УДК 004.021

Методы классификации вредоносных программ на основе машинного обучения

Абедалхуссайн Ахмед Али[1] m2000009@edu.misis.ru
Ляпунцова Елена Вячеславовна[2] lev86@bmstu.ru

[1] *НИТУ МИСиС, Москва, Россия*
[2] *МГТУ им. Н.Э. Баумана, Москва, Россия*

Аннотация. *Рассмотрены сигнатурные методы и простые эвристики, которые 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.*

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

Machine-learning–based methods for malware classification

Abedalhussain Ahmed Ali[1] m2000009@edu.misis.ru
Lyapuntsova Elena Vyacheslavovna[2] lev86@bmstu.ru

[1] *NUST MISIS, Moscow, Russia*
[2] *BMSTU, Moscow, Russia*

Abstract. The article considers 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.

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

Введение. Поток файлов, попадающих в контуры корпоративной защиты (почта, веб-шлюзы, рабочие станции, репозитории), неизбежно включает долю неизвестных образцов, для которых сигнатурные правила и репутацион-

ные 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. 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 программного обеспечения (ПО) [1]. Классификация вредоносных программ с использованием методов машинного обучения — это важная задача в области кибербезопасности. Она позволяет выявлять и предотвращать атаки, основанные на вредоносных программах, таких как вирусы, черви, трояны и шпионские программы. Основные методы и подходы, которые применяются для классификации вредоносных программ являются: сбор и предобработка данных; статические методы и анализ кода; хеширование; динамические методы; алгоритмы машинного обучения; комбинированные подходы и другие.

Это может включать статические анализы (например, анализ файловых заголовков, строк и метаданных) и динамические анализы (например, поведение программ в песочнице). Удаление шумов, нормализация данных и преобразование их в формат, подходящий для алгоритмов машинного обучения. Статические методы и анализ кода: использование статического анализа для извлечения характеристик из кода вредоносной программы. Это может включать анализ байт-кода, вызовов функций и структур данных. Хеширование: создание хеш-значений для файлов и сравнение их с известными вредоносными образцами. Динамические методы, которые включают песочницы: запуск вредоносных программ в контролируемой среде для наблюдения за их поведением. Это может помочь выявить вредоносные действия, такие как изменение системных файлов или сетевые запросы. А также мониторинг системы: использование системных логов и мониторинга для выявления подозрительных действий.

Методы и материалы; результаты. 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») [2]. 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 [2].

Алгоритмы машинного обучения особенно часто применяются и зарекомендовали себя как эффективные методы. Классификаторы на основе деревьев решений: алгоритмы CART или Random Forest, которые могут эффективно обрабатывать большие объемы данных и выявлять сложные паттерны. Методы опорных векторов (SVM): эффективны для бинарной классификации и могут быть использованы для разделения вредоносных и безопасных программ. Нейронные сети: глубокое обучение, включая сверточные и рекуррентные нейронные сети, может быть использовано для обработки сложных данных и выявления скрытых паттернов. Байесовский классификатор: простой, но эффективный метод, который может быть использован для классификации на основе вероятностных моделей.

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 [3, 4].

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. Высокие значения AUC в таком режиме перестают быть гарантией полезности, если регламент требует удерживать ложные срабатывания в пределах жесткого бюджета. 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.

Политика «серой зоны» дала предсказуемую, но дорогую по ресурсу песочницы стабилизацию поведения при переносимости. 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.

Комбинированные подходы также применяются. Ансамблирование: использование нескольких моделей для повышения точности классификации. Это может быть реализовано через методы, такие как Bagging и Boosting, которые комбинируют результаты нескольких классификаторов для достижения более надежного результата. Гибридные методы: сочетание статических и динамических подходов для улучшения точности классификации.

Заключение. Использование метрик, таких как точность, полнота, F-мера и ROC-AUC, для оценки производительности моделей. Применение кросс-валидации для проверки устойчивости модели и предотвращения переобучения. Обновление, адаптация и обучение на новых данных дает возможность постоянного обновления моделей на основе новых образцов вредоносных программ для поддержания их актуальности. Разработка методов, которые могут адаптироваться к новым угрозам и изменяющимся паттернам поведения вредоносного ПО.

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. Подготовлен воспроизводимый протокол загрузки и согласования признаков для объединенного набора статических и динамических характеристик. Реализованы сценарии оценки, разделяющие «удобный» случай (случайное смешивание источников) и переносимые режимы (обучение на одном источнике, тест на другом). Построены базовые модели машинного обучения и показано, что переносимость ухудшается прежде всего в пороговых режимах при фиксированном бюджете ложных блокировок. Разработаны и сопоставлены три политики порогового контроля с учетом эксплуатационного ресурса песочницы. Получено доказательное преимущество политики как более управляемой и переносимой при заданных ограничениях на ложные блокировки.

Классификация вредоносных программ с использованием машинного обучения — это динамичная и развивающаяся область, которая требует постоянного обновления знаний и навыков, чтобы эффективно противостоять новым угрозам в киберпространстве.

Список источников

[1] Стародубов М.И., Атомов В.В., Ерофеев В.А. Генерация синтетических данных для обучения моделей машинного обучения в задаче обнаружения вредоносного ПО. *Вопросы кибербезопасности*, 2025, № 2(66), с. 105–113. https://doi.org/10.21681/2311-3456-2025-2-105-113

[2] Malware static and dynamic features VxHeaven and Virus Total: *набор данных*. URL: https://archive.ics.uci.edu/dataset/541/malware+static+and+dynamic+features+vxheaven+and+virus+total (accessed 20.12.2025).

[3] Kan Z., McFadden S., Arp D. TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time (Extended Version), 2024, 30 p. URL: https://www.bifold.berlin/impact-transfer/publications/view/publication-detail/tesseract-eliminating-experimental-bias-in-malware-classification-across-space-and-time-extended-version (accessed 20.12.2025).

[4] Botacin M., Gomes H. Cross-regional malware detection via model distilling and federated learning. Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2024), 2024, pp. 97–113. https://doi.org/10.1145/3678890.3678893