Надежность, том 26, №1, 2026 / Dependability 1, 2026
Методы и системы защиты информации, информационная безопасность

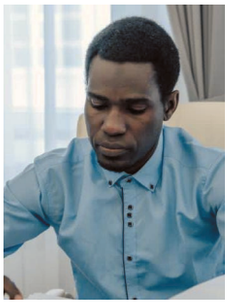
УДК 004.738.5:004.891.2:004.056.55
Оригинальная статья
https://doi.org/10.21683/1729-2646-2026-26-1-49-61

Применение ансамблевого обучения для определения типа вторжения в IoT

Using Ensemble Learning for Identifying the Type in IoT Intrusion

Нианг П.М.[1]*, Сидоренко В.Г.[1]
Niange P.M.[1]*, Sidorenko V.G.[1]

[1] РУТ(МИИТ), Москва, Российская Федерация
[1] Russian University of Transport (MIIT), Moscow, Russian Federation
** malickdiarra30@gmail.com*



Нианг П.М.

Сидоренко В.Г.

Резюме. Цель. *Целью работы является повышение качества многоклассовой классификации для систем обнаружения вторжений (IDS) в среде Интернета вещей (IoT). Целью исследования является определение влияния предварительной фильтрации бинарного трафика и применения ансамблевых моделей на точность прогнозирования, особенно для меньшинства классов атак, с учетом вычислительных ограничений сред IoT.* **Методы.** *Были изучены три архитектурных подхода: прямая многоклассовая классификация, прямая многоклассовая классификация (включая класс «нормальный») и иерархическая архитектура, основанная на начальном бинарном обнаружении с последующей классификацией по типу атаки. Были оценены восемь алгоритмов машинного обучения, а также три ансамблевых метода (мягкого голосования (Soft Voting Classifier (SVC)), жесткого голосования (Hard Voting Classifier (HVC)) и Stacking Classifier (SC)). Эксперименты проводились на наборе данных UNSW-NB15 с использованием таких метрик, как Precision, Recall и F1-score.* **Результаты.** *Результаты показывают, что прямая классификация обеспечивает лучшее общее покрытие атак (средняя оценка F1-score до 63% для градиентного бустинга (GBC)), но может потребовать больших затрат времени на обучение (более 2000 секунд для GBC). Иерархическая бинарная фильтрация значительно сокращает время вычислений, но может снизить производительность на некоторых редких классах. Алгоритмы GBC, случайный лес (RF) и дополнительные деревья (ET) выделяются своей производительностью. Среди ансамблевых методов наилучшие результаты (оценка F1-score 73,87%) демонстрирует SC, превосходящий индивидуальные классификаторы, но при этом требует очень много времени на обучение.* **Заключение.** *Данное исследование показывает, что внедрение бинарной фильтрации является актуальной стратегией для снижения вычислительных затрат, но необходимо найти компромисс между производительностью, охватом и эффективностью. GBC остается наиболее эффективным методом для редких атак, но из-за своей стоимости плохо подходит для встраиваемых систем. ET и RF представляют собой отличный компромисс между точностью и скоростью. SC, хотя и наиболее эффективен, требует значительных ресурсов. Научная новизна исследования заключается в систематической оценке иерархических и ансамблевых подходов к IDS в Интернете вещей, что открывает путь к созданию более надежных архитектур, адаптированных к задачам кибербезопасности IoT.*

Abstract. Aim. *The aim of this work is to improve the quality of multi-class classification for Intrusion Detection Systems (IDS) in the Internet of Things (IoT) environment. The goal of the research is to determine the impact of preliminary binary traffic filtering and the application of ensemble models on prediction accuracy, especially for minority attack classes, taking into account the computational constraints of IoT environments.* **Methods.** *Three architectural approaches were studied: direct multi-class classification, direct multi-class classification (including the “normal” class), and a hierarchical architecture based on initial binary detection followed by classification by attack type. Eight machine learning algorithms, as well as three ensemble methods (Soft Voting Classifier (SVC), Hard Voting Classifier (HVC), and Stacking Classifier (SC)), were evaluated. Experiments were conducted on the UNSW-NB15 dataset using metrics such as Precision, Recall, and F1-score.* **Results.** *The results show that direct classification provides better overall attack coverage (average F1-score up to 63% for Gradient Boosting Classifier (GBC)), but may require longer training times (over 2000 seconds for GBC). Hierarchical binary filtering significantly reduces computation time but can decrease performance for some rare classes. The GBC, Random Forest (RF), and Extra Trees (ET) algorithms stand out for their performance. Among the ensemble methods, the Stacking Classifier (SC) demonstrates the best results (F1-score of 73.87%), surpassing individual classifiers, although it also requires substantial training time.* **Conclusion.** *This research shows that implementing*

binary filtration is a relevant strategy for reducing computational costs, but a trade-off must be found between performance, coverage, and efficiency. GBC remains the most effective method for rare attacks but, due to its computational cost, is poorly suited for embedded systems. ET and RF represent an excellent compromise between accuracy and speed. SC, while the most effective, requires significant resources. The scientific novelty of the research lies in the systematic evaluation of hierarchical and ensemble approaches for IDS in IoT, paving the way for creating more robust architectures adapted to IoT cybersecurity tasks.

Ключевые слова: *Интернет вещей, обнаружение вторжений, алгоритмы машинного обучения, многоклассовая классификация.*
Keywords: *Internet of Things, intrusion detection, machine learning algorithms, multi-class classification.*

Для цитирования: *Нианг П.М., Сидоренко В.Г. Применение ансамблевого обучения для определения типа вторжения в IoT // Надежность. 2026. №1 С. 49-61. https://doi.org/10.21683/1729-2646-2026-26-1-49-61*
For citation: *Niange, P.M., Sidorenko, V.G. Using Ensemble Learning for Identifying the Type in IoT Intrusion. Dependability 2026;1: 49-61. https://doi.org/10.21683/1729-2646-2026-26-1-49-61*

Поступила: *23.10.2025* / **После доработки:** *06.11.2025* / **К печати:** *01.02.2026*
Received on: *23.10.2025* / **Revised on:** *06.11.2025* / **For printing:** *01.02.2026*

Введение

Системы Интернета вещей (*IoT*) все чаще подвергаются сложным кибератакам из-за своей неоднородности, ограниченности ресурсов и постоянной связанности. Системы обнаружения вторжений (*IDS*) на основе машинного обучения (*ML*) в настоящее время становятся эффективным подходом для выявления аномального поведения в этих средах [1, 2]. Однако наборы данных, используемые для *IDS*, такие как *UNSW-NB15*, демонстрируют высокий дисбаланс с большинством нормальных образцов [3, 4]. Этот дисбаланс отрицательно влияет на производительность многоклассовых классификаторов, особенно при детальном распознавании различных типов атак.

После предыдущих работ по выбору соответствующих наборов данных [5] и оценке результатов применения различных алгоритмов *ML* для решения задач компьютерной безопасности в среде *IoT* с использованием бинарной [6] и многоклассовой классификации [7] и их объединения [8] настоящее исследование авторов фокусируется на механизме обнаружения атак, основанном на иерархической архитектуре.

Точность работы алгоритмов бинарной классификации превышает 95% [6], а у алгоритмов многоклассовой классификации точность не превышает 75% [7]. Объединение этих алгоритмов в очень редких случаях позволяет повысить точность обнаружения атаки [8].

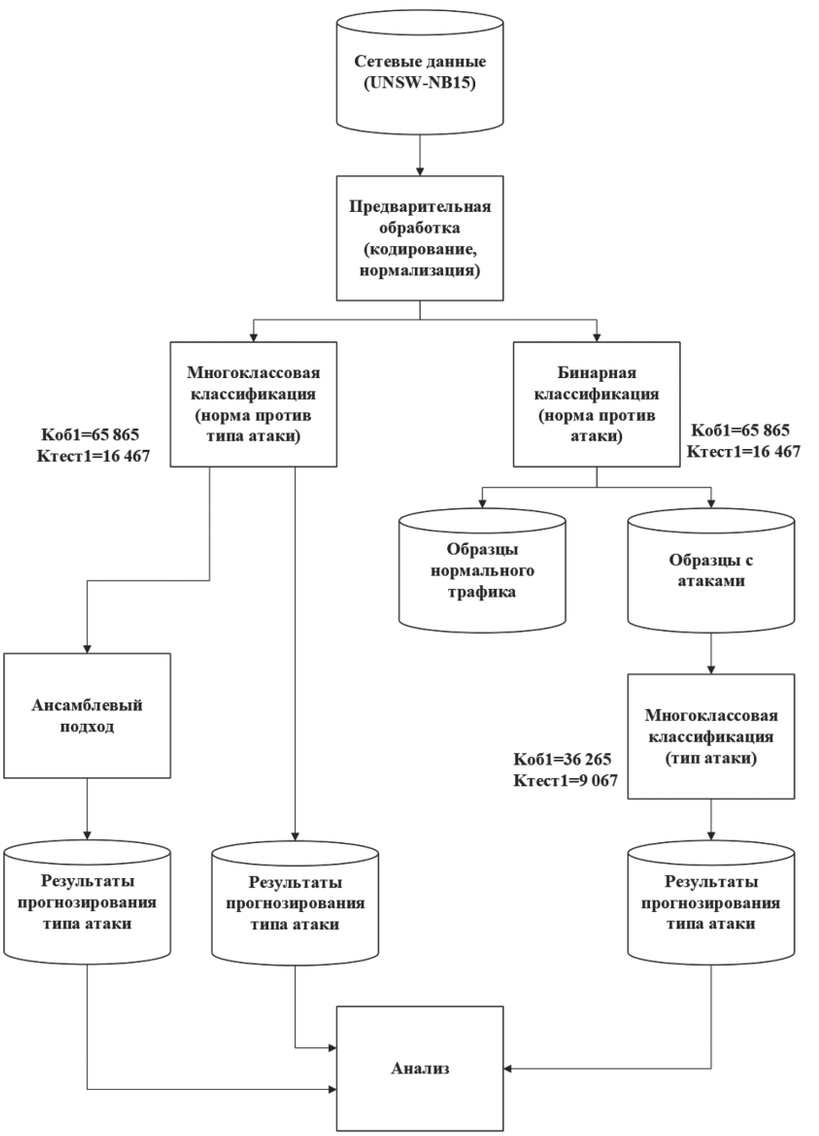


Рис. 1. План исследования

Цель этого исследования – найти путь повышения качества многоклассовой классификации для обнаружения вторжений в *IoT*, определив влияние предшествующей бинарной фильтрации и ансамбля моделей на точность прогнозирования, в частности для классов атак меньшинства, и выявив наиболее эффективные алгоритмы, принимая во внимание как производительность прогнозирования, так и вычислительные ограничения, характерные для сред *IoT*.

1. План исследования

Рис. 1 иллюстрирует ход проведенного исследования, в рамках которого были реализованы и оценены различные подходы:

- прямая многоклассовая классификация образцов (левая ветвь) с последующим ансамблевым подходом;
- прямая многоклассовая классификация (с «нормальным» классом) (центральная ветвь);
- иерархическая архитектура, основанная на первоначальном бинарном обнаружении с последующей классификацией типов атак, примененной к предположительно вредоносным образцам, выявленным в ходе бинарной классификации (без «нормального» класса) (правая ветвь).

В ходе исследования использована та же методология, что и в предыдущих работах [6-8], основанная на использовании библиотек языка программирования *Python* и среды *Python Jupyter Notebook*. Используются модели *ML*, сведения о качестве функционирования которых при решении задачи многоклассовой классификации уже имеются у авторов [7]:

- логистическая регрессия *Logistic Regression* (*LR*);
- случайный лес *Random Forest* (*RF*);
- *K*-ближайших соседей (*KNN)*;
- наивное дерево Байеса *Naive Bayes* (*NB*);
- дерево решений *Decision Tree* (*DT*);
- дополнительные деревья *Extra Trees* (*ET*);
- градиентный бустинг *Gradient Boosting Classifier* (*GBC*);
- нейронная сеть прямого распространения, состоящей из полностью связанных нейронов с нелинейными функциями активации *MLP.*

В отличие от предыдущих работ рассмотрены ансамблевые методы обучения [9, 10]: голосование (*Voting)* [11] и двухуровневая стратегия ансамблевого обучения (*Stacking)* [12 ,13], позволяющие объединять неоднородные модели, в отличие от бэггинга (*Bagging*) и бустинга (*Boosting*), которые, как правило, ограничены наборами однородных моделей [14]. В наших экспериментах использовались восемь перечисленных выше гетерогенных классификаторов в качестве базовых моделей и алгоритм *GBC* в качестве метаклассификатора из-за его хорошей производительности в качестве индивидуальной модели [9].

В ходе работы программы выполняются следующие шаги:

– подготовка данных;
– применение алгоритмов *ML*;
– определение значений метрик;
– оценка и сравнение качества исследуемых моделей.

Эксперименты проводились на наборе данных *UNSW-NB15*, который широко используется в литературе для оценки систем *IDS*, особенно в контексте *IoT*. Этот набор данных использовался авторами и ранее [3-8]. Извлеченные образцы охватывают следующие типы атак:

Фаззерс (*Fuzzerss*) – попытка вызвать зависание программы или сети путем подачи на нее случайно сгенерированных данных;

Анализ (*Analysis*) – различные атаки порта, сканирование, проникновение спама и html-файлов;

Бэкдоры (*Backdoor*) – скрытный обход системы безопасности системы для получения доступа к компьютеру или его данным;

Дос (*Dos*) – злонамеренная попытка сделать сервер или сетевой ресурс недоступным для пользователей, обычно это временное прерывание или приостановка предоставления услуг хоста, подключенного к Интернету;

Эксплойты (*Exploits*) – использование злоумышленником знания о проблеме безопасности в операционной системе или части программного обеспечения при эксплуатации уязвимости;

Общий (*Generic*) – атаки криптоанализа;

Разведка (*Reconnaissance*) – атаки, направленные на сбор информации;

Шеллкод (*Shellcode*) – использование маленького фрагмента программного кода в качестве полезной нагрузки в эксплуатации его уязвимости;

Черви (*Worms*) – копирование злоумышленником самого себя для распространения на другие компьютеры, часто используется в компьютерной сети для распространения себя и получения доступа к целевому компьютеру, полагаясь на сбои в системе безопасности.

Исходными данными для построения оригинальной модели *ML* для *IDS* является набор данных *UNSW_NB15_training-set*. Обучающая выборка включает в себя $K_{об1}$, равное 65 865 записей для бинарной классификации и прямой многоклассовой классификации, и $K_{об2}$, равное 36 265 записей для многоклассовой классификации после бинарной фильтрации. Тестовая выборка включает в себя $K_{тест1}$, равное 16 467 записей для бинарной классификации и прямой многоклассовой классификации и $K_{тест2}$, равное 9 067 записей для многоклассовой классификации после бинарной фильтрации.

2. Анализ результатов прямого и иерархического подхода

В табл. 1 представлены результаты для различных методов *ML*, применяемых к бинарной классификации. Общая производительность многоклассовой классификации суммирована в табл. 2 для двух представленных на рис. 1 способов ее выполнения: независимо и после

Табл. 1. Сравнение значений метрик различных алгоритмов бинарной классификации [6], $K_{тест1}$ = 16 467

	LR	*KNN*	*RF*	*NB*	*DT*	*ET*	*GBC*	*MLP*
Precision,%	93,02	95,19	97,69	79,53	97	98	96	96
Recall, %	92,98	95,11	97,68	76,84	97	98	96	96
F1-score, %	92,99	95,12	97,68	76,73	97	97	96	96
TN	6 922	7 146	7 256	6 611	7 207	7 262	7 119	7 153
FP	478	254	144	789	193	138	281	247
FN	678	551	238	3 025	365	278	327	353
TP	8 389	8 516	8 829	6 042	8 702	8 789	8 740	8 714
Тренировочное время, с	101,41	10,93	6,76	0,16	4,21	9,41	49,44	53,43
Время прогнозирования, с	0,00	16,74	0,19	0,04	0,01	0,21	0,05	0,02
Общее время, с	101,41	27,67	6,95	0,2	4,22	9,62	49,49	53,45
AUC	0,93	0,95	0,98	0,78	0,97	0,98	0,96	0,96
Precision – точность распознания типа атак; *Recall* – доля правильно классифицированных объектов классов от числа всех объектов класса; *F1-score* – гармоническое значение *Precision* и *Recall*, *TN* – истинно-положительный результат; *FP* – ложно-положительный результат; *FN*-ложно-отрицательный результат; *TP* – истинно-отрицательный результат; *AUC*(площадь под кривой) – инструмент для оценки.								

Табл. 2. Сравнение значений метрик рассмотренных алгоритмов многоклассовой классификации

Прямая многоклассовая классификация [7], $K_{тест1}$ = 16 467								
	LR	KNN	RF	NB	DT	ET	GBC	*MLP*
Precision, %	59	53	66	45	60	64	75	69
Recall, %	43	47	59	43	59	58	60	50
F1-score, %	42	48	61	29	59	59	63	51
Тренировочное время, с	15,10	0,70	44,29	97	7,24	28,95	2016,89	85,26
Время прогнозирования, с	0,02	68,63	1,10	0,38	0,03	1,54	1,08	0,04
Общее время, с	15,12	69,33	45,39	1,35	7,27	30,49	2017,97	85,30
Многоклассовая классификация после бинарной фильтрации, $K_{тест2}$ = 9 067								
Precision, %	43	38	69	28	73	57	73	0,04
Recall, %	39	31	51	21	59	49	55	0,09
F1-score, %	38	32	52	16	59	51	56	00
Тренировочное время, с	87,20	10,17	10,61	0,14	04,21	9,41	49,44	53,43
Время прогнозирования, с	0,00	16,74	0,19	0,04	0,01	0,21	0,05	0,02
Общее время, с	87,26	26,91	10,80	0,18	04,22	9,62	49,49	53,45

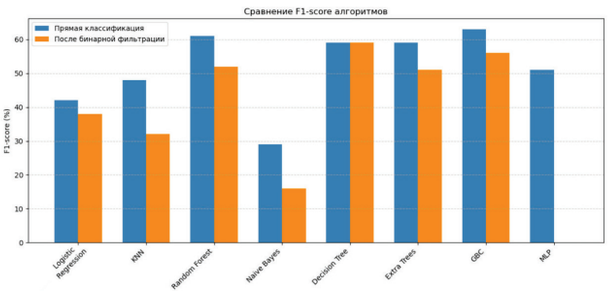


Рис. 2 Зависимость значений среднего *F1-score* от используемого алгоритма *ML*

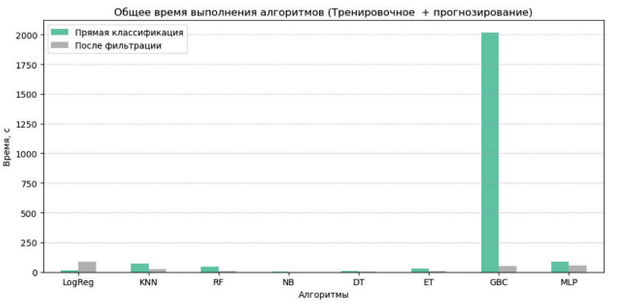


Рис. 3. Сравнение общего времени выполнения

бинарной. Матрицы ошибок, связанные с алгоритмами многоклассовой классификации, проиллюстрированы на рис. 2 и 3. Подробный анализ производительности в зависимости от типа атаки представлен в табл. 3–5. Рис. 4–6 дополняют и иллюстрируют эти результаты.

В случае прямой многоклассовой классификации алгоритмы *GBC*, *RF* и *ET* продемонстрировали превосходную производительность со средними *F1-score* около 60% (Таблица 5). Однако эти показатели сопровождаются весьма изменчивым временем тренировки, в то время как *ET* и *RF* остаются относительно быстрыми, *GBC* требует значительно большего времени тренировки, которое может достигать нескольких тысяч секунд. Этот последний пункт может стать серьезным ограничением в средах *IoT*.

Табл. 3. Точность распознания типа атак (*Precision*, %)

Прямая многоклассовая классификация [7], $K_{тест1}$ = 16 467									
	LR	KNN	RF	NB	DT	ET	GBC	*MLP*	Среднее по всем методам
Analysis	100	100	100	100	100	100	100	100	100
Backdoor	57	67	80	18	78	80	90	90	70
DoS	74	78	88	84	87	87	90	79	83,375
Exploits	43	29	39	45	37	39	43	38	39,125
Fuzzers	60	65	75	89	74	74	64	61	70,25
Generic	00	04	13	04	15	12	68	00	14,5
Normal	68	10	24	11	21	25	79	74	39
Reconnaissance	94	100	99	94	99	99	100	99	98
Shellcode	100	54	68	04	54	66	61	49	57
Worms	00	25	71	00	32	62	55	100	43,125
Многоклассовая классификация после бинарной фильтрации, $K_{тест2}$ = 9 067									
Analysis	50	09	80	00	100	21	100	00	45
Backdoor	00	04	62	00	56	11	69	00	25,25
DoS	41	35	42	27	45	44	46	00	35
Exploits	62	46	64	79	76	64	69	34	61,75
Fuzzers	64	39	72	56	67	76	81	00	56,875
Generic	99	99	100	61	99	100	100	00	82,25
Reconnaissance	73	74	93	16	93	90	94	00	66,625
Shellcode	00	36	72	06	68	65	67	01	39,375
Worms	00	00	33	03	57	40	30	00	20,375

Табл. 4. Доля найденных объектов классов от числа всех объектов класса (*Recall*, %)

Прямая многоклассовая классификация[7], $K_{тест1}$ = 16 467									
	LR	KNN	RF	NB	*DT*	*ET*	*GBC*	*MLP*	Среднее по всем методам
Analysis	100	100	100	100	100	100	100	100	100
Backdoor	65	56	77	12	77	75	76	70	63,5
DoS	71	73	89	14	86	89	85	85	74
Exploits	04	32	38	01	40	37	23	06	22,625
Fuzzers	84	67	79	16	72	79	87	91	73,125
Generic	00	07	14	64	16	12	17	00	16,25
Normal	08	15	21	23	21	21	17	14	17,5
Reconnaissance	96	98	98	87	98	98	98	97	96,25
Shellcode	01	19	61	99	53	49	56	27	45,625
Worms	00	00	16	16	28	16	38	06	15
Многоклассовая классификация после бинарной фильтрации, $K_{тест2}$ = 9 067									
Analysis	01	06	06	00	05	06	06	02	04
Backdoor	00	02	04	00	07	02	07	00	2,75
DoS	24	34	39	01	62	46	61	00	33,375
Exploits	77	65	75	11	68	73	75	01	55,625
Fuzzers	80	30	81	13	86	78	81	00	61,125
Generic	96	96	97	97	97	97	98	00	84,75
Reconnaissance	74	39	80	51	81	81	81	00	61,625
Shellcode	00	05	64	02	53	42	61	73	37,5
Worms	00	00	09	18	73	18	27	00	18,125

После внедрения бинарной фильтрации наблюдается общее снижение *F1-score* для большинства алгоритмов из-за удаления нормального контекста, который помогает некоторым алгоритмам в различении классов. Однако этот шаг также позволяет уменьшить размер выборки, что приводит к значительному уменьшению времени тренировки и прогнозирования для большинства алгоритмов. Таким образом, некоторые модели показывают лучший компромисс времени и производительности в этой конфигурации, что делает иерархическую классификацию привлекательной в рамках ограниченного развертывания.

Табл. 5. Гармоническое значение *Precision* и *Recall* (*F1-score*, %)

Прямая многоклассовая классификация [7], $K_{тест1}$ = 16 467									
	LR	*KNN*	*RF*	*NB*	*DT*	*ET*	*GBC*	*MLP*	Среднее по всем методам
Analysis	100	100	100	100	100	100	100	100	100
Backdoor	61	61	79	14	77	77	82	79	66,25
DoS	72	75	89	24	87	88	87	82	75,5
Exploits	07	31	38	01	38	38	30	11	24,25
Fuzzers	70	66	77	27	73	76	74	73	67
Generic	00	05	14	08	15	12	27	00	10,125
Normal	15	12	23	15	21	23	29	23	20,125
Reconnaissance	95	99	99	90	98	99	99	98	97,125
Shellcode	01	28	64	08	53	56	58	35	37,875
Worms	00	06	26	00	30	25	44	12	17,875
Многоклассовая классификация после бинарной фильтрации, $K_{тест2}$ = 9 067									
Analysis	01	07	11	00	10	09	11	00	6,125
Backdoor	00	03	07	00	12	04	12	00	4,75
DoS	31	35	41	01	52	45	52	00	32,125
Exploits	69	54	69	19	72	68	72	02	56,875
Fuzzers	71	34	76	21	76	77	81	00	54,5
Generic	98	98	99	75	98	98	99	00	83,125
Reconnaissance	73	51	86	24	86	85	87	00	61,5
Shellcode	00	09	68	03	59	51	64	02	32
Worms	00	00	14	05	64	25	29	00	17,125

Табл. 6. Атаки обнаружены и классифицированы по типу

Прямая многоклассовая классификация, $K_{тест1}$ = 16 467								
	LR	KNN	RF	NB	DT	ET	GBC	MLP
Analysis	-	131	101	1 418	8	98	23	11
Backdoor	-	11	98	1 645	12	99	27	-
DoS	894	795	713	32	1 078	690	933	812
Exploits	1 912	2 483	2 461	1 222	2 513	2 489	2 034	2 348
Fuzzers	1 320	1 209	1 220	228	1 006	1 231	1 591	1 392
Generic	4 204	3 683	3 718	2 185	3 739	3 713	3 731	3 739
Reconnaissance	695	676	638	296	592	652	609	670
Shellcode	-	-	75	1 613	77	52	65	51
Worms	-	-	1	386	-	1	19	2
Всего обнаружено атак	9025	9021	9025	9025	9025	9025	9032	9025
Многоклассовая классификация после бинарной фильтрации, $K_{тест2}$ = 9 067								
	LR	KNN	RF	NB	DT	ET	GBC	MLP
Analysis	-	-	-	-	-	1	-	83
Backdoor	278	-	-	-	-	-	3 002	167
DoS	492	6 084	1 972	-	3 111	1 622	2 492	-
Exploits	817	2 706	2 226	-	5 697	2 359	352	7 472
Fuzzers	449	-	1 093	-	-	1 061	203	-
Generic	3 685	-	3 667	6 783	-	3 685	-	-
Reconnaissance	1 223	-	-	-	-	161	3	115
Shellcode	1 944	-	-	-	-	1	-	974
Worms	-	-	-	-	73	-	2 948	165
Всего обнаружено атак	8 888	8 790	8 958	6 783	8 881	8 890	9 000	8 976
Символ «-» указывает на то, что атака не была обнаружена соответствующим алгоритмом. Если атака обнаружена, указывается количество выявленных случаев.								

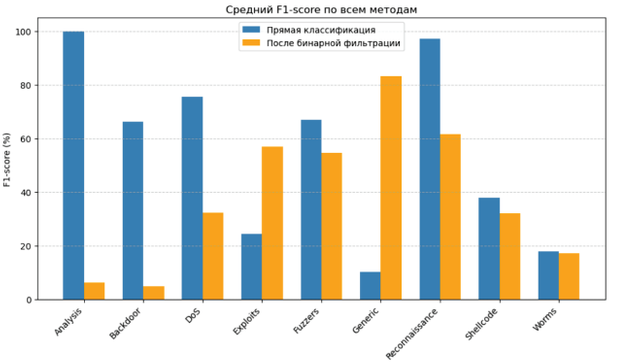
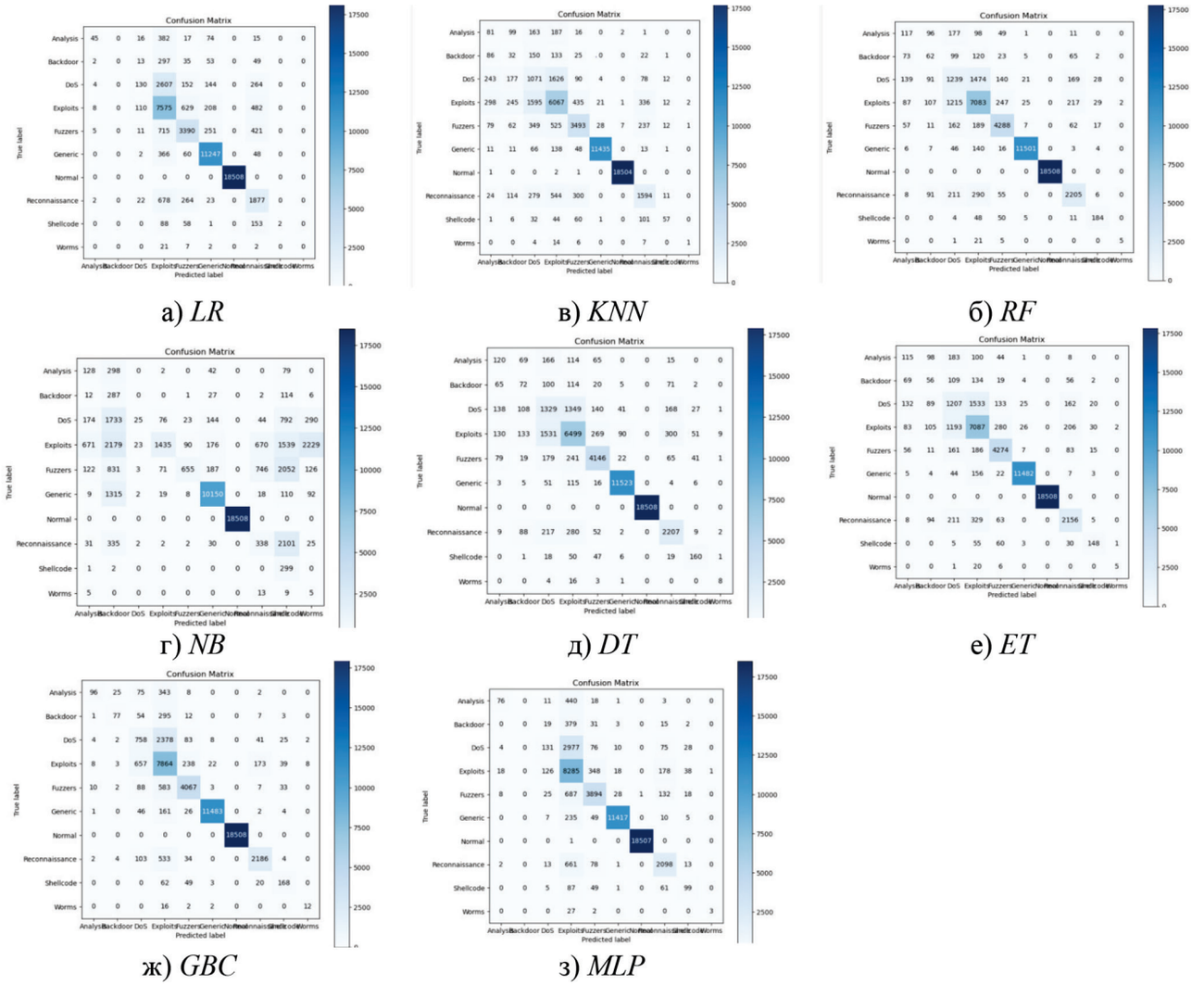


Рис. 4. Зависимость средних (по всем алгоритмам *ML*) значений *F1-score* от типа атаки

Подробный анализ по типу атаки, представленный на Рисунке 4, показывает, что для некоторых классов, таких как «*Analysis*» или «*Backdoor*», которые хорошо обнаруживаются при прямой классификации, после фильтрации возникает значительное снижение производительности. Напротив, для класса «*Generic*» наблюдается заметное улучшение, что говорит о том, что снижение шума посредством бинарной фильтрации облегчает обнаружение определенных типов атак.

Подводя итог, можно сказать, что прямая классификация обеспечивает лучшее общее покрытие атак с выгодным компромиссом *F1-score*, но может быть ограничена значительным временем тренировки, особенно для некоторых алгоритмов, таких как *GBC*. Иерархическая классификация сокращает это время, сохраняя при этом хорошую производительность на основных классах, хотя иногда и за счет более малочисленных классов.

GBC достигает наилучших общих результатов *F1-score* (63% прямых, 56% отфильтрованных) и демонстрирует замечательную способность обнаруживать даже редкие классы. Однако это достигается ценой очень большого времени тренировки (более 2 000 секунд в прямой многоклассовой классификации). Для сравнения, *ET* является хорошим компромиссом: производительность близка к *GBC* (*F1-score* ≈ 59%), но с гораздо меньшим временем выполнения (30 секунд в реальном времени, 9 секунд после фильтрации).

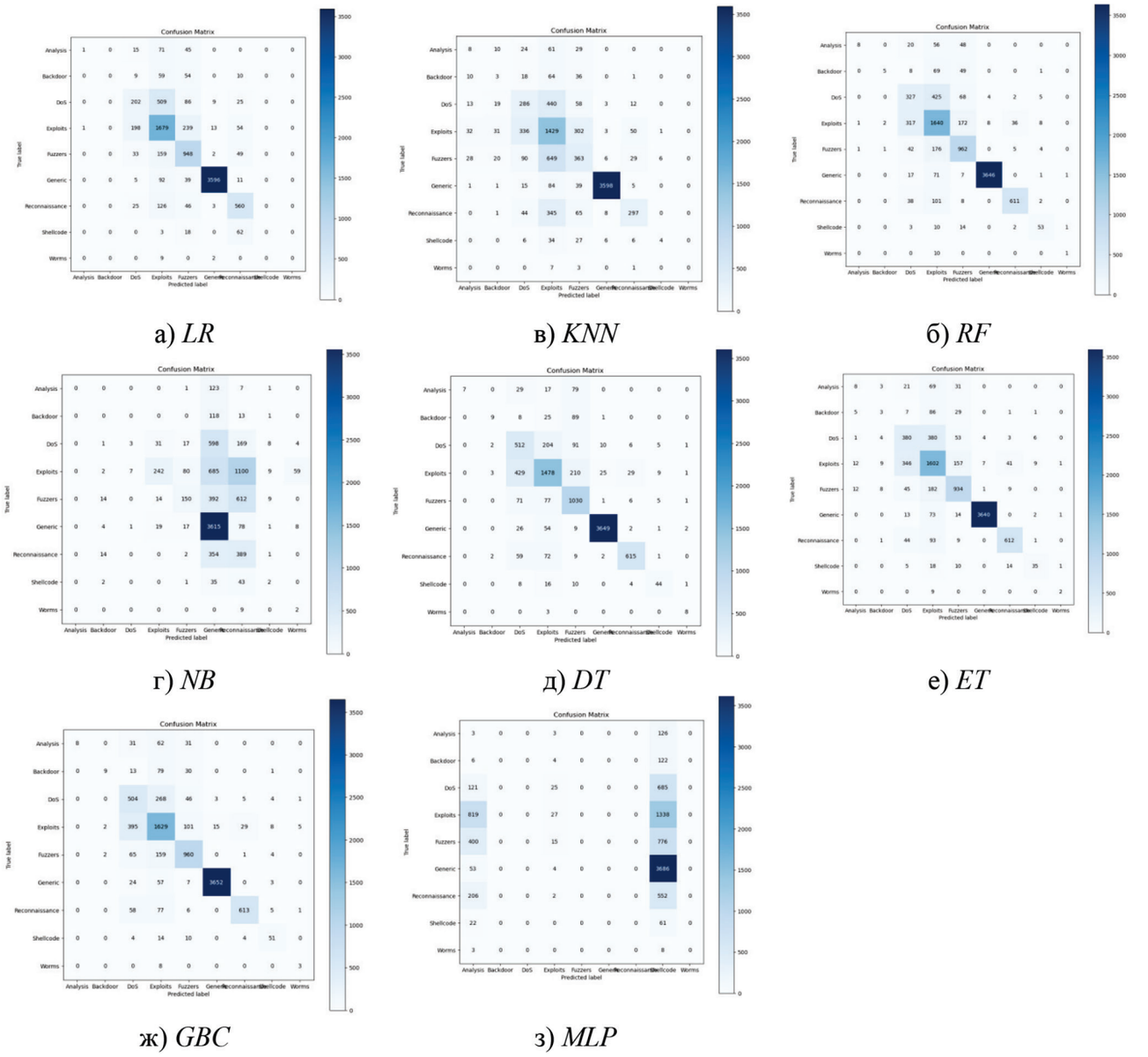


а) *LR* в) *KNN* б) *RF*

г) *NB* д) *DT* е) *ET*

ж) *GBC* з) *MLP*

Рис. 5. Матрицы ошибок для рассмотренных алгоритмов (Прямая многоклассовая классификация) [7]



а) *LR* в) *KNN* б) *RF*

г) *NB* д) *DT* е) *ET*

ж) *GBC* з) *MLP*

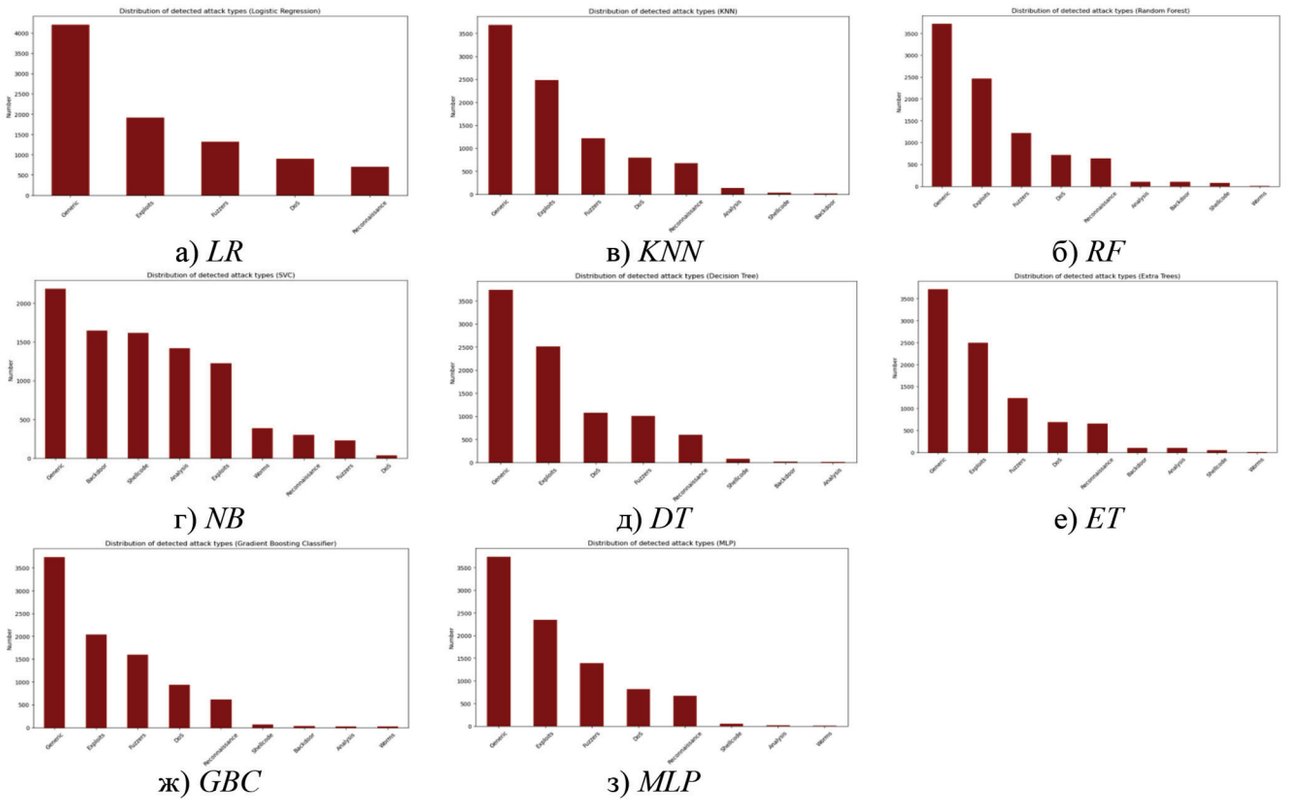
Рис. 6. Матрицы ошибок для рассмотренных алгоритмов (Многоклассовая классификация после бинарной фильтрации)

Наконец, алгоритмы *ET* и *RF* представляются рациональными решениями для использования в *IoT*, предлагая баланс между точностью, надежностью и скоростью выполнения как при прямой классификации, так и после бинарной фильтрации.

Табл. 6 показывает количество случаев, правильно обнаруженных и классифицированных по типу атаки. Она позволяет более подробно проанализировать способность алгоритмов распознавать различные типы вторжений. Рисунки 7–10 дополняют и иллюстрируют эти результаты.

В дополнение к количественному анализу *F1-score*, важно изучить способность различных алгоритмов эффективно обнаруживать и классифицировать различные типы атак, как показано в табл. 6 и на рис. 7–10. В табл. 6 суммировано количество примеров атак, правильно идентифицированных и отнесенных к своему типу каждым алгоритмом, с разграничением прямой многоклассовой классификации и классификации, выполненной после фазы бинарной фильтрации.

В прямой настройке многоклассовой классификации алгоритмы *RF*, *ET* и *GBC* обнаруживают большое количество атак в большинстве классов, подтверждая их надежность, уже отмеченную их общей производительностью *F1-score*. Например, *RF* и *ET* обнаруживают более 2 000 случаев в типах *Exploits* и *Fuzzers*, в то время как *GBC* также демонстрирует превосходное покрытие для таких классов, как *Generic*, с более чем 3 700 правильно классифицированными примерами. Напротив, такие модели, как *KNN* или *NB*, показывают более неравномерное обнаружение с заметными

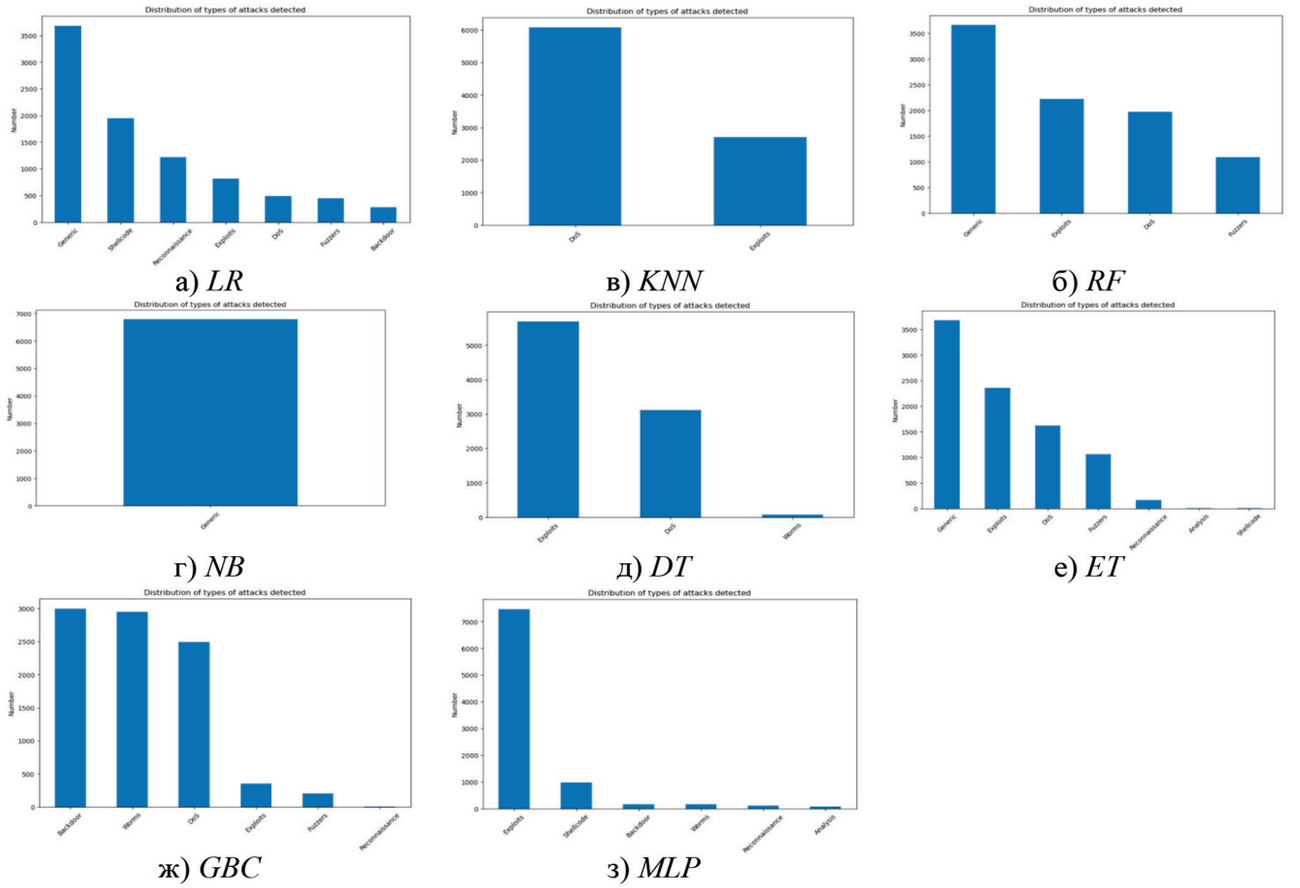


а) *LR* в) *KNN* б) *RF*

г) *NB* д) *DT* е) *ET*

ж) *GBC* з) *MLP*

Рис. 7. Распределение типов атак, обнаруженных алгоритмами (прямая многоклассовая классификация)



а) *LR* в) *KNN* б) *RF*

г) *NB* д) *DT* е) *ET*

ж) *GBC* з) *MLP*

Рис. 8. Распределение типов атак, обнаруженных алгоритмами (многоклассовая классификация после бинарной фильтрации)

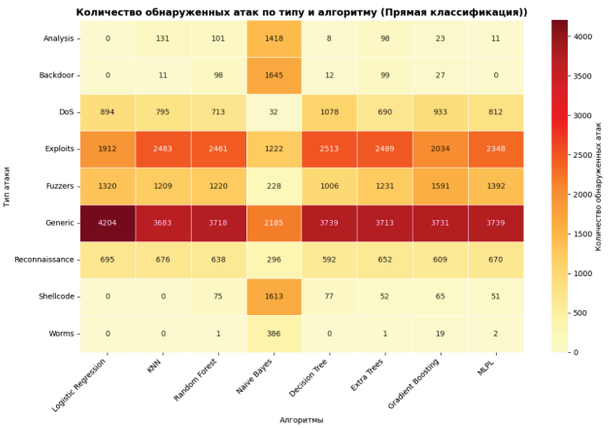


Рис. 9. Количество обнаруженных атак по типу и алгоритму (Прямая многоклассовая классификация)

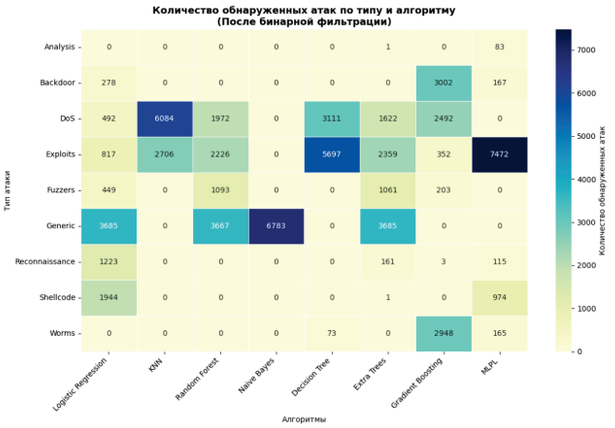


Рис. 10. Количество обнаруженных атак по типу и алгоритму (Многоклассовая классификация после бинарной фильтрации)

сбоями в некоторых редких или специфических классах, таких как *Shellcode* и *Worms*, что отражает их ограничения в распознавании менее представленных типов атак.

После применения бинарной фильтрации динамика обнаружения существенно меняется. Общее количество обнаруженных атак немного уменьшается в результате сокращения базы примеров для классификации, но некоторые модели, такие как *GBC* и *RF*, поддерживают высокий уровень обнаружения по ключевым классам. Например, *GBC* теперь обнаруживает более 3 000 случаев *Backdoor* и почти 2 500 случаев *DoS*, показывая, что бинарная фильтрация не ставит под угрозу его способность идентифицировать эти критические атаки. *ET* также поддерживает хорошее покрытие по типам *Generic*, *Exploits* и *Fuzzers*. Однако некоторые алгоритмы, в частности, *LR* и *NB*, показывают, что количество обнаруженных атак резко падает, что отражает их большую зависимость от глобального контекста, включая обычный трафик. Аналогично, *KNN* теряет способность обнаруживать несколько классов после фильтрации, что иллюстрирует его ограничения в этой настройке.

Подводя итог, анализ Таблицы 6 подтверждает, что модели *GBC*, *RF* и *ET* сочетают в себе как хорошую производительность *F1-score*, так и эффективную способность обнаруживать широкий спектр атак, даже после уменьшения проблемы с помощью бинарной фильтрации. Это наблюдение подтверждает их роль в качестве предпочтительных решений для обнаружения вторжений в *IoT*, где разнообразие атак и вычислительные ограничения являются основными проблемами.

После бинарной фильтрации результаты многоклассовой классификации остаются посредственными. Однако в обоих подходах – прямой многоклассовой классификации и классификации после фильтрации – алгоритм *GBC* обеспечивает наилучшую производительность.

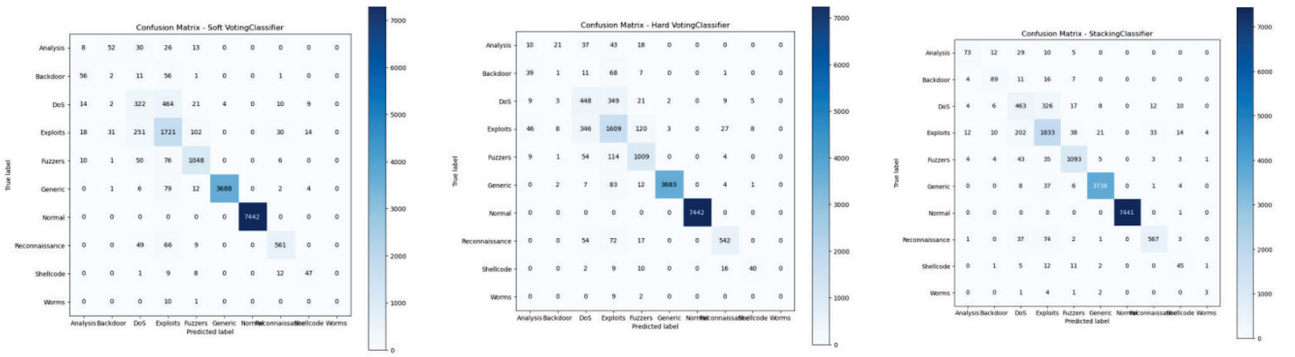
3. Анализ результатов ансамблевых подходов

В табл. 7 представлено подробное сравнение трех ансамблевых моделей (мягкого голосования (*Soft Voting Classifier (SVC)*), жесткого голосования (*Hard Voting Classifier (HVC)* и *Stacking Classifier (SC)*) и высокопроизводительного индивидуального алгоритма *GBC*, применяемого для многоклассовой классификации. Рис. 11 и 12 иллюстрируют эти результаты.

Табл. 7 показывает, что *SC* обеспечивает наилучшую общую производительность с точностью 75,67%, полнотой 72,48% и *F1-score 73*,87%. Эта эффективность обусловлена его способностью оптимально комбинировать результаты нескольких классификаторов. Однако эта высокая производительность сопровождается значительными вычислительными затратами: время обучения составляет 8 208,05 с, а общее время выполнения – 8 231,01 с. Модели *SVC* и *HVC*, напротив, показывают более скромные результаты (*F1-score* ≈ 55,7%), но выигрывают от более короткого времени

Табл. 7. Сравнение значений метрик рассмотренных алгоритмов многоклассовой классификации, $K_{тест1}$ = 16 467

	GBC	Soft Voting	Hard Voting	Stacking
Precision, %	75	56,30	57,34	75,67
Recall, %	60	55,34	54,87	72,48
F1-score, %	63	55,72	55,78	73,87
Время обучения, с	2 016,89	1 288,25	1 007,75	8 208,05
Время прогнозирования, с	1,08	20,95	20,22	22,97
Общее время, с	2 017,97	1 309,20	1 027,98	8 231,01



а) *SVC* в) *HVC* б) *SC*

Рис. 12. Матрицы ошибок алгоритмов

обучения и прогнозирования, особенно для *HVC*. В свою очередь, *GBC*, хотя и является отдельной моделью, выделяется своей хорошей точностью (75%) и *F1-score*, более высокой, чем у подходов *Voting* (63%), демонстрируя при этом наименьшее время прогнозирования (1,08 с).

Рис. 11 и 12 завершают этот анализ, наглядно иллюстрируя преимущество модели *SC* с точки зрения качества классификации.

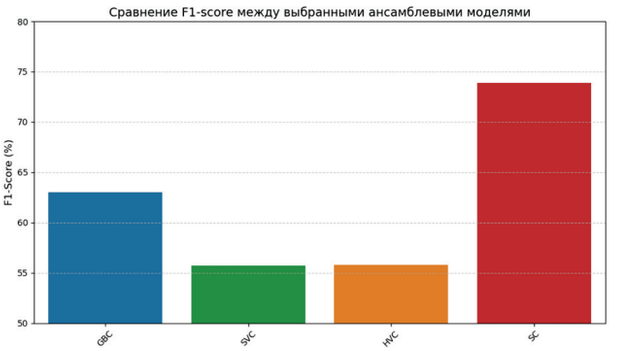


Рис. 11. Сравнение F1-score между алгоритмами

Подводя итог, можно сказать, что, хотя *SC* является наиболее эффективным, его применение следует оценивать с учетом временных ограничений и доступных ресурсов. Одновременно *GBC* представляет собой интересный компромисс между эффективностью и скоростью.

Заключение

Данной исследование показывает, что введение предварительной бинарной фильтрации является актуальной стратегией для улучшения специализации алгоритмов на определенных атаках, одновременно снижая вычислительные затраты. Однако эта фильтрация может поставить под угрозу обнаружение редких классов. Таким образом, необходимо найти компромисс между производительностью, покрытием и эффективностью по времени. *GBC* остается наиболее эффективным при редких атаках, но его ресурсоемкость делает его плохо подходящим для встраиваемых систем. *ET* является отличным компромиссом между точностью и скоростью, в то время как *RF* надежен и стабилен в обеих конфигурациях. Эти три алгоритма оказываются наиболее подходящими для требований систем обнаружения вторжений в средах *IoT* из-за их способности эффективно обнаруживать широкий спектр атак, включая самые редкие.

Оценка трех ансамблевых методов (*HVC*, *SVC* и *SC*), сравнение их эффективности с лучшим индивидуальным классификатором (*GBC*) показывают, что *Stacking* превосходит все модели, включая *GBC*, демонстрируя заметное улучшение общего результата *F1-score*, обеспечивая при этом лучшее покрытие сложных классов. Таким образом, научная новизна выполненного исследования заключается в применении иерархического подхода к определению типа вторжения в *IoT*, который ранее для этого не использовался, а его практическая ценность заключается в том, что иерархический подход прокладывает путь для создания более надежных и экономически эффективных архитектур *IDS*, способных адаптироваться к конкретным проблемам кибербезопасности в *IoT*.

Благодарности

Работа выполнена за счет бюджетного финансирования в рамках государственного задания от 20.03.2025 № 103-00001-25-02.

Список литературы

1. Sadhu P.K., Yanambaka V.P., Abdelgawad A. Internet of things: Security and solutions survey // Sensors. 2022. Vol. 22. No. 19. P. 7433. URL: https://www.mdpi.com/1424-8220/22 /19/7433 (дата обращения: 06.01.2026). DOI: 10.3390/s22197433 EDN: ZFJPEH

2. Janabi A.H., Kanakis T., Johnson M. Survey: Intrusion detection system in software-defined networking // IEEE Access. 2024. URL: https://ieeexplore.ieee.org/abstract/document/10746482/ (дата обращения: 06.01.2026). DOI: 10.1109/access.2024.3493384 EDN: JQSXDN

3. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In : 2015 military communications and information systems conference (MilCIS). IEEE, 2015. Pp. 1-6. DOI: 10.1109/MilCIS.2015.7348942

4. Lin W.C., Ke S.W., Tsai C.F. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors // Knowledge-based systems. 2015. Vol. 78. Pp. 13-21. DOI: 10.1109/MilCIS.2015.7348942

5. Нианг П.М. Анализ наборов данных для исследования уязвимостей компьютерных сетей. // Труды III Международной научно-практической конференции «Интеллектуальные транспортные системы» (30 мая 2024 г.). М.: Издательство Перо, 2024. С. 699-709. DOI: 10.30932/9785002446094-2024-699-709 EDN: TTNUUB

6. Нианг П.М., Сидоренко В.Г. Выбор алгоритма машинного обучения для обнаружения вторжений в IoT. // Надежность. 2024. Т. 24. № 3. С. 44-51. DOI: 10.21683/1729-2646-2024-24-3-44-51 EDN: HYJCRA

7. Malik N.P., Sidorenko V.G. Application of Multiclassification for Detecting Intrusions in IoT and Their Type Recognizing. In: 2024 International Conference “Quality Management, Transport and Information Security, Information Technologies” (QM&TIS&IT). IEEE, 2024. Pp. 78-83. DOI: 10.1109/QMTISIT63393.2024.10762926

8. Кулагин М.А., Логинова Л.Н., Сидоренко В.Г. и др. Формирование навыков использования алгоритмов машинного обучения у специалистов по информационной безопасности // Информатизация образования и науки. 2025. № 1(65). С. 56-65. URL: https://journal.ficto.ru/archive.html#journal_65 (дата обращения: 06.01.2026).

9. Ganaie M.A., Hu M., Malik A.K. et al. Ensemble deep learning: A review // Engineering Applications of Artificial Intelligence. 2022. Vol. 115. P. 105151. DOI: 10.1016/j.engappai.2022.105151 EDN: OZQDBQ

10. Yang Y., Lv H., Chen N. A survey on ensemble learning under the era of deep learning. // Artificial Intelligence Review. 2023. Vol. 56. No 6. Pp. 5545-5589. DOI: 10.1007/s10462-022-10283-5 EDN: UNDPUQ

11. Mienye I.D., Sun Y. A survey of ensemble learning: Concepts, algorithms, applications, and prospects // IEEE Access. 2022. Vol. 10. Pp. 99129-99149. DOI: 10.1109/ACCESS.2022.3207287 EDN: YZJLXM

12. Rezk S.S., Selim K.S. Metaheuristic-based ensemble learning: an extensive review of methods and applications // Neural Computing and Applications. 2024. Vol. 36. No 29. Pp. 17931-17959. DOI: 10.1007/s00521-024-10203-4 EDN: IQOHYX

13. Zhang Y., Liu J., Shen W. A review of ensemble learning algorithms used in remote sensing applications // Applied Sciences. 2022. Vol. 12. No 17. P. 8654. DOI: 10.3390/app12178654 EDN: GVPHOT

14. Ahuja R., Sharma S.C. Stacking and voting ensemble methods fusion to evaluate instructor performance in higher education // International Journal of Information Technology. 2021. Vol. 13. No 5. Pp. 1721-1731. DOI: 10.1007/s41870-021-00729-4 EDN: GLMTHI

References

1. Sadhu P.K., Yanambaka V.P., Abdelgawad A. Internet of things: Security and solutions survey. *Sensors* 2022;22(19):7433. (accessed 06.01.2026). Available at: https://www.mdpi.com/1424-8220/22 /19/7433. DOI: 10.3390/s22197433 EDN: ZFJPEH.

2. Janabi A.H., Kanakis T., Johnson M. Survey: Intrusion detection system in software-defined networking. *IEEE Access* 2024. (accessed: 06.01.2026). Available at: https://ieeexplore.ieee.org/abstract/document/10746482/. DOI: 10.1109/access.2024.3493384 EDN: JQSXDN.

3. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 military communications and information systems conference (MilCIS). IEEE; 2015. Pp. 1-6. DOI: 10.1109/MilCIS.2015.7348942.

4. Lin W.C., Ke S.W., Tsai C.F. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based systems* 2015;78:13-21. DOI: 10.1109/MilCIS.2015.7348942.

5. Niang P.M. Analysis of data sets for the study of computer network vulnerabilities. In: Proceedings of the III International Science and Practice Conference Intelligent Transport Systems (May 30, 2024). Moscow: Pero Publishing; 2024. Pp. 699-709. DOI: 10.30932/9785002446094-2024-699-709 EDN: TTNUUB. (in Russ.)

6. Niang P.M., Sidorenko V.G. Choosing the machine learning algorithm for detecting intrusions into IoT. *Dependability* 2024;24(3):44-51. (In Russ.) DOI: 10.21683/1729-2646-2024-24-3-44-51 EDN: HYJCRA.

7. Malik N.P., Sidorenko V.G. Application of Multiclassification for Detecting Intrusions in IoT and Their Type Recognizing. In: 2024 International Conference Quality Management, Transport and Information Security, Information Technologies (QM&TIS&IT). IEEE; 2024. Pp. 78-83. DOI: 10.1109/QMTISIT63393.2024.10762926.

8. Kulagin M.A., Loginova L.N., Sidorenko V.G. et al. Formation of skills in using machine learning algorithms for information security specialists. *Informatizatsiya obrazovaniya i nauki* 2025;1(65):56-65. (accessed: 06.01.2026). Available at: https://journal .ficto.ru/archive.html#journal_65. (in Russ.)

9. Ganaie M.A., Hu M., Malik A.K. et al. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence* 2022;115:105151. DOI: 10.1016/j.engappai.2022.105151 EDN: OZQDBQ.

10. Yang Y., Lv H., Chen N. A survey on ensemble learning under the era of deep learning. *Artificial Intelligence Review* 2023;56(6):5545-5589. (in Russ.) DOI: 10.1007/s10462-022-10283-5 EDN: UNDPUQ.

11. Mienye I.D., Sun Y. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access* 2022;10:99129-99149. DOI: 10.1109/ACCESS.2022.3207287 EDN: YZJLXM.

12. Rezk S.S., Selim K.S. Metaheuristic-based ensemble learning: an extensive review of methods and applications. *Neural Computing and Applications* 2024;36(29):17931-17959. DOI: 10.1007/s00521-024-10203-4 EDN: IQOHYX.

13. Zhang Y., Liu J., Shen W. A review of ensemble learning algorithms used in remote sensing applications. *Applied Sciences* 2022;12(17):8654. DOI: 10.3390/app12178654 EDN: GVPHOT.

14. Ahuja R., Sharma S.C. Stacking and voting ensemble methods fusion to evaluate instructor performance in higher education. *International Journal of Information Technology* 2021;13(5):1721-1731. DOI: 10.1007/s41870-021-00729-4 EDN: GLMTHI.

Сведения об авторах

Папа Малик Нианг – аспирант РУТ (МИИТ), ул. Образцова, д.9, стр.9, Москва, Российская Федерация, 127994, e-mail: malickdiarra30@gmail.com

Валентина Геннадьевна Сидоренко – доктор технических наук; профессор; профессор кафедры «Управление и защита информации» РУТ (МИИТ), ул. Образцова, д.9, стр.9, Москва, Российская Федерация, 127994, e-mail: valenfalk@mail.ru

About the authors

Papa Malik Niange, Postgraduate student, Russian University of Transport (RUT MIIT), ul. Obraztsova 9, building 9, Moscow, Russian Federation, 127994, e-mail: malickdiarra30@gmail.com

Valentina G. Sidorenko, Doctor of Sciences (Engineering); Professor; Professor, Department of Control and Information Security, Russian University of Transport (RUT MIIT), ul. Obraztsova 9, building 9, Moscow, Russian Federation, 127994, e-mail: valenfalk@mail.ru

Вклад авторов в статью

П.М. Нианг. Проведение обзора литературы по теме исследования. Разработка методологии и архитектуры исследования, включая прямую многоклассовую классификацию и иерархический подход с бинарной фильтрацией. Реализация и параметризация алгоритмов машинного обучения и ансамблевых методов (*Voting, Stacking*). Проведение вычислительных экспериментов на наборе данных UNSW-NB15. Предварительная обработка и визуализация результатов.

В.Г. Сидоренко. Анализ полученных результатов.

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.