 'DoS', 'Exploits', 'Analysis', 'Fuzzers', 'Worms', 'Shellcode', 'Generic', nan], dtype=object). Так же необходимо провести кодирование категориальных переменных в числовые коды, что может помочь в обработке данных для дальнейшего использования в моделировании. Из приведенных на рис. 3 данных видно, что столбцы proto, service, state были преобразованы в числовые значения.

	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	...	ct_dst_sport_ltm	ct_dst_src_ltm	is_ftp_log
0	0.000011	119	0	5	2	0	496	0	90909.0902	254	...	1.0	2.0	0
1	0.000008	119	0	5	2	0	1762	0	125000.0003	254	...	1.0	2.0	0
2	0.000005	119	0	5	2	0	1068	0	200000.0051	254	...	1.0	3.0	0
3	0.000006	119	0	5	2	0	900	0	166666.6608	254	...	1.0	3.0	0
4	0.000010	119	0	5	2	0	2126	0	100000.0025	254	...	1.0	3.0	0

Рис. 3. Фрагмент отображения data frame
Fig. 3. Data frame display fragment

3. Визуализация категорий атак. Получим визуализацию категорий атак, основанных на столбце attack_cat, в случаях, где метка label равна 1 (то есть, где атака обнаружена). На рис. 4 изображена диаграмма значений для категорий атак из столбца attack_cat, где метка label равна 1 (то есть, когда обнаружена атака). Каждое значение показывает количество случаев каждой категории атаки в этом датасете. Вот что означает каждое из этих значений:

- Exploits: 27292 случая атак, которые классифицированы как эксплуатации уязвимостей.
- Generic: 18871 случай, где использованы общие методы атаки.
- Fuzzers: 15222 случая, связанные с фаззингом (грубой проверкой на входные данные).
- DoS: 9537 случаев атак типа «Отказ в обслуживании» (Denial of Service).
- Reconnaissance: 8350 случаев, где проводилась разведка.
- Analysis: 1745 случаев анализа уязвимостей или сети.

– Backdoor: 1420 случаев использования задней двери для доступа к системе.
– Shellcode: 916 случаев использования shellcode для атаки.
– Worms: 107 случаев, где использовались черви для атаки.
– Normal: 0 случаев, которые были классифицированы как нормальная активность.

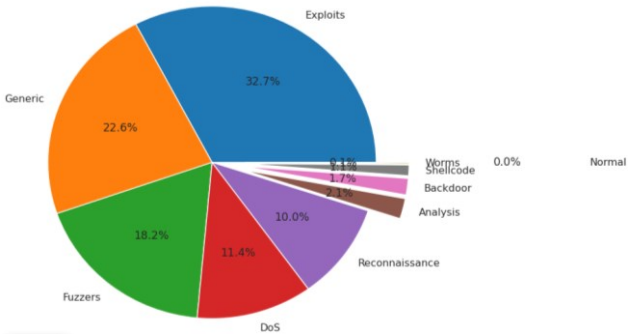

Рис. 4. Визуализация категорий атак
Fig. 4. Visualization of attack categories

4. Разделение данных для обучающей и тестовой сборки. Выполняется создание матрицы признаков x, удаление столбцов attack_cat и label из исходных данных. Это станет набором признаков для обучения модели. Массив меток y из столбца label будет целевой переменной для обучения модели. Разделение данных на обучающий и тестовый наборы: test_size=0.3 означает, что 30% данных будут выделены для тестирования, а random_state=11 устанавливает случайное начальное значение для разделения данных, обеспечивая воспроизводимость. Вывод информации о размере данных обучающего и тестового наборов после разделения:

– X_train shape: (120519, 42) — обучающий набор данных содержит 120519 строк (наблюдений) и 42 столбца (признаков);
– y_train shape: (120519,) — количество меток (целевых значений) в обучающем наборе данных. У нас есть 120519 меток для соответствующих строк в обучающем наборе;
– X_test shape: (51651, 42) — тестовый набор данных содержит 51651 строк (наблюдений) и также 42 столбца (признаков), что соответствует обучающему набору;
– y_test shape: (51651,) — количество меток (целевых значений) в тестовом наборе данных. У нас есть 51651 меток для соответствующих строк в тестовом наборе.

5. Модель дерева решений. Данный алгоритм использует модель решающего дерева (Decision Tree Classifier) для создания набора критериев для обнаружения кибератак. Он также выполняет поиск по сетке параметров (grid search) для оптимизации модели с целью максимизации метрики recall (полноты). Recall должен быть высоким, чтобы создать первый уровень защиты и обнаруживать как можно больше кибератак.

По результатам поиска по сетке наилучшие параметры для модели решающего дерева: Best parameters: {'criterion': 'gini', 'max_depth': 2, 'min_samples_leaf': 1, 'min_samples_split': 2}

Best recall score: 1.0

Критерий разделения: Gini (criterion='gini')

Максимальная глубина дерева: 2 (max_depth=2)

Минимальное количество выборок в листе: 1 (min_samples_leaf=1)

Минимальное количество выборок для разделения: 2 (min_samples_split=2)

Также лучший показатель recall, достигнутый вовремя кросс-валидации, составляет 1.0 (или 100%). Это означает, что модель на обучающих данных идеально справилась с обнаружением положительных классов (атак) без ошибок ложного отрицания (false negatives). Важно отметить, что идеальный показатель recall на обучающем наборе не всегда гарантирует такие же идеальные результаты на новых данных. Поэтому необходимо

провести тестирование модели на отдельном тестовом наборе, чтобы убедиться в ее обобщающей способности.

6. Визуализация правил. Фрагмент, кода представляет процесс визуализации правил и структуры решающего дерева, построенного моделью машинного обучения. Это нужно для понимания внутренней логики модели, интерпретации правил, принятых деревом решений, и визуального анализа структуры дерева. Такой анализ помогает увидеть, как модель делает прогнозы и какие признаки оказывают наибольшее влияние на принятие решений. На рис. 5 изображено текстовое представление древа правил.

```
:::::::> The RULES FOR HIGH RECALL RATE <:::::::
 |--- sttl <= 61.00
|   |--- sinpkt <= 0.00
|   |   |--- class: 1
|   |--- sinpkt >  0.00
|   |   |--- class: 0
|--- sttl >  61.00
|   |--- synack <= 0.04
|   |   |--- class: 1
|   |--- synack >  0.04
|   |   |--- class: 1
```

Рис. 5. Текстовое представление древа правил
Fig. 5. Textual representation of the rule tree

Графическое представление части правил, которые модель решающего дерева использовала для предсказания классов, изображено на рис. 6. Модель решающего дерева построила дерево принятия решений, которое использует различные характеристики данных для классификации каждого экземпляра.

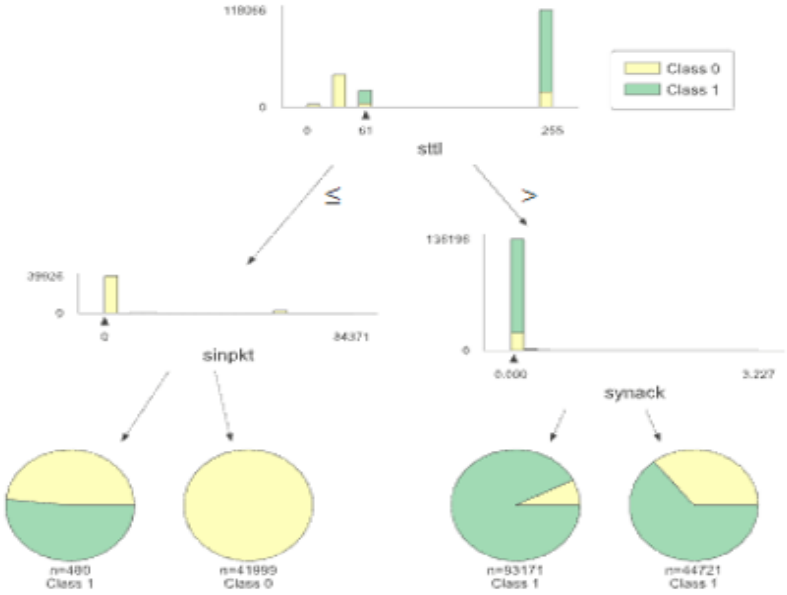

Рис. 6. Графическое представление древа правил
Fig. 6. Graphical representation of the rule tree

Интерпретация правил:

Если sttl (Time to Live) меньше или равно 61.00:

Если sinpkt (Interpacket Arrival Time) меньше или равен 0.00, модель предсказывает класс 1 (положительный класс).

Если sinpkt больше 0.00, модель предсказывает класс 0 (отрицательный класс).

Если sttl больше 61.00:

Если synack (SYN ACK Service Time) меньше или равен 0.04, модель предсказывает класс 1.

Если synack больше 0.04, модель также предсказывает класс 1. Например, если у нас есть новые данные, модель использует эти правила, чтобы определить, какой класс (1 или 0) должен быть присвоен каждому экземпляру на основе значений его характеристик (sttl, sinpkt, synack и т.д.).

7. Фильтрация данных на предмет потенциальных атак. Данная часть кода фильтрует тестовый набор данных X_test на основе некоторых правил и отображает процент данных, которые соответствуют этим правилам.

```
X_test = X_test.reset_index(drop=True)
rules= "(sttl <= 61.00 & sinpkt<= 0.00) | (sttl > 61.00)"
ind = X_test.query(rules).index
X_test_2 = X_test.loc[ind,:]
y_test_2 = y_test[ind]
print(X_test.shape)
print (X_test_2. shape)
print ("filtered data», (1- np.round(X_test_2.shape[0] / X_test.shape[0],2))*100, "%")
```

Из результатов кода видно следующее: исходный размер тестового набора данных X_test составляет (77302, 42), что означает 77302 строк и 42 столбца.

После применения заданных правил и фильтрации данных, размер X_test_2 составляет (59425, 42), что означает, что после фильтрации осталось 59425 строк и 42 столбца. Процент отфильтрованных данных составляет 23.0%. Это значение рассчитывается как отношение количества строк в отфильтрованном наборе X_test_2 к общему количеству строк в исходном тестовом наборе X_test. Это указывает на то, что после применения заданных правил и фильтрации данных остается только 23% от исходного тестового набора. Оставшиеся данные соответствуют заданным условиям rules.

8. Оценка производительности модели с помощью библиотеки Scikit-learn. Функция model_evaluation предназначена для оценки производительности модели машинного обучения. Она принимает модель в качестве входного аргумента model, обучает ее на обучающих данных, предсказывает значения на тестовых данных и оценивает несколько ключевых метрик (рис. 7).

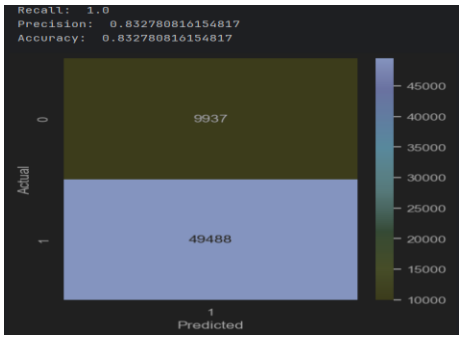

Рис. 7. Оценка производительности модели
Fig. 7. Model performance evaluation

С учетом этих данных можно сделать вывод, что модель достаточно хорошо выявляет реальные положительные случаи (Recall = 1.0) и демонстрирует высокую общую точность (Accuracy около 83%), что может быть удовлетворительным для многих задач классификации. Однако, стоит обратить внимание на Precision в 83%, чтобы оценить, насколько уверенно модель классифицирует положительные случаи. Precision равное 83% означает, что из всех случаев, которые модель отметила, как положительные, около 83% действительно являются таковыми. Тем не менее, решение о том, является ли это хорошим или плохим результатом, зависит от контекста конкретной задачи. В некоторых случаях, где критически важно минимизировать ложноположительные результаты, Precision 83% может быть недостаточно высоким. Например, в медицинских задачах неверное обозначение заболевания как отсутствующее может быть более критичным, чем ложное обозначение здорового человека как больного. Поэтому оценка Precision требует анализа в контексте конкретной задачи и баланса между минимизацией ложноположительных и ложноотрицательных результатов.

9. Модель машинного обучения Random Forest. Создадим модель случайного леса с заданным random_state, что обеспечивает воспроизводимость результатов. Получим

метрики оценки производительности модели, такие как recall, precision, accuracy и матрицу ошибок (confusion matrix) с помощью графика (рис. 8).

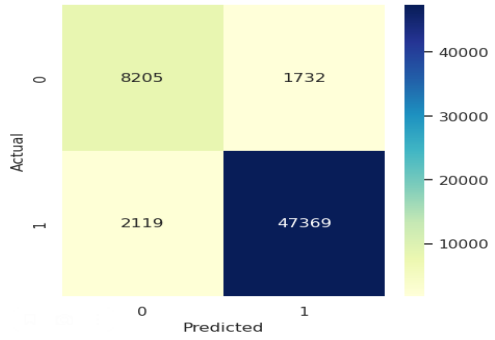

Рис. 8. Матрица ошибок
Fig. 8. Error matrix

Перенесем в переменную results производительность случайного леса, которая нам интересна для сравнения с другими моделями или для дальнейшего анализа.

Получим: Recall: 0.9571815389589395
Precision: 0.9647257693326001
Accuracy: 0.9351956247370635

Данные метрики указывают на хорошую производительность модели на нашем тестовом наборе данных. Высокий показатель Recall является особенно важным в задачах обнаружения атак, поскольку он показывает, что модель хорошо выявляет реальные атаки, минимизируя ложно отрицательные результаты.

10. Визуализация правил, присутствующих в дереве RandomForest. Получаем вывод в текстовом представление правил, использованных 100-м деревом в нашей модели случайного леса для классификации данных. Это позволяет нам понять, какие признаки и значения используются в этом конкретном дереве для принятия решений. Текстовое представление части дерева представлено на рис. 9.

```
|--- dpkts <= 0.50
|   |--- ct_srv_src <= 2.50
|   |   |--- sbytes <= 53.00
|   |   |   |--- sinpkt <= 30000.02
|   |   |   |   |--- sbytes <= 26.00
|   |   |   |   |   |--- class: 1.0
|   |   |   |   |--- sbytes >  26.00
|   |   |   |   |   |--- class: 0.0
|   |   |   |--- sinpkt >  30000.02
|   |   |   |   |--- class: 0.0
|   |   |--- sbytes >  53.00
|   |   |   |--- rate <= 0.11
|   |   |   |   |--- sbytes <= 62.50
|   |   |   |   |   |--- class: 1.0
|   |   |   |   |--- sbytes >  62.50
```

Рис. 9. Текстовое представление части дерева
Fig. 9. Text representation of a part of the tree

Каждый узел в дереве представляет собой условие для разделения данных, а листовые узлы содержат прогнозируемое значение. Имеются некоторые ключевые элементы в этом текстовом представлении:

– Узлы дерева содержат условия, например, dpkts <= 0.50 или sbytes > 62.50;
– Если условие истинно, переходите к левому поддереву, если ложно — к правому;
– Листовые узлы содержат значения (в данном случае class: 1.0 или class: 0.0), которые являются прогнозируемым классом или значением.

Это дает представление о том, как модель принимает решения на основе различных признаков.

11. Оценка с помощью модели градиентного бустинга библиотеки Xgboost. График производительности модели машинного обучения представлен на рис. 10. Метрики Recall, Precision и Accuracy являются показателями производительности модели машинного обучения.

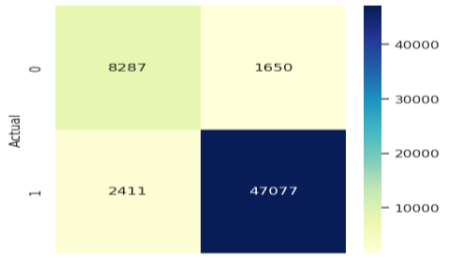

Рис. 10. График производительности модели
Fig. 10. Model performance graph

Recall (полнота) - это доля истинных положительных результатов, которые были правильно предсказаны моделью относительно всех реальных положительных результатов. Значение 0.95 говорит о том, что модель правильно классифицировала 95% всех истинно положительных случаев.

Precision (точность) - это доля истинных положительных результатов среди всех предсказанных положительных результатов. Значение 0.97 указывает на то, что 97% объектов, которые модель предсказала как положительные, действительно являются положительными. Accuracy (точность классификации) - это общая доля правильных предсказаний, сделанных моделью. Значение 0.93 означает, что модель правильно предсказала класс объекта в 93% случаев.

Метрики позволяют оценить качество работы модели: Recall и Precision важны для баланса между количеством правильно классифицированных объектов и количеством ложноположительных и ложноотрицательных результатов, в то время как Accuracy — это общая метрика правильности классификации. Например, если важно избежать ложноотрицательных результатов, Recall будет ключевой метрикой. Если важно избегать ложноположительных результатов, более важной становится Precision.

Визуализация дерева решений, используемого в модели XGBoost, помогает лучше понять, как модель принимает решения. Это полезно для интерпретации работы модели, особенно когда используются алгоритмы градиентного бустинга, такие как XGBoost. Путем визуализации дерева можно увидеть, какие функции (признаки) и значения используются в отдельном дереве решений, входящем в композицию модели XGBoost, для принятия конкретных решений.

12. Корреляции. Для улучшения понимания переменных, участвующих в обнаружении кибератак, необходимо проанализировать сетевые данные. Диаграммы корреляций могут помочь визуализировать, как различные переменные связаны друг с другом и с кибератаками (рис. 11).

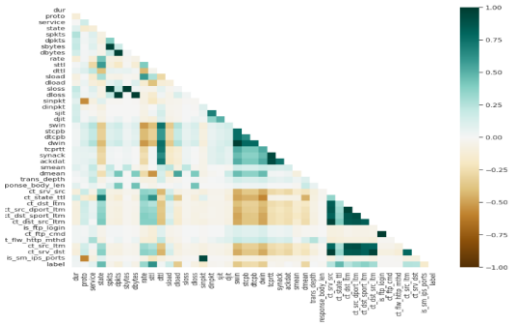

Рис. 11. Диаграмма корреляции
Fig. 11. Correlation diagram

Кроме того, модели случайного леса могут помочь определить важность различных признаков при прогнозировании целевой переменной (кибератак). Можно сравнить ранжирование признаков из случайного леса с результатами корреляционного анализа, чтобы лучше понять ключевые признаки, на которые стоит обратить внимание для эффективного обнаружения кибератак. Получим тепловую карту (heatmap), которая показывает корреляцию между различными переменными и целевой переменной label, которая, вероятно, связана с наличием или отсутствием кибератак (рис. 12).

Рис. 12. Признаки, коррелирующие с меткой
Fig. 12. Features correlating with the label

Следующие переменные положительно коррелируют с кибератаками:

- sttl: Время жизни сетевых пакетов от источника к получателю;
- ct_state_ttl и state: отражают различные этапы TCP-соединений и могут быть связаны со сканированием портов, SYN-флудом или DDoS-атаками;
- ct_dst_sport_ltm: Количество соединений от одного источника к тому же порту назначения в короткий промежуток времени;
- rate: может отображать различные типы частоты или скорости сетевого трафика.

Следующие переменные имеют отрицательную корреляцию с кибератаками: swin; dload. Положительная корреляция переменных с кибератаками подразумевает, что при увеличении значений этих переменных вероятность возникновения кибератак также может увеличиваться. Отрицательная корреляция же означает, что с увеличением значений этих переменных вероятность кибератак склонна снижаться.

13. Ранжирование объектов из случайного леса. Необходимо оценить важности признаков в модели случайного леса. feature_imp создает DataFrame, в котором содержатся названия признаков (X.columns) и их соответствующие значения важности, рассчитанные с использованием rf.feature_importances_ из модели случайного леса (rf). Они упорядочиваются по убыванию важности (рис. 13). Таблица на рис. 13 показывает важность различных признаков для модели, оцененную с помощью метода случайного леса. Колонка Name содержит названия признаков, а Importance отображает их значимость для модели. Важность измеряется относительно других признаков: чем больше значение Importance, тем важнее признак для модели. Например, признак sttl имеет наивысшую оценку важности, равную 0.127183, что означает, что этот признак считается самым важным при обучении модели.

	Name	Importance
0	sttl	0.127183
1	ct_state_ttl	0.098745
2	rate	0.056321
3	dload	0.049838
4	sload	0.045773
5	sbytes	0.043195
6	ct_srv_dst	0.040537
7	smean	0.039754
8	ct_dst_src_ltm	0.038934
9	tcprtt	0.033016

Рис. 13. Представление данных о важности различных признаков для модели в виде таблицы
Fig. 13. Presentation of data on the importance of various features for the model in the form of a table

После этого используется feat_importances, который строит горизонтальную столбчатую диаграмму для двадцати наиболее важных признаков из модели случайного леса (рис. 14). Каждый столбец этого графика представляет важность соответствующего признака. Выберем только 10 лучших признаков и найдем их связь с типом кибератаки.

Вывод: ['sttl', 'ct_state_ttl','rate','dload','sload', 'sbytes','ct_srv_dst','smean', 'ct_dst_src_ltm','tcprtt']. Далее создадим новый набор данных X, содержащий только топ-10 наиболее важных признаков, определенных предыдущим анализом важности признаков.

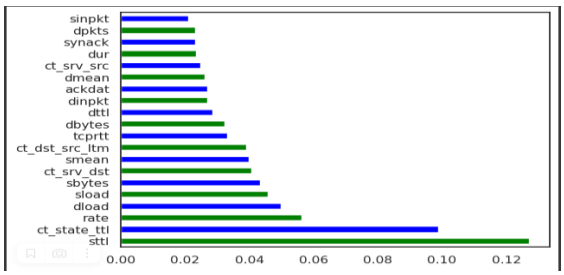

Рис. 14. Диаграмма для двадцати наиболее важных признаков из модели случайного леса
Fig. 14. Plot for the top twenty features from the random forest model

Затем модель случайного леса обучается на этих выбранных признаках, а точность (accuracy) модели оценивается на тестовом наборе данных. Полученное значение точности (acc = 0.9443869498848672). Accuracy говорит о том, что модель, использующая только эти 10 самых важных признаков, правильно классифицирует примерно 94.43% тестовых данных. Это предоставляет информацию о том, что эти 10 признаков, согласно важности, достаточно информативны для модели для высокого уровня предсказательной способности.

14. Обучение модели Random Forest с категорией атаки в качестве меток прогнозирования. Значение точности (accuracy) модели случайного леса, равное 0.824, означает, что модель правильно предсказывает категории атак в 82.4% случаев на тестовом наборе данных. Это показывает общую эффективность модели в предсказании категорий атак. График, который отображается с помощью sns.heatmap, представляет собой матрицу ошибок (confusion matrix) (рис.15), он позволяет оценить, сколько объектов каждого класса было предсказано правильно или ошибочно. Визуализация содержит информацию о результатах предсказаний для разных категорий атак, где по диагонали матрицы расположены правильные предсказания, а вне диагонали — ошибочные.

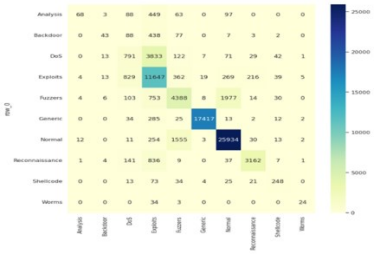

Рис. 15. Матрица ошибок
Fig. 15. Error matrix

15. Классификационный отчет. Классификационный отчет показывает метрики точности (precision), полноты (recall) и F1-меры для каждой категории атак, а также для среднего (macro avg) и взвешенного (weighted avg) усредненных значений этих метрик по всем классам (рис. 16).

	precision	recall	f1-score	support
Analysis	0.76	0.09	0.16	768
Backdoor	0.52	0.07	0.12	658
DoS	0.38	0.16	0.23	4909
Exploits	0.63	0.87	0.73	13403
Fuzzers	0.66	0.60	0.63	7283
Generic	1.00	0.98	0.99	17790
Normal	0.91	0.93	0.92	27814
Reconnaissance	0.91	0.75	0.82	4198
Shellcode	0.63	0.59	0.61	418
Worms	0.69	0.39	0.50	61
accuracy			0.82	77302
macro avg	0.71	0.54	0.57	77302
weighted avg	0.82	0.82	0.81	77302

Рис. 16. Классификационный отчет
Fig. 16. Classification report

Эти метрики дают представление о производительности модели машинного обучения для каждой конкретной категории атак и в целом по всем категориям.

Метрики можно интерпретировать следующим образом:

– Precision (точность): это доля правильно предсказанных положительных случаев из всех предсказанных положительных случаев. Например, для категории Normal

precision равен 0.91, что означает, что 91% из всех предсказанных как Normal действительно являются Normal;

– Recall (полнота): это доля правильно предсказанных положительных случаев из всех истинных положительных случаев. Например, для категории Normal recall равен 0.93, что означает, что модель поймала 93% всех реальных Normal случаев;
– F1-score (F1-мера): это среднее гармоническое между точностью и полнотой. Это полезная метрика для сбалансированной оценки производительности модели, особенно при несбалансированных классах.

Значения precision, recall и F1-меры для каждой категории атак показывают, насколько хорошо модель справляется с предсказанием этой конкретной категории. Рассмотрим более точно матрицы ошибок каждой категории атак с использованием multilabel_confusion_matrix. Этот блок кода визуализирует матрицы ошибок для каждой категории атаки из тестового набора данных. Каждая матрица ошибок содержит четыре различных значения:

– True Negative (TN) Число в левом верхнем углу. Это количество образцов, которые были правильно предсказаны как отсутствующие для данной категории атаки;
– False Positive (FP): Число в верхнем правом углу. Это случаи, когда модель неправильно предсказывает наличие категории атаки, когда ее на самом деле нет;
– False Negative (FN): Число в правом нижнем углу. Оно показывает, сколько экземпляров данной категории атаки были неправильно предсказаны как отсутствующие (ложноотрицательные результаты);
– True Positive (TP): Число находится в нижнем левом углу. Это случаи, когда модель правильно предсказывает наличие категории атаки, и она действительно присутствует.

Каждая матрица ошибок показывает количество верных и неверных прогнозов для определенной категории атаки (рис. 17, 18).

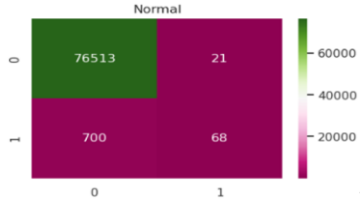

Рис. 17. Нормальное состояние без атак
Fig. 17. Normal state without attacks

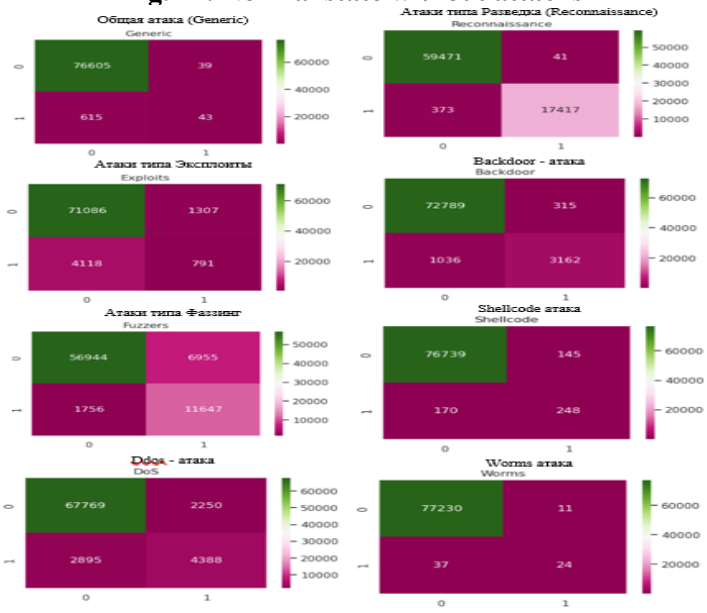

Рис. 18. Состояния при различных видах атак
Fig. 18. States for different types of attacks

Этот процесс полезен для понимания того, где модель делает ошибки в предсказаниях для каждой категории атак. Красные ячейки на диагонали матрицы показывают количество правильно предсказанных случаев, тогда как некоторые другие ячейки могут указывать на ложные срабатывания или пропущенные атаки. Каждая матрица соответствует категории атаки из набора данных, и они построены итеративно для всех категорий. Это позволяет быстро оценить производительность модели по каждой категории атак.

Вывод. Проделанная работа представляет собой комплексный анализ модели машинного обучения, предназначенной для обнаружения кибератак. Она включает в себя несколько ключевых шагов и методов, позволяющих оценить эффективность модели, выделить наиболее важные признаки и проанализировать ее производительность для различных категорий атак.

Оценка и анализ данных: изучение корреляций между переменными позволило выявить важные признаки, связанные с кибератаками; использование модели случайного леса для ранжирования признаков позволило выделить топ-10 наиболее значимых признаков для модели.

Оценка модели: обучение модели случайного леса на этих топ-10 признаках показало высокий уровень точности (около 94%), указывая на их информативность для предсказаний модели.

Интерпретация модели: визуализация дерева решений на основе этих 10 признаков помогла понять, как модель принимает решения и какие признаки оказывают большее влияние на классификацию.

Оценка на категории атак: модель обучалась и тестировалась с использованием категорий атак, показав приемлемую точность в предсказании типов атак (около 82%); анализ матриц ошибок для каждой категории атаки дал представление о том, где модель делает ошибки в предсказаниях для конкретных типов атак, что помогает улучшить ее работу.

Метрики оценки модели: классификационный отчет дал детальное представление о производительности модели по каждой категории атаки: precision, recall и F1-мера.

Библиографический список:

1. Гайдук, К. А. К вопросу о реализации алгоритмов выявления внутренних угроз с применением машинного обучения / К. А. Гайдук, А. Ю. Исхаков // Вестник СибГУТИ. – 2022. – Т. 16, № 4. – С. 80-95. – DOI 10.55648/1998-6920-2022-16-4-80-95. – EDN SGBSIH.
2. Савицкий, Д. Е. Выявление аномалий при обработке потоковых данных в реальном времени / Д. Е. Савицкий, М. Е. Дунаев, К. С. Зайцев // International Journal of Open Information Technologies. – 2022. – Т. 10, № 6. – С. 70-76. – EDN IGAWAO.
3. Токарев, Д.М. Обнаружение аномалий на основе машинного обучения с использованием сочетания алгоритмов K-MEAN и SMO / Д.М. Токарев, М.Г. Городничев // Телекоммуникации и информационные технологии. – 2023. – Т. 10, № 1. – С. 5-13. – EDN ILCJZP.
4. Мельник М. В., Котенко И. В. Обнаружение аномального поведения пользователей и сущностей в контейнерных системах на основе методов машинного обучения// Информационная безопасность регионов России (ИБРР-2023) : XIII Санкт-Петербургская межрегиональная конференция. Материалы конференции, Санкт-Петербург, 25–27 октября 2023 года. Санкт-Петербург: Санкт-Петербургское Общество информатики, вычислительной техники, систем связи и управления, 2023. – С. 97-98. – EDN QOBTZP.
5. Терских М.Г., Тишина Е.М. Обнаружение аномального поведения пользователей в журналах событий безопасности Windows с применением алгоритмов машинного обучения// Теория и практика современной науки. – 2018. – № 5(35). – С. 821-839. – EDN UYMTHC.
6. Сафин, А. Р. Обнаружение аномального поведения сетевого трафика на основе статистических методов при помощи машинного обучения / А. Р. Сафин // Информационная безопасность и защита персональных данных. Проблемы и пути их решения: Материалы XIII Межрегиональной научно-практической конференции, Брянск, 30 апреля 2021 года. – Брянск: Брянский государственный технический университет, 2021. – С. 228-231. – EDN UDRGDA.
7. Динамическая аутентификация пользователей на основе анализа работы с компьютерной мышью / А.В. Березникер, М.А. Казачук, И.В. Машечкин [и др.] // Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. – 2021. – № 4. – С. 3-16. – EDN XIQNIZ.

8. Попова, И.А. Обнаружение аномалий в наборе данных с помощью алгоритмов машинного обучения без учителя Isolation Forest и Local Outlier Factor / И.А. Попова // StudNet. – 2020. – Т. 3, № 12. – С. 1460-1470. – EDN XILRBX.
9. А. Асунсьон, Д. Ньюман. Репозиторий машинного обучения UCI, 2007.
10. М. М. Брейниг, Х.-П. Кригель, Р. Т. Нг и Дж. Сандер. LOF: идентификация локальных выбросов на основе плотности. Запись ACM SIGMOD, 29(2):93-104, 2000.
11. Т. Ши, С. Хорват. Неконтролируемое обучение со случайными лесными предикторами. Журнал вычислительной и графической статистики, 15 (1): 118-138, март 2006.

References:

1. Gaiduk, K. A. On the implementation of algorithms for identifying internal threats using machine learning / K. A. Gaiduk, A. Yu. Iskhakov. *Bulletin of SibSUTI.* 2022;16(4): 80-95. DOI 10.55648/1998-6920-2022-16-4-80-95. - EDN SGBSIH. (In Russ)
2. Savitsky D. E., M. E. Dunaev, K. S. Zaitsev. Detecting anomalies in real-time streaming data processing. *International Journal of Open Information Technologies*. 2022;10(6):70-76. - EDN IGAWAO. (In Russ)
3. Tokarev D.M., Gorodnichev M. G. Machine Learning-Based Anomaly Detection Using a Combination of K-MEAN and SMO Algorithms. *Telecommunications and Information Technologies.* 2023;10(1):5-13. - EDN ILCJZP. (In Russ)
4. Melnik, M. V. Detection of Anomalous Behavior of Users and Entities in Container Systems Based on Machine Learning Methods / M. V. Melnik, I. V. Kotenko. Information Security of Russian Regions (IBRR-2023): XIII St. Petersburg Interregional Conference. Conference Proceedings, St. Petersburg, October 25-27, 2023. - St. Petersburg: St. Petersburg Society for Informatics, Computer Engineering, Communications and Control Systems, 2023; 97-98. - EDN QOBTZP. (In Russ)
5. Terskikh M.G., E.M. Tishina. Detecting abnormal user behavior in Windows security event logs using machine learning algorithms. *Theory and practice of modern science*. 2018; 5(35):821-839. - EDN UYMTHC(In Russ)
6. Safin, A.R. Detecting abnormal network traffic behavior based on statistical methods using machine learning / A.R. Safin // Information security and personal data protection. Problems and solutions: Proceedings of the XIII Interregional Scientific and Practical Conference, Bryansk, April 30, 2021. - Bryansk: Bryansk State Technical University, 2021;228-231. - EDN UDRGDA. (In Russ)
7. Dynamic user authentication based on the analysis of work with a computer mouse / A.V. Bereznik, M.A. Kazachuk, I.V. Mashechkin [et al.] *Bulletin of Moscow University. Series 15: Computational Mathematics and Cybernetics*. 2021;4:3-16. - EDN XIQNIZ. (In Russ)
8. Popova I.A. Detecting anomalies in a dataset using unsupervised machine learning algorithms Isolation Forest and Local Outlier Factor StudNet. 2020;3(12):1460-1470. - EDN XILRBX. (In Russ)
9. Asuncion, D. Newman. UCI Machine Learning Repository, 2007.
10. M.M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Density-based local outlier identification. ACM SIGMOD Record, 29(2):93–104, 2000.
11. T. Shi and S. Horvath. Unsupervised learning with random forest predictors. Journal of Computational and Graphical Statistics, 15(1):118–138, March 2006.

Сведения об авторах:

Кечеджиев Александр Сергеевич, магистрант кафедры «Вычислительные системы и информационная безопасность»; Kechedzhiev.alex@mail.ru

Цветкова Ольга Леонидовна, кандидат технических наук, доцент, доцент кафедры «Вычислительные системы и информационная безопасность»; olga_cvetkova@mail.ru; ORCID: 0000-0003-4071-6313

Дубровина Ангелина Игоревна, ассистент кафедры «Вычислительные системы и информационная безопасность»; ministrelia69@yandex.ru; ORCID: 0009-0005-8562-9389

Information about authors:

Alexander S. Kechedzhiev, Master's Student, Department of Computer Systems and Information Security; Kechedzhiev.alex@mail.ru

Olga L. Tsvetkova, Cand. Sci. (Eng), Assoc. Prof., Assoc. Prof., Department of Computer Systems and Information Security; olga_cvetkova@mail.ru; ORCID: 0000-0003-4071-6313

Angelina I. Dubrovina, Assistant, Department of Computer Systems and Information Security; ministrelia69@yandex.ru; ORCID: 0009-0005-8562-9389

Конфликт интересов/Conflict of interest.

Авторы заявляют об отсутствии конфликта интересов/The authors declare no conflict of interest.

Поступила в редакцию/ Received 26.03.2024.

Одобрена после рецензирования / Reviced 17.04.2024.

Принята в печать /Accepted for publication 17.04.2024.