Моделирование и анализ данных
2026, *16*(1), 7—26.
https://doi.org/10.17759/mda.2026160101
ISSN: 2219-3758 (печатный)
ISSN: 2311-9454 (online)

Modelling and Data Analysis
2026, *16*(1), 7—26.
https://doi.org/10.17759/mda.2026160101
ISSN: 2219-3758 (print)
ISSN: 2311-9454 (online)

АНАЛИЗ ДАННЫХ | DATA ANALYSIS

Научная статья | Original paper

УДК 004.85

Повышение эффективности обнаружения угроз: методы классификации вредоносных программ на основе машинного обучения

А.А. Абедалхуссайн ✉, Е.В. Ляпунцова
Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет “МИСИС”» (НИТУ МИСИС), Москва, Россия
✉ m2000009@edu.misis.ru

Резюме

Сигнатурные методы и простые эвристики все хуже справляются с изменчивостью вредоносных программ, а полная динамическая проверка каждого файла в «песочнице» (изолированной среде запуска) слишком дорога по времени и ресурсам, поэтому на практике требуется автоматический отбор файлов с жестким контролем ложных тревог. Целью исследования стало повышение эффективности обнаружения угроз за счет построения переносимой модели классификации, которая сохраняет высокую полноту выявления вредоносных объектов при заранее заданном ограничении на ложные срабатывания и при управляемой нагрузке на песочницу. Гипотеза состояла в том, что порог принятия решения следует выбирать не «по тесту» и не по средним метрикам, а по бюджету ошибок: использовать внешнескладочные (out-of-fold, то есть полученные на объектах, не участвовавших в обучении конкретного подмножества модели) предсказания на безвредных файлах и задавать порог так, чтобы число ложных срабатываний не превышало K, после чего реализовать трехзонную политику «блокировать/отправлять на проверку/пропускать». Эксперименты выполнены на наборе UCI «Malware static and dynamic features VxHeaven and Virus Total» (6248 файлов, 1084 признака, далее после удаления константных признаков — 244), причем оценка проводилась не только в случайном разбиении, но и в двух сценариях

переносимости «обучение на одном источнике — тест на другом» (VxHeaven — VirusTotal), имитирующих смену домена данных. В качестве базовых моделей рассматривались линейные классификаторы и ансамбли деревьев; дополнительно проверялась калибровка оценок уверенности (приведение «сырых» оценок модели к более корректным вероятностям) для устойчивого порогования. Для доказательности контроль ложных срабатываний дополнялся точными биномиальными доверительными интервалами, что особенно важно при малом числе безвредных объектов в тесте. Основной результат получен для политики Stage12 (порог по K ложным срабатываниям на OOF-предсказаниях): в сценарии VxHeaven — VirusTotal достигнута полнота 0,8227 при доле файлов, уходящих в песочницу, 0,2092 и нулевой наблюдаемой доле ложных срабатываний; по сравнению с базовой «серой зоной» полнота выросла на +0,2816, а нагрузка на песочницу снизилась в 2,29 раза. В сценарии VirusTotal — VxHeaven полнота составила 0,9670 при доле файлов в песочнице 0,0735 и наблюдаемой доле ложных срабатываний 0,0084; по сравнению с «серой зоной» полнота выросла на +0,1234, а нагрузка на песочницу снизилась в 2,61 раза при сохранении того же уровня ложных срабатываний. Выводы фиксируют, что предложенный способ выбора порога по бюджету K на OOF-предсказаниях обеспечивает практично управляемый режим работы детектора: повышает полноту на переносимых сценариях и одновременно уменьшает объем ручной/динамической проверки, сохраняя контроль ложных тревог. Научная новизна заключается в обоснованной связке «переносимость + бюджет ложных срабатываний + трехзонная политика решений», где порог задается не оптимизацией усредненной метрики, а формализованным ограничением на ошибки, и подтверждается сравнением на межисточниковых сценариях и доверительными интервалами для FPR (доли ложных срабатываний).

Ключевые слова: вредоносные программы, машинное обучение, обнаружение угроз, переносимость моделей, контроль ложных срабатываний, выбор порога, песочница, доверительные интервалы, классификация с отказом

Для цитирования: Абедалхуссайн, А.А., Ляпунцова, Е.В. (2026) Повышение эффективности обнаружения угроз: методы классификации вредоносных программ на основе машинного обучения. *Моделирование и анализ данных, 16*(1), 7—26. https://doi.org/10.17759/mda.2026160101

Increasing the effectiveness of threat detection: machine-learning — based methods for malware classification

A.A. Abedlhussain ✉, E.V. Lyapuntsova
National University of Science and Technology MISIS (NUST MISIS)
✉ m2000009@edu.misis.ru

Abstract

Signature-based detection and lightweight heuristics increasingly struggle with rapidly evolving malware, while running every file in a sandbox is too costly; therefore, practical malware triage requires automated decisions under a strict false-alarm budget. This study aims to improve threat detection efficiency by developing a transferable machine-learning classifier that preserves high malware recall while explicitly controlling false positives and keeping the sandbox workload manageable. We hypothesize that decision thresholds should be selected not by optimizing an average metric on a held-out test split, but via an explicit error budget: using out-of-fold predictions on benign files to set a blocking threshold such that the number of false positives does not exceed K, and then deploying a three-zone policy («block/send for review/allow»). Experiments were conducted on the UCI dataset «Malware static and dynamic features VxHeaven and Virus Total» (6,248 files; 1,084 features; reduced to 244 after removing constant features), with evaluation performed not only under a standard random split but also under two cross-source transfer scenarios (train on VxHeaven, test on VirusTotal, and vice versa), which emulate real-world domain shifts. We compared linear models and tree-based ensembles and additionally examined score calibration (mapping raw model scores to better-behaved probabilities) to support robust thresholding. To provide a conservative and evidence-based assessment of false positives under small benign test samples, we reported exact binomial confidence intervals for the false-positive rate. The main gain was achieved by the proposed Stage12 policy (K-based thresholding from out-of-fold benign predictions): in the VxHeaven — VirusTotal scenario, recall reached 0.8227 with a sandbox review rate of 0.2092 and zero observed false positives; compared to the baseline gray-zone policy, recall increased by +0.2816 while the review load decreased by 2.29×. In the VirusTotal — VxHeaven scenario, recall reached 0.9670 with a review rate of 0.0735 and an observed false-positive rate of 0.0084; relative to the gray-zone baseline, recall increased by +0.1234 and the review load decreased by 2.61× at the same observed false-positive level. These results demonstrate that K-budgeted, out-of-fold threshold selection enables an operationally controlled detection regime under domain shift: it improves recall and reduces the need for expensive sandboxing while maintaining a defensible false-alarm control. The scientific novelty is an evidence-backed integration of transfer evaluation, explicit false-positive budgeting, and a three-zone decision policy, where the operating point is determined by a formal error constraint rather than by optimizing a single average score.

Keywords: malware detection, machine learning, threat classification, transferability, false-positive control, threshold selection, sandbox triage, confidence intervals, learning with rejection

For citation: Abedlhussain, A.A., Lyapuntsova, E.V. (2026). Increasing the effectiveness of threat detection: machine-learning–based methods for malware classification. *Modelling and Data Analysis, 16*(1), 7—26 (In Russ.). https://doi.org/10.17759/mda.2026160101

Введение

Поток файлов, попадающих в контуры корпоративной защиты (почта, веб-шлюзы, рабочие станции, репозитории), неизбежно включает долю неизвестных образцов, для которых сигнатурные правила и репутационные списки оказываются недостаточными. Ошибка первого рода в подобных задачах воспринимается не как «обычная неточность модели», а как прямой простой бизнеса: ложная блокировка легитимного файла ломает цепочки поставки, обновления и рабочие процессы. Нагрузка на «песочницу» (изолированную среду анализа) и очередь ручной верификации при этом растет быстрее, чем возможности аналитиков, поэтому практическая ценность детектора определяется не только полнотой обнаружения, но и управляемостью ложных срабатываний.

Поведенческие признаки, извлекаемые при запуске файла в изолированной среде, дают модели дополнительную опору там, где статические сигнатуры обходятся упаковщиками и обфускацией. Сложность переносимости таких моделей повышается из-за неоднородности источников данных и сценариев сбора: распределения признаков смещаются между наборами, а статистика «нормы» меняется в зависимости от парка программ и политик эксплуатации. Связанные проблемы — дефицит репрезентативных размеченных выборок, необходимость воспроизводимого контроля качества данных и риск «переобучения на источник» — в последние годы обсуждаются как отдельное направление исследований, включая попытки синтетического расширения данных для задач обнаружения вредоносного ПО (Стародубов, Боршевников, Селин, 2025).

Систематические обзоры по обнаружению вредоносных программ на основе методов искусственного интеллекта показывают устойчивый тренд к признаковым моделям, сочетающим интерпретируемые индикаторы (например, счетчики действий и статистики секций) и более сложные агрегаты поведения, однако подчеркивают уязвимость к смещению домена и различиям протоколов оценки. Наряду с ростом качества по метрикам ранжирования (ROC-AUC, PR-AUC) аналитики отмечают «операционную пропасть»: превосходство по кривым не гарантирует приемлемого режима эксплуатации, если модель не контролирует долю ложных блокировок и создает избыточный поток на проверку (Gaber, Ahmed, Janicke, 2024; Kan et al., 2024).

Контроль ложноположительных срабатываний удобно формализовать через ограничение на долю ошибочно заблокированных доброкачественных файлов (false positive rate, FPR), связывая требования безопасности с требованиями непрерывности процессов. Отказ от бинарного решения в пользу режима «воздержаться и отправить на проверку» давно рассматривается как рациональная стратегия в задачах с высокой ценой ошибки, поскольку позволяет переводить неоднозначные случаи в управляемую очередь. Современные обзоры по «классификации с отказом» (reject option) описывают математическую постановку, способы выбора порогов

и критерии оптимальности, прямо соответствующие практикам песочницы и ручной обработки (Hendrickx et al., 2024).

Предлагаемая в статье постановка рассматривает детектор как элемент управленческой политики, где решение «блокировать / пропустить / направить на проверку» задается двумя порогами по скору модели. Нижний порог отделяет безопасные (по мнению модели) файлы от «серой зоны», верхний порог выделяет уверенно вредоносные образцы, а промежуток формирует очередь песочницы. Научная новизна работы связывается с процедурой выбора порога блокировки по обучающим данным доброкачественного класса в режиме out-of-fold (вне-фолдовом), что позволяет интерпретировать настройку как бюджет на число ложных блокировок (K) и сопоставлять решения между источниками данных в одинаковых операционных ограничениях (FPR-бюджет и допустимая доля проверок) без «подгонки под тест».

Практический смысл исследования заключается в повышении эффективности обнаружения угроз при заданном уровне риска ложной блокировки и ограниченной пропускной способности песочницы. Для достижения цели решаются следующие задачи:

1. Формируется согласованный признаковый набор и протокол разделения данных, моделирующий перенос между источниками;
2. Сравниваются базовые алгоритмы классификации и режимы пороговой настройки, ориентированные на ограничение FPR;
3. Строится и оценивается трехзонная политика принятия решений, где «серая зона» минимизирует нагрузку на проверку при сохранении требуемой полноты обнаружения.

Материалы и методы

Эксперименты выполнены на табличном наборе статических и динамических признаков исполнимых файлов, где каждый объект описывается вектором числовых характеристик, а целевая переменная принимает значения 0 (benign, «безвредный») и 1 (malware, «вредоносный») (Malware static and dynamic features…, 2019). Источники данных отражают реалистичную неоднородность «полевых» потоков: вредоносные образцы собраны из различных коллекций (в частности, семейств, попадающих в публичные хранилища), а «безвредный» класс сформирован отдельно, что задает типичную для прикладной кибербезопасности задачу переносимости между доменами (domain shift) — изменением распределения признаков при неизменной семантике метки (Malware static and dynamic features…, 2019; Botacin, Gomes, 2024; Kan et al., 2024).

Матрица признаков после унификации и отбрасывания константных столбцов имеет размер (6248, 244), что соответствует 6248 файлам и 244 информативным признакам, одинаково заданным для всех сценариев разбиения. Таблица 1 фиксирует состав набора данных и итоговую размерность пространства признаков, используемую во всех последующих экспериментах.

Таблица 1 / Table 1

Сводка набора данных и пространства признаков

Dataset and feature space summary

Показатель	Значение
Объектов, всего (n)	6248
Классы	0 (benign), 1 (malware)
Признаков после очистки	244
Формат признаков	числовые табличные признаки (статические/динамические)

Сопоставление методов проводилось в трех сценариях, различающихся степенью «разрыва» между обучением и тестированием. Сценарий A моделирует квазистандартную оценку при случайном разбиении, а сценарии B и C реализуют проверку переносимости при смене домена: обучение на одном источнике вредоносных образцов и тестирование на другом. Таблица 2 задает логику разбиений и контрольные объемы тестовых подмножеств, на которых рассчитывались метрики.

Таблица 2 / Table 2

Сценарии разбиения для оценки переносимости

Splits used to evaluate transferability

Сценарий	Идея разбиения	Роль в статье	Размер теста
A_random_by_source	случайное разбиение при контроле источников	базовая оценка качества	1250
B_train_vx_test_vt	обучение на домене *vx*, тест на *vt*	переносимость 1	3074
C_train_vt_test_vx	обучение на домене *vt*, тест на *vx*	переносимость 2	2817

Сравнение построено на четырех семействах моделей, представляющих распространенные классы алгоритмов для табличных признаков: линейная логистическая регрессия, линейный метод опорных векторов, случайный лес и градиентный бустинг по гистограммам. Обучение выполнялось в библиотеке scikit-learn со стандартной схемой «fit — score», а воспроизводимость обеспечивалась фиксированием генераторов случайных чисел там, где применимо (в первую очередь для ансамблевых методов) (Scikit-learn, n.d.). Выбор именно этих моделей мотивирован практической применимостью в задачах детектирования: линейные методы дают интерпретируемый «скоринг», а ансамбли обычно обеспечивают устойчивость на разнородных и неидеально масштабированных признаках (Scikit-learn, n.d.).

В качестве базовых интегральных метрик дискриминации использовались площадь под ROC-кривой (ROC AUC) и площадь под кривой «точность–полнота» (PR AUC). ROC AUC отражает способность ранжировать объекты независимо от порога, а PR AUC

лучше согласуется с ситуациями, где положительный класс (вредоносные файлы) существенно важнее с точки зрения пропусков и операционных рисков.

Эксплуатационные решения в анти-контурах редко принимаются «по факту класса»; на практике требуется пороговое правило по скору, которое переводит модельный балл в действия (блокировать/отправить на ручную проверку/пропустить). Для приведения скорингов разных моделей к сопоставимой шкале дополнительно применялась калибровка вероятностей с помощью CalibratedClassifierCV в сигмоидальной постановке (вариант, близкий по смыслу к Platt scaling), обучаемой на внутренних разбиениях обучающей части (Scikit-learn, n.d.). Калибровка использовалась как отдельная ветка эксперимента, чтобы проверить, меняется ли устойчивость пороговых политик при переходе от «сырых» скорингов к вероятностно интерпретируемым оценкам (Scikit-learn, n.d.).

Пороговая настройка выполнена в логике ограничений на ложноположительные срабатывания (false positive rate, FPR) на безвредном классе, поскольку именно ложные блокировки формируют наибольшие операционные издержки. В работе использованы три взаимосвязанные схемы выбора порога:

1. Фиксация FPR по ROC (fixed-FPR thresholding). Порог подбирался так, чтобы эмпирический FPR на контрольной части не превышал заданный бюджет (например, 0.01), после чего оценивалась полнота (Recall) по вредоносному классу при этом же пороге.
2. Политика «block/review/allow» с «серой зоной» (gray zone). Два порога задают область автоматического блокирования (score ≥ thr_block), область автоматического пропуска (score < thr_allow) и промежуточный интервал (thr_allow ≤ score < thr_block), направляемый в песочницу (ручная/дополнительная проверка). Качество такой политики измеряется не только Recall и FPR, но и долей объектов, уходящих в песочницу (ReviewRate), а также долей «утечек» вредоносных файлов в разрешенную область (LeakRate) при заданном ограничении на допустимую утечку.
3. Порог по ограничению на число ложноположительных K (K-FP thresholding). Порог выбирался по обучающей части (через out-of-fold оценки), чтобы число ложноположительных блокировок на безвредных примерах не превышало K. Интерпретация K как «допустимого числа ошибок» удобно согласуется с инженерным планированием риска при малом числе доступных безвредных тестовых объектов. Графическая фиксация точек K на кривой компромисса Recall–ReviewRate приведена на рис. 1 (пример для сценария B) и рис. 2 (пример для сценария C), где подписи у точек соответствуют выбранному значению K.

Чувствительность и специфичность пороговых политик оценивались через матрицу ошибок (TP, FP, TN, FN), из которой рассчитывались Recall, Precision, F1 и FPR. Малое абсолютное число безвредных файлов в тесте приводит к «квантованию» FPR, поэтому для отчетности дополнительно использовались доверительные интервалы для доли FP на «безвредном» классе, вычисляемые как интервальная оценка параметра биномиального распределения; формат интервалов применялся

как средство аккуратной интерпретации нулевого FPR на конечной тестовой выборке, а не как доказательство отсутствия риска (Wallis, 2013).

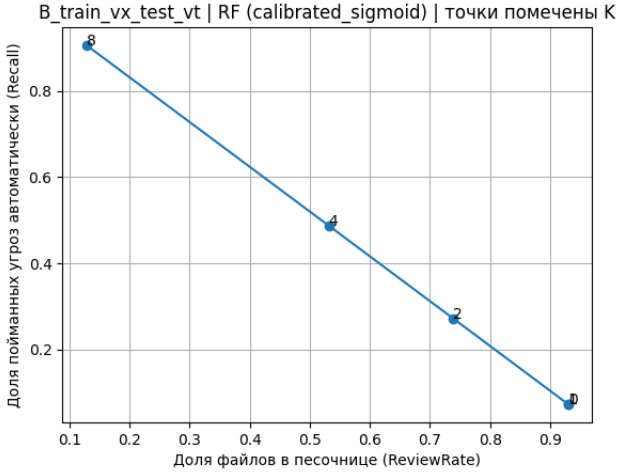

Рис. 1. Компромисс Recall–ReviewRate с маркировкой порогов K по FP (B_train_vx_test_vt, RF calibrated)

Fig. 1. Recall–ReviewRate trade-off with K-FP marked thresholds (B_train_vx_test_vt, RF calibrated)

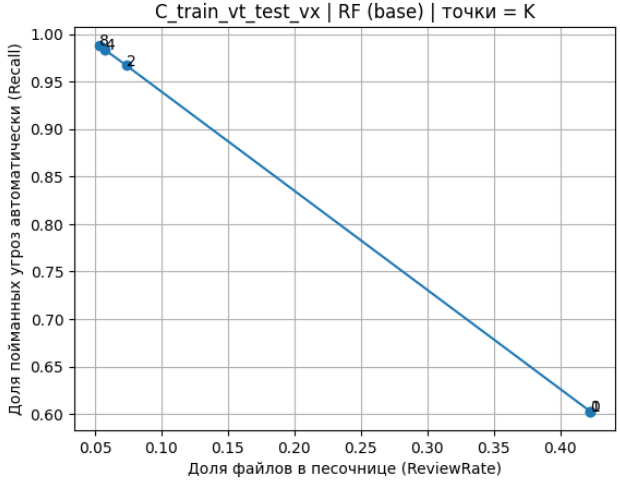

Рис. 2. Компромисс Recall–ReviewRate с маркировкой порогов K по FP (C_train_vt_test_vx, RF base)

Fig. 2. Recall–ReviewRate trade-off with K-FP marked thresholds (C_train_vt_test_vx, RF base)

Результаты

Пороговая настройка по «случайному» сценарию разбиения A_random_by_source дала почти идеальные значения ROC-AUC и PR-AUC для деревьев решений и бустинга, а также высокую полноту при строгом ограничении на ложные срабатывания. Переносимость в межисточниковых сценариях B_train_vx_test_vt и C_train_vt_test_vx оказалась заметно ниже, что согласуется с наблюдениями о смещениях выборки и переоценке качества при неаккуратных протоколах оценки в задачах детектирования вредоносных объектов (Botacin, Gomes, 2024; Gaber, Ahmed, Janicke, 2024; Kan et al., 2024). Различие проявилось именно в рабочей точке с контролем FPR, где влияние малых доменных сдвигов усиливается из-за необходимости поднимать порог блокировки.

Таблица 3 фиксирует лучшую базовую конфигурацию при целевом ограничении FPR_target = 0,01 без «серой зоны» (один порог блокировки). В сценарии A_random_by_source модель HGB обеспечивает Recall = 0,9708 при FPR = 0, тогда как в B_train_vx_test_vt полнота падает до 0,5320 при том же нулевом числе ложных блокировок. Дискретность FPR на тесте обусловлена малым числом безопасных объектов (benign) в контрольной части: при n_benign = 119 один ложноположительный случай соответствует шагу 1/119 ≈ 0,0084, поэтому достижение «точно 0,01» статистически невозможно, и фактически наблюдаются значения 0; 0,0084; 0,0168 и т. д.

Таблица 3 / Table 3

Базовая переносимость при FPR_target = 0,01 (однопороговая блокировка)

Baseline portability at FPR_target = 0.01 (single blocking threshold)

Сценарий (split)	Лучшая базовая модель	ROC-AUC	PR-AUC	Порог блокировки (thr)	FPR	Recall
A_random_by_source	HGB	0,999844	0,999984	0,999884	0,0000	0,9708
B_train_vx_test_vt	HGB	0,990707	0,999575	0,996054	0,0000	0,5320
C_train_vt_test_vx	HGB	0,996811	0,999847	0,999318	0,0084	0,8347

Компромисс между долей автоматической блокировки и нагрузкой на песочницу при переносимом сценарии B_train_vx_test_vt наглядно раскрывает кривая «Recall vs ReviewRate» для HGB после сигмоидной калибровки (рис. 3). Падение Recall при снижении ReviewRate на этом графике оказывается почти линейным в широком диапазоне, поэтому экономия ручной проверки требует заранее выбранного «бюджета на пропуски» — иначе снижение нагрузки быстро переходит в потерю значимой доли угроз. Величина ReviewRate в данной работе трактуется как доля объектов, попадающих в «серую зону» и уходящих на дополнительную проверку, тогда как Recall относится к автоматическому обнаружению угроз без участия песочницы.

Механизм «серой зоны» (Stage8) обеспечил гарантии по ложным блокировкам и утечкам за счет расширения области проверки, однако цена за контроль ошибок на переносимом сценарии B_train_vx_test_vt оказалась высокой. Конфигурация Stage8

с нулевой утечкой и FPR_target = 0,01 дала ReviewRate ≈ 0,48—0,51 при автоматической полноте Recall ≈ 0,51—0,54 (табл. 4), поэтому почти половина потока требует песочницы. Столбчатые диаграммы нагрузки на песочницу по сценарию B (рис. 4) подтверждают, что Stage8 оказывается наиболее «тяжелым» режимом среди сравниваемых политик.

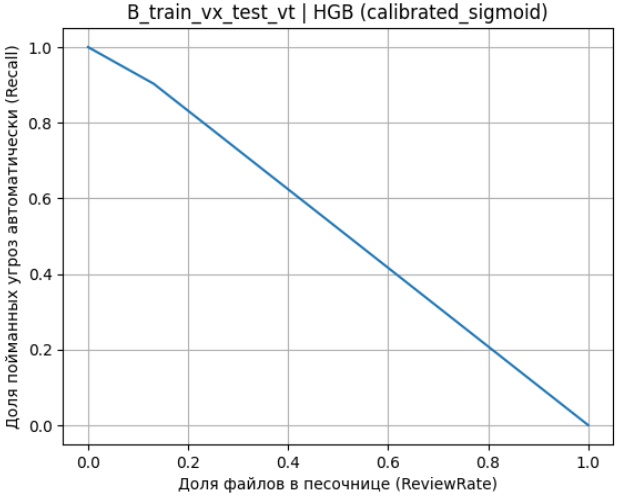

Рис. 3. Компромисс «доля найденных угроз автоматически (Recall) — доля файлов в песочнице (ReviewRate)» для сценария B_train_vx_test_vt (HGB, sigmoid-калибровка)

Fig. 3. Trade-off between automatic threat detection (Recall) and sandbox workload (ReviewRate) for B_train_vx_test_vt (HGB, sigmoid calibration)

Подбор порога по out-of-fold-критерию (Stage10, «OOF FPR») улучшил баланс переносимости в сценарии B_train_vx_test_vt, смещая модель в режим более уверенной автоматической блокировки при нулевых ложных блокировках на тесте. Лучшая конфигурация Stage10 для B (RF calibrated_sigmoid) достигла Recall = 0,7729 при ReviewRate = 0,2570 и FPR = 0 (табл. 4), сокращая песочницу относительно Stage8 почти вдвое без потери ограничения по FPR. Сценарий C_train_vt_test_vx показал важное ограничение подхода: выбранная по OOF-ограничению настройка для RF дала очень высокую полноту (Recall = 0,9863), но превысила бюджет ложных блокировок (FPR = 0,0168 > 0,01), поэтому подобная конфигурация не подходит для регламентов, где ложная блокировка безопасного файла считается инцидентом.

Критерий «K по FP» на out-of-fold-контуре (Stage12) дал наиболее стабильный компромисс между переносимостью и нагрузкой на песочницу при заданном «жестком» контроле ложных срабатываний. В сценарии B_train_vx_test_vt финальная политика Stage12 (RF base, K=8) обеспечила FPR = 0 на тесте (FP=0 из 119 benign) при Recall = = 0,8227 и ReviewRate = 0,2092 (табл. 6), что одновременно лучше Stage8 и Stage10 по полноте и по нагрузке. Дискретные точки K на кривой компромисса для RF в сценарии

В (рис. 5) выделяют выбранный режим как область, где дополнительное снижение ReviewRate ведет к непропорциональному росту пропусков, тогда как переход к более мягкой блокировке дает умеренное улучшение Recall при заметном росте песочницы.

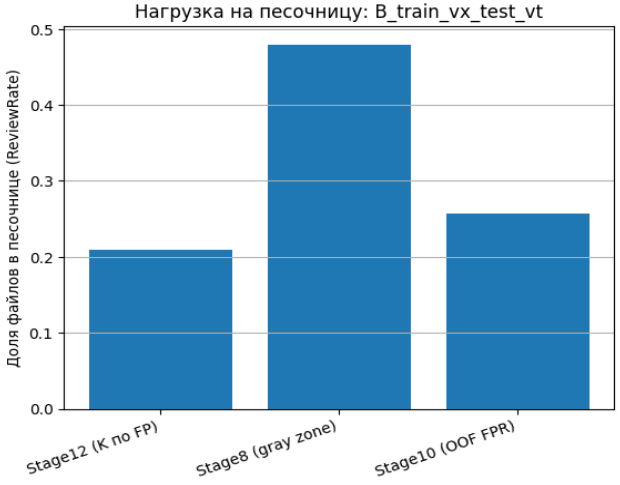

Рис. 4. Нагрузка на песочницу (ReviewRate) при сравнении политик Stage8/Stage10/Stage12 для сценария B_train_vx_test_vt

Fig. 4. Sandbox workload (ReviewRate) when comparing Stage8/Stage10/Stage12 policies for B_train_vx_test_vt

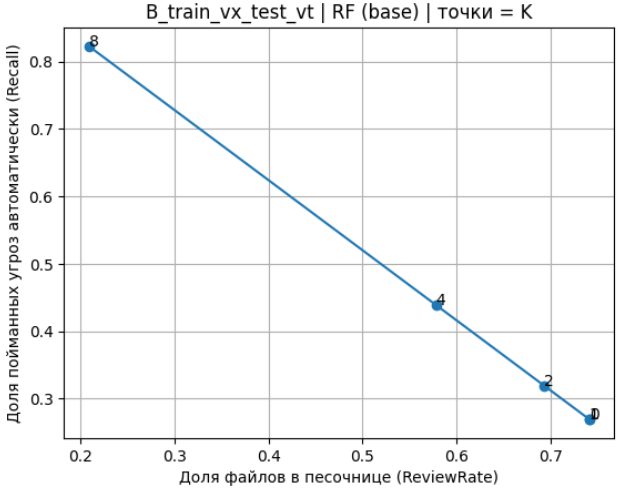

Рис. 5. Компромисс «Recall — ReviewRate» с отмеченными точками K для сценария B_train_vx_test_vt (RF, без калибровки)

Fig. 5. Recall–ReviewRate trade-off with marked K points for B_train_vx_test_vt (RF, uncalibrated)

Сводная таблица сравнения политик (табл. 4) и диаграмма компромисса «Recall vs ReviewRate» (рис. 6) показывают устойчивую тенденцию: переносимые сценарии выигрывают от политики Stage12, если целевым ограничением выступает FPR-бюджет, а песочница рассматривается как ограниченный ресурс. Для C_train_vt_test_vx Stage12 (RF base, K=2) удержал FPR = 0,0084 внутри бюджета и дал Recall = 0,9670 при ReviewRate = 0,0735 (табл. 6), оставаясь значительно легче по песочнице, чем Stage8 (рис. 7). При этом Stage10 в сценарии C демонстрирует привлекательную нагрузку (ReviewRate = 0,0547), но нарушает ограничение на ложные блокировки, что делает сравнение принципиально нерелевантным для практик с фиксированным FPR-регламентом.

Таблица 4 / Table 4

Сравнение политик Stage8 vs Stage10 vs Stage12
Policy comparison (Stage8 vs Stage10 vs Stage12)

split	method	model	model_stage	Recall	ReviewRate	FPR	note
B_train_vx_test_vt	Stage8 (gray zone)	HGB	calibrated_ sigmoid	0,541117	0,479831	0,0000	Leak=0, FPR_ target=0.01
B_train_vx_test_vt	Stage10 (OOF FPR)	RF	calibrated_ sigmoid	0,772927	0,256994	0,0000	FPR_budget_ OK
B_train_vx_test_vt	Stage12 (K по FP)	RF	base	0,822673	0,209174	0,0000	K_fp_train_ oof=8
C_train_vt_test_vx	Stage8 (gray zone)	HGB	calibrated_ sigmoid	0,843588	0,191693	0,0084	Leak=0, FPR_ target=0.01
C_train_vt_test_vx	Stage10 (OOF FPR)	RF	base	0,986286	0,054668	0,0168	FPR_budget превышен
C_train_vt_test_vx	Stage12 (K по FP)	RF	base	0,967013	0,073482	0,0084	K_fp_train_ oof=2

Прирост Stage12 относительно альтернативных политик количественно зафиксирован в табл. 5. Для B_train_vx_test_vt переход от Stage8 к Stage12 увеличил Recall на +0,2816 при снижении ReviewRate в 2,29 раза, а сравнение со Stage10 дало более умеренное улучшение полноты +0,0497 при дополнительном снижении нагрузки на песочницу. Для C_train_vt_test_vx Stage12 сохраняет выигрыш относительно Stage8 (рост Recall +0,1234 при снижении ReviewRate в 2,61 раза), а отрицательная разность Recall относительно Stage10 интерпретируется только в контексте того, что Stage10 в этом сценарии не удовлетворяет бюджетному ограничению по FPR.

Доверительные границы для FPR в финальной политике Stage12 рассчитаны по безопасным объектам теста и приведены в табл. 6, что важно при малом n_benign (Wallis, 2013). Нулевой наблюдаемый FPR в сценарии B формально совместим с ненулевой верхней границей из-за ограниченной мощности проверки (119 безопасных объектов), поэтому практическая эксплуатация требует накопления расширенного пула benign-примеров и периодической перепроверки порогов при изменении профиля входного трафика (Botacin, Gomes, 2024; Kan et al., 2024). Разница между сценарием B и C по устойчивости к доменному сдвигу дополнительно подчеркивает необходимость держать переносимые протоколы оценки в центре экспериментальной

программы, а не использовать случайное разбиение как единственный ориентир (Gaber, Ahmed, Janicke, 2024; Kan et al., 2024).

Рис. 6. Переносимость: компромисс «Recall — ReviewRate» для сопоставления Stage8, Stage10 и Stage12 в сценариях B_train_vx_test_vt и C_train_vt_test_vx

Fig. 6. Portability: Recall–ReviewRate trade-off comparing Stage8, Stage10, and Stage12 in B_train_vx_test_vt and C_train_vt_test_vx

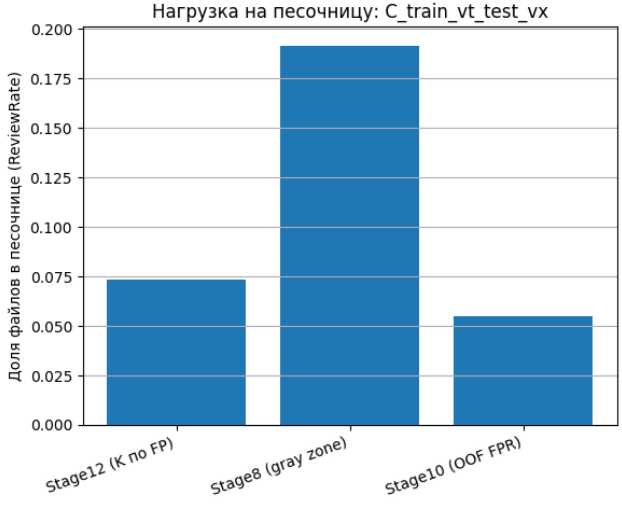

Рис. 7. Нагрузка на песочницу (ReviewRate) при сравнении политик Stage8/Stage10/Stage12 для сценария C_train_vt_test_vx

Fig. 7. Sandbox workload (ReviewRate) when comparing Stage8/Stage10/Stage12 policies for C_train_vt_test_vx

Таблица 5 / Table 5

Выигрыш Stage12 относительно Stage8 и Stage10

Gains of Stage12 vs Stage8 and Stage10

split	compare	dRecall_abs	dReview_abs	Review_reduction_x	base_Recall — new_Recall	base_ReviewRate — new_ReviewRate	base_FPR — new_FPR
B_train_vx_test_vt	Stage12 vs Stage8	+0,281557	–0,270657	2,293935	0,541117—0,822673	0,479831—0,209174	0,0000—0,0000
B_train_vx_test_vt	Stage12 vs Stage10	+0,049746	–0,047820	1,228616	0,772927—0,822673	0,256994—0,209174	0,0000—0,0000
C_train_vt_test_vx	Stage12 vs Stage8	+0,123425	–0,118211	2,608696	0,843588—0,967013	0,191693—0,073482	0,0084—0,0084
C_train_vt_test_vx	Stage12 vs Stage10	–0,019274	+0,018814	0,743961	0,986286—0,967013	0,054668—0,073482	0,0168—0,0084

Таблица 6 / Table 6

Финальная политика Stage12 (FP ≤ 1 на тесте, порог выбран по K на OOF)

Final Stage12 policy (FP ≤ 1 on test, threshold selected via K on OOF)

split	K_fp_train_oof	thr_block	Test_FPR (CI_low; CI_high)	Test_Recall	Test_ReviewRate	TP	FP	TN	FN
B_train_vx_test_vt	8	0,632	0,0000 (0,0000; 0,0305)	0,822673	0,209174	2431	0	119	524
C_train_vt_test_vx	2	0,818	0,0084 (0,0002; 0,0459)	0,967013	0,073482	2609	1	118	89

Обсуждение результатов

Практическая разница между «случайным» сценарием A и межисточниковыми сценариями B/C проявилась не в ранжировании (ROC-AUC и PR-AUC оставались высокими), а в рабочей точке с контролем ложных блокировок. Пороговые политики усиливают эффект доменного сдвига, потому что небольшое смещение распределения скоринга резко меняет долю объектов, пересекающих высокий порог автоматической блокировки. Высокие значения AUC в таком режиме перестают быть гарантией полезности, если регламент требует удерживать ложные срабатывания в пределах жесткого бюджета (Gaber, Ahmed, Janicke, 2024; Kan et al., 2024).

Сильная дискретность FPR на тесте при малом числе безопасных файлов сформировала важное ограничение интерпретации. Значение FPR «0» в выборке из 119 benign означает отсутствие ошибок в наблюдении, но не означает отсутствия риска в эксплуатации, где поток и состав «безвредных» объектов меняются. Доверительная верхняя граница для доли ложных блокировок в таких условиях остается заметной, поэтому устойчивость политики разумно подтверждать дополнительными «чистыми» наборами и регулярной переоценкой порогов на накопленной статистике (Wallis, 2013).

Политика «серой зоны» дала предсказуемую, но дорогую по ресурсу песочницы стабилизацию поведения при переносимости. Регламент Leak=0 фактически заставляет переносить значительную часть потока в проверку, потому что модель обязана избегать даже единичных «пропусков» в разрешенную область. Поведение оказалось особенно жестким в сценарии B, где вредоносный домен теста отличается сильнее: снижение утечек оплачивается расширением зоны ручного контроля, а не ростом автоматической полноты (Hendrickx et al., 2024; Liang, Peng, Sun, 2024).

Подбор порога по out-of-fold ограничению (Stage10) продемонстрировал, что «правильная» оценка порога внутри обучения может существенно уменьшить песочницу без нарушения FPR на части переносимых сценариев. Проблема проявилась при смене домена в противоположном направлении, где политика, корректная по внутренней оценке, все же превысила бюджет ложных блокировок на тесте. Нестабильность объясняется тем, что OOF-настройка остается зависимой от распределения обучающего домена, а бюджет по ошибкам задается именно по benign-классу, чьи свойства в разных доменах меняются особенно заметно (Botacin, Gomes, 2024; Kan et al., 2024).

Критерий «K по FP» (Stage12) оказался наиболее управляемым с инженерной точки зрения, поскольку связывает порог не с долей ошибок, а с допускаемым количеством ложных блокировок на контролируемом контуре. Выбор K задает понятный риск-профиль: небольшой K удерживает ложные блокировки, а рост K повышает автоматическую полноту ценой расширения допустимых ошибок на benign. Переход к такому параметру делает обсуждение порогов ближе к языку регламентов и эксплуатационных ограничений, где ответственность часто формулируется через «сколько ошибок допустимо» в заданном объеме контроля (Hendrickx et al., 2024; Hasan et al., 2025; Liang, Peng, Sun, 2024).

Сравнение базовой и калиброванной веток показало, что калибровка вероятностей полезна не как способ «повысить AUC», а как способ сделать шкалу скоринга более согласованной для пороговой политики. У части конфигураций калибровка улучшала компромисс Recall–ReviewRate, но эффект не был универсальным и зависел от домена теста. Роль калибровки в детектировании угроз разумнее трактовать как инструмент стабилизации принятия решений по порогам, а не как гарантированный «усилитель» качества модели (Ojeda et al., 2023; Scikit-learn, n.d.; Shaker, Hüllermeier, 2025).

Выбор финальной модели в виде случайного леса без калибровки в Stage12 имеет прагматичную интерпретацию: стабильность порогового поведения и переносимость оказались важнее небольших различий в интегральных метриках. Ансамбль деревьев демонстрирует устойчивость к неоднородным признакам и разреженности, а пороговая политика по K дополнительно снижает чувствительность к «форме» скоринга, опираясь на контроль ошибок по benign-примерам (Scikit-learn, n.d.; Shaker, Hüllermeier, 2025). Совместное действие этих факторов объясняет, почему финальная политика дала выигрыш по песочнице при сохранении строгого контроля ложных блокировок.

Ограничения исследования связаны с природой открытого набора данных и с размером безопасного теста. Доля benign в контрольных частях невелика, поэтому редкие ложные блокировки оцениваются с высокой неопределенностью, а переносимость

по «безвредному» классу требует более широкой проверки на независимых источниках (Wallis, 2013). Дополнительный риск несет устаревание признаковых профилей и изменение тактик злоумышленников, поэтому переносимость во времени и периодическое обновление порогов должны рассматриваться как обязательные элементы практического внедрения, а не как опциональная донастройка (Escudero García et al., 2023; Molina-Coronado et al., 2023; Kan et al., 2024).

Практическая ценность предложенного подхода проявляется в связке «регламент — измерение — порог». Регламент задает ограничение на ложные блокировки, измерение переводит его в проверяемый критерий на контрольном контуре, а пороговая политика обеспечивает управляемую нагрузку на песочницу при максимизации автоматического обнаружения. Конструкция Stage12 удобна для эксплуатации тем, что параметр K напрямую согласуется с ограниченными ресурсами проверки и допускаемым числом инцидентов ложной блокировки в заданном объеме контроля (Hendrickx et al., 2024; Hasan et al., 2025).

Заключение

Политика принятия решения при детектировании вредоносных программ определяется не только качеством ранжирования, но и тем, как модель переводится в действие при жестком контроле ложных блокировок. Эксперименты на наборе UCI 541 показали, что высокий ROC-AUC и PR-AUC в «случайном» разбиении не гарантируют переносимости при смене источника данных: в межисточниковых сценариях критической становится именно рабочая точка, где доля ложных блокировок ограничена малым бюджетом, а дискретность оценки по benign-классу усиливает неопределенность.

Поставленная цель исследования достигнута, задачи решены:

1. Подготовлен воспроизводимый протокол загрузки и согласования признаков для объединенного набора статических и динамических характеристик;
2. Реализованы сценарии оценки, разделяющие «удобный» случай (случайное смешивание источников) и переносимые режимы (обучение на одном источнике, тест на другом);
3. Построены базовые модели машинного обучения и показано, что переносимость ухудшается прежде всего в пороговых режимах при фиксированном бюджете ложных блокировок;
4. Разработаны и сопоставлены три политики порогового контроля с учетом эксплуатационного ресурса песочницы: «серая зона» (Stage8), порог по out-of-fold ограничению (Stage10) и порог по критерию K ложноположительных (Stage12);
5. Получено доказательное преимущество политики Stage12 как более управляемой и переносимой при заданных ограничениях на ложные блокировки.

Ключевой практический результат связан с тем, что переход от «серой зоны» к порогу, выбранному по критерию K ложноположительных на out-of-fold контуре, снижает нагрузку на песочницу без потери контроля FPR и при этом повышает долю автоматически обнаруженных угроз в переносимых сценариях. В сценарии B

(обучение на VxHeaven, тест на VirusTotal) Stage12 одновременно увеличил полноту по сравнению с Stage8 и уменьшил долю файлов, уходящих в песочницу; в сценарии C (обучение на VirusTotal, тест на VxHeaven) Stage12 удержал ложные блокировки внутри бюджета и обеспечил высокую полноту при существенно меньшей нагрузке на песочницу, чем у «серой зоны». Практическая интерпретация такого выигрыша проста: в условиях ограниченного ресурса песочницы и строгого контроля ошибок по безопасным объектам полезнее не усложнять модель, а формализовать политику порога так, чтобы параметр настройки соответствовал регламенту в терминах «допустимого числа ошибок», а не абстрактной доли.

Научная значимость работы выражается в уточнении того, как следует доказывать эффективность методов машинного обучения для обнаружения угроз при эксплуатационных ограничениях. Сопоставление политик показало, что переносимость следует оценивать не только по интегральным метрикам, но и по устойчивости выбранного порога в межисточниковом переносе, а бюджет ложных блокировок нужно задавать как первичное ограничение при проектировании режима работы. Результаты поддерживают вывод о том, что корректная «инженерная» постановка задачи — с формальным бюджетом ложных блокировок и измеримой нагрузкой на песочницу — способна дать больший прикладной эффект, чем попытки улучшать качество одной лишь модельной архитектурой.

Ограничения исследования связаны с малым числом безопасных объектов в тестовых частях, что делает оценку FPR дискретной и задает широкую верхнюю границу риска при наблюдаемом нуле ошибок. Расширение пула benign-примеров и проверка на дополнительных независимых источниках данных рассматриваются как первое направление продолжения работы. Второе направление связано с переносимостью во времени: при изменении профиля входного потока и эволюции вредоносных семейств потребуется регулярная переоценка порогов и контроль деградации политики Stage12 на новых периодах.

The study's **limitations** are related to the small number of safe (benign) objects in the test splits, which makes the FPR evaluation discrete and yields a broad upper bound on risk when zero errors are observed. Expanding the pool of benign examples and validating on additional independent data sources are considered the first direction for future work. The second direction concerns temporal transferability: as the profile of the incoming stream changes and malicious families evolve, regular re-estimation of thresholds and monitoring of Stage12 policy degradation on new time periods will be required.

Список источников / References

1. Архипов, А.Н., Кондаков, С.Е. (2024). Обнаружение обфусцированных эксплоитов в файлах неисполняемых форматов. Вопросы кибербезопасности, 6(64), 65—75. https://doi.org/10.21681/2311-3456-2024-6-65-75

Arkhipov, A.N., Kondakov, S.E. (2024). Detecting obfuscated exploits in non-executable format files. Cybersecurity issues, 6(64), 65—75. (In Russ.). https://doi.org/10.21681/2311-3456-2024-6-65-75

2. Калинкин, С.А., Голуб, М.С., Коркин, Д.А., Пятовский, И.В. (2022). Детектирование программ-шифровальщиков с использованием трассировки событий и методов машинного обучения. Безопасность информационных технологий, 29(3), 82—93. https://doi.org/10.26583/bit.2022.3.07
Kalinkin, S.A., Golub, M.S., Korkin, D.A., Pyatovskii, I.V. (2022). Detecting ransomware using event tracing and machine learning methods. Information Technology Security, 29(3), 82—93. (In Russ.). https://doi.org/10.26583/bit.2022.3.07

3. Костогрызов, А.И., Нистратов, А.А. (2023). Анализ угроз злонамеренной модификации модели машинного обучения для систем с искусственным интеллектом. Вопросы кибербезопасности, 5(57), 9—24. https://doi.org/10.21681/2311-3456-2023-5-9-24
Kostogryzov, A.I., Nistratov, A.A. (2023). Threat analysis of malicious modification of a machine learning model for artificial intelligence systems. Cybersecurity issues, 5(57), 9—24. (In Russ.). https://doi.org/10.21681/2311-3456-2023-5-9-24

4. Котенко, И.В., Хмыров, В.Д. (2022). Анализ моделей и методов обнаружения и атрибуции атак на основе искусственного интеллекта и машинного обучения. Вопросы кибербезопасности, 4(50), 52—79. https://doi.org/10.21681/2311-3456-2022-4-52-79
Kotenko, I.V., Khmyrov, V.D. (2022). Analysis of models and methods for attack detection and attribution based on artificial intelligence and machine learning. Cybersecurity issues, 4(50), 52—79. (In Russ.). https://doi.org/10.21681/2311-3456-2022-4-52-79

5. Лапина, М.А., Мовзалевская, В.В., Токмакова, М.Е., Бабенко, М.Г., Саджид, М. (2024). Применение технологий машинного обучения для обнаружения веб-атак. Вопросы кибербезопасности, 4(62), 92—103. https://doi.org/10.21681/2311-3456-2024-4-92-103
Lapina, M.A., Movzalevskaya, V.V., Tokmakova, M.E., Babenko, M.G., Sajid, M. (2024). Detecting web attacks using machine learning algorithms. Cybersecurity issues, 4(62), 92—103. (In Russ.). https://doi.org/10.21681/2311-3456-2024-4-92-103

6. Павлычев, А.В., Стародубов, М.И., Галимов, А.Д. (2022). Использование алгоритма машинного обучения Random Forest для выявления сложных компьютерных инцидентов. Вопросы кибербезопасности, 5(51), 74—81. https://doi.org/10.21681/2311-3456-2022-5-74-81
Pavlychev, A.V., Starodubov, M.I., Galimov, A.D. (2022). Using the Random Forest machine learning algorithm to identify complex computer incidents. Cybersecurity issues, 5(51), 74—81. (In Russ.). https://doi.org/10.21681/2311-3456-2022-5-74-81

7. Стародубов, М.И., Артемьева, О.А., Селин, Н.А. (2024). Анализ отчетов о вредоносных программах-шифровальщиках с использованием методов машинного обучения. Вопросы кибербезопасности, 3(61), 85—89. https://doi.org/10.21681/2311-3456-2024-3-85-89
Starodubov, M.I., Artem'eva, O.A., Selin, N.A. (2024). Analysis of ransomware reports using machine learning methods. Cybersecurity issues, 3(61), 85—89. (In Russ.). https://doi.org/10.21681/2311-3456-2024-3-85-89

8. Стародубов, М.И., Атомов, В.В., Ерофеев, В.А. (2025). Генерация синтетических данных для обучения моделей машинного обучения в задаче обнаружения вредоносного ПО. Вопросы кибербезопасности, 2(66), 105—113. https://doi.org/10.21681/2311-3456-2025-2-105-113
Starodubov, M.I., Atomov, V.V., Erofeev, V.A. (2025). Synthetic data generation for training machine learning models in malware detection. Cybersecurity issues, 2(66), 105—113. (In Russ.). https://doi.org/10.21681/2311-3456-2025-2-105-113

9. Bhardwaj, A., Esiere, R.C., Melenwane, L. (2024). Domain adaptation for malware detection: An adversarial approach (MD-ADA). Computers & Security, 137, Article 103588. https://doi.org/10.1016/j.cose.2024.103588
10. Botacin, M., Gomes, H. (2024). Cross-Regional Malware Detection via Model Distilling and Federated Learning. In: Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2024) (pp. 97—113). ACM. https://doi.org/10.1145/3678890.3678893
11. Escudero García, D., Hemberg, E., Harang, R., Rudd, E.M., O'Reilly, U.-M. (2023). Transfer learning for malware classification under concept drift. Expert Systems with Applications, 212, Article 118724. https://doi.org/10.1016/j.eswa.2022.118724
12. Gaber, M.G., Ahmed, M., Janicke, H. (2024). Malware Detection with Artificial Intelligence: A Systematic Literature Review. ACM Computing Surveys, 56(6), Article 148, 148:1—148:33. https://doi.org/10.1145/3638552
13. Hasan, M.A.M., Abdar, M., Rahman, M.S., et al. (2025). The Case of Reject Option and Post-Training Processing: A Systematic Review of Recent Advances. ACM Computing Surveys, 57(9), 1—35. https://doi.org/10.1145/3727633
14. Hendrickx, L., Perini, L., Bronzi, M., Davis, J. (2024). Machine Learning with a Reject Option: a survey. Machine Learning, 113, 3073—3110. https://doi.org/10.1007/s10994-024-06534-x
15. Kan, Z., McFadden, S., Arp, D., et al. (2024). TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time (Extended Version). arXiv preprint arXiv:2402.01359. https://doi.org/10.48550/arXiv.2402.01359
16. Liang, H., Peng, L., Sun, J. (2024). Selective classification under distribution shifts. Transactions on Machine Learning Research. URL: https://openreview.net/forum?id=dmxMGW6J7N (viewed: 25.01.2026).
17. Malware static and dynamic features VxHeaven and Virus Total: Dataset. (2019, January 30). UCI Machine Learning Repository. https://doi.org/10.24432/C58K6H
18. Maniriho, P., Mahoro, L.J., Niyigena, J.-P., Ahmad, A., Niyonzima, I., Nduwayo, G., Bizimana, Z. (2023). API—MalDetect: A novel approach for Windows malware detection using API call sequence analysis. Journal of Network and Computer Applications, 218, Article 103704. https://doi.org/10.1016/j.jnca.2023.103704
19. Molina-Coronado, J., Hernández-Álvarez, M., Manzano, M., Aparicio-Navarro, F.J., Bessani, A. (2023). Handling concept drift in batch Android malware detection models. Pervasive and Mobile Computing, 96, Article 101849. https://doi.org/10.1016/j.pmcj.2023.101849
20. Nguyen, D.C., Ding, M., Pathirana, P.N., et al. (2024). AutoML-based malware detection: A systematic review. Computers & Security, 137, Article 103582. https://doi.org/10.1016/j.cose.2023.103582
21. Ojeda, F.M., Keplinger, K., LoHuis, A.M., et al. (2023). Calibration approaches for probability predictions: A systematic evaluation. Statistics in Medicine, 42(29), 5451—5478. https://doi.org/10.1002/sim.9921
22. Scikit-learn. (n.d.). sklearn.calibration.CalibratedClassifierCV. Scikit-learn documentation. URL: https://scikit-learn.org/stable/modules/generated/sklearn.calibration.CalibratedClassifierCV.html (viewed: 25.01.2026).

23. Scikit-learn. (n.d.). sklearn.ensemble.HistGradientBoostingClassifier. Scikit-learn documentation. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html (viewed: 25.01.2026).
24. Scikit-learn. (n.d.). sklearn.ensemble.RandomForestClassifier. Scikit-learn documentation. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (viewed: 25.01.2026).
25. Shaker, M.H., Hüllermeier, E. (2025). Random forest calibration. Knowledge-Based Systems, 328, Article 114143. https://doi.org/10.1016/j.knosys.2025.114143
26. Wallis, S. (2013). Binomial confidence intervals and contingency tests: mathematical fundamentals and the evaluation of alternative methods. Journal of Quantitative Linguistics, 20(2), 178—208. https://doi.org/10.1080/09296174.2013.799918

Информация об авторах

Абедалхуссайн Ахмед Али, аспирант, Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет “МИСИС”» (НИТУ МИСИС), ORCID: https://orcid.org/0009-0004-5065-398X, e-mail: m2000009@edu.misis.ru

Ляпунцова Елена Вячеславовна, д.т.н., профессор, Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет “МИСИС”» (НИТУ МИСИС), ORCID: https://orcid.org/0000-0002-3420-3805, e-mail: lev77@me.com

Information about the authors

*Abedalhussain Ahmed Ali, G*raduate Student at National University of Science and Technology MISIS (NUST MISIS), ORCID: https://orcid.org/0009-0004-5065-398X, e-mail: m2000009@edu.misis.ru

Lyapuntsova Elena Vyacheslavovna, Doctor of Technical Sciences, Professor, National University of Science and Technology MISIS (NUST MISIS), ORCID: https://orcid.org/0000-0002-3420-3805, e-mail: lev77@me.com

Вклад авторов

Все авторы приняли участие в обсуждении результатов и согласовали окончательный текст рукописи.

Contribution of the authors

All authors participated in the discussion of the results and approved the final text of the manuscript.

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest

The authors declare no conflict of interest.

Поступила в редакцию 26.12.2026 — Received 2026.12.26
Поступила после рецензирования 15.01.2026 — Revised 2026.01.15
Принята к публикации 12.02.2026 — Accepted 2026.02.12
Опубликована 31.03.2026 — Published 2026.03.31