Alzubairi Shaymaa M. Jawad Kadhim – Ural Federal University; e-mail: Shaymaaalzubairi77@gmail.com; Yekaterinburg, Russia; research engineer.

Petunin Alexander Alexandrovich – Ural Federal University; e-mail: a.a.petunin@urfu.ru; Yekaterinburg, Russia; dr. of eng. sc., associate professor; professor, leading researcher of the N.N. Krasovskii Institute of Mathematics and Mechanics.

Ukolov Stanislav Sergeevich – Ural Federal University; e-mail: s.s.ukolov@urfu.ru; Yekaterinburg, Russia; cand. of eng. sc.; senior researcher.

УДК 004.056.5+004.492 DOI 10.18522/2311-3103-2025-5-18-35

Ал.В. Козачок, С.С. Матовых, Ан.В. Козачок

КАСКАДНЫЙ АЛГОРИТМ КЛАССИФИКАЦИИ ДЛЯ ОБНАРУЖЕНИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕТОДОМ СТАТИЧЕСКОГО АНАЛИЗА

Представлено исследование, посвященное разработке и экспериментальной валидации двухуровневой каскадной архитектуры статической классификации исполняемых файлов формата Portable Executable (PE). Целью работы является разработка и экспериментальная оценка каскадного алгоритма статической классификации, направленного на снижение вычислительных затрат при сохранении качества обнаружения вредоносного программного обеспечения. На первом уровне каскада применяется модель дерево решений, обученная на десяти наиболее информативных признаках, обеспечивающая высокую полноту обнаружения Recall 0,990 при приемлемой ошибке 1 рода. Второй уровень реализован моделью случайный лес на сорока признаках и предназначен для уточняющей классификации, достигая метрик Precision 0,988 и Recall 0,987 при F1-мере 0,988. Порог классификации на первом уровне был установлен эмпирически с учётом минимизации ошибок второго рода, тогда как на втором уровне оптимальное значение порога определялось по индексу Юдена, обеспечивающему сбалансированное соотношение чувствительности и специфичности. Эксперименты на репрезентативной выборке показали, что при доле вредоносного трафика ≤ 20 % предложенный каскад сокращает среднее время анализа одного объекта на 5–12 % по сравнению с моделью на 40 признаках при сохранении сопоставимого качества классификации. Аналитически выведена граница применимости каскада по времени P_M = 20,6 %, подтвержденная эмпирическими данными. Практическая значимость работы заключается в возможности интеграции предложенного алгоритма в антивирусные шлюзы и средства защиты конечных точек, где требуются быстрый отклик и высокая полнота обнаружения при массовом сканировании преимущественно легитимного кода.

Вредоносное программное обеспечение; статический анализ; файлы формата Portable Executable; каскадный классификатор; машинное обучение; индекс Юдена.

Al.V. Kozachok, S.S. Matovykh, An.V. Kozachok

CASCADE CLASSIFICATION ALGORITHM FOR DETECTING MALICIOUS SOFTWARE BY STATIC ANALYSIS

A study is presented on the development and experimental validation of a two-level cascading architecture for static classification of Portable Executable (PE) format executable files. The aim of the work is to reduce computing costs without compromising the quality of malware detection. At the first level of the cascade, a decision tree model is used, trained on the ten most informative features, providing a high completeness of Recall 0.990 detection with an acceptable error of 1 kind. The second level is implemented by the random forest model on forty features and is intended for clarifying classification, reaching the metrics Precision 0.988 and Recall 0.987 with an F1 measure of 0.988. The classification threshold at the first level was established empirically, taking into account the minimization of errors of the second kind, while at the second level the optimal threshold value was determined by the Juden index, which provides a balanced ratio of sensitivity and specificity. Experiments on a representative sample have shown that with a malicious traffic fraction of < 20%, the proposed cascade reduces the average analysis time of one object by 5-12% compared to the 40-feature model while maintaining comparable classification quality. The time limit of the cascade, P_M = 20.6%, is analytically derived, confirmed by empirical data. The prac-

tical significance of the work lies in the possibility of integrating the proposed algorithm into antivirus gateways and endpoint protection tools, where fast response and high completeness of detection are required during mass scanning of mostly legitimate code.

Malicious software; static analysis; Portable Executable files; cascade classifier; machine learning; Yuden index.

Введение. Уже в ранних исследованиях по применению методов машинного обучения к задаче классификации исполняемых файлов формата Portable Executable (PE) была продемонстрирована принципиальная возможность автоматического выявления вредоносных объектов с использованием байтовых n-грамм в качестве признаков. Однако высокая вычислительная сложность таких моделей и необходимость обработки больших объемов входных данных существенно ограничивали их практическую применимость в реальных системах анализа [1].

С развитием глубоких нейронных сетей удалось добиться значительного повышения точности детектирования при одновременном снижении уровня ложноположительных классификаций. Тем не менее, как отмечается в ряде работ, такие методы остаются чувствительными к временным затратам и вычислительной нагрузке, особенно в условиях потоковой обработки многомиллионных массивов PE-файлов [2].

Более поздние исследования показали, что использование структурных и семантических признаков, извлекаемых из заголовков, секций, таблиц импорта и других полей формата PE, позволяет построить более эффективные модели. В частности, комбинирование различных статических признаков (например, DLL- и API-импорты, характеристики секций, размеры ресурсов и т.п.) обеспечивает высокую устойчивость классификаторов к вариантам вредоносных программ «нуля дня», при сохранении приемлемого уровня точности и низкой чувствительности к обфускации [3]. Эти результаты подтверждают, что статический анализ остается фундаментально значимым подходом в архитектуре современных систем обнаружения вредоносного ПО.

Ключевым фактором надежного статического анализа является корректное моделирование внутренней структуры формата Portable Executable, регламентированной спецификацией Microsoft PE/COFF [4]. Использование структурных полей (размеры секций, флаги загрузчика, адреса экспортов и импортов) обеспечивает интерпретируемость признаков и минимизирует риск обхода за счет поверхностных изменений. Тем не менее, модели, опирающиеся на сотни признаков, демонстрируют линейный рост времени обработки с объемом данных, что неприемлемо для систем превентивной фильтрации почтового и веб-трафика, работающих в режиме реального времени.

Настоящая работа продолжает исследования, изложенные в статье «Структурная модель файлов формата Portable Executable, содержащих вредоносный код» [5]. В ней была предложена структурная модель, включающая исходно 333 бинарных признака, а также описан подход оптимизации признакового пространства, позволивший существенно уменьшить число признаков до 40 наиболее информативных без потери точности классификации (F1-мера 0,982). Было установлено, что сокращение числа признаков до 10 обеспечивает приемлемый уровень точности (F1-мера 0,918), существенно сокращая время обработки файлов и обеспечивая возможность быстрого первичного анализа.

Целью данного исследования является разработка и экспериментальная оценка каскадного алгоритма классификации на основе указанных наборов признаков. Предлагаемый алгоритм предусматривает двухуровневую схему обработки файлов, где первый уровень выполняет первичную быструю фильтрацию с минимальным набором из 10 признаков, быстро отсеивая очевидно легитимные или явно вредоносные объекты. А второй уровень, состоящий из 40 признаков предназначен для детальной и точной классификации, обеспечивая высокую точность финального решения с оптимальным выбором адаптивных порогов, определенных по индексу Юдена. Таким образом, достигается экономия ресурсов, где каждый файл обрабатывается ровно до того уровня детализации, который необходим для вынесения решения с заданной достоверностью.

Обзор литературы. Методы статического анализа PE-файлов являются надежной основой для классификации вредоносных программ. Ранние исследования показывают, что использование признаков из заголовков, секций и таблиц импорта позволяет достичь точности свыше 98–99%. Работа [6] рассматривает детекторы на базе PE Header, а также анализирует эффективность классификаторов, построенных на ограниченном числе признаков. Авторы приходят к выводу, что чрезмерное увеличение размерности признакового пространства не всегда приводит к повышению точности, но существенно влияет на производительность.

В то же время возникает необходимость балансировки между качеством классификации и временем обработки. Исследование [7] демонстрирует подход, при котором используется фильтрация на основе простых эвристик, за которой следуют более точные, но ресурсоемкие модели. Концептуально близкой является идея каскадной классификации, впервые предложенная в контексте обработки изображений Виолой и Джонсом [8]. В их работе применялась цепочка классификаторов с возрастающей сложностью для ускоренного обнаружения объектов.

Ближайшим прямым аналогом разработанной каскадной архитектуры является работа [9], в которой предложена многоступенчатая схема классификации на основе ансамблей моделей (Random Forest, Bagging, Gradient Boosting) с мягким голосованием и регуляризацией. В отличие от большинства традиционных решений, признаки в данной работе формируются на уровне байтов и опкодов по схеме TF-IDF с учётом межклассовой вариативности, что позволяет повысить обобщающую способность моделей. Экспериментальная валидация выполнена на репрезентативном наборе Microsoft Big2015. По результатам испытаний получены следующие показатели на полной выборке Accuracy 98,97 %, Precision 98,59 %, Sensitivity 98,94 %, Specificity 98,87 %, F1-мера 98,18 %.

В области информационной безопасности концепции каскадных моделей получили развитие в работах [10]. Авторы разработали архитектуры, включающие несколько уровней фильтрации PE-файлов с использованием кластеризации и нейросетей. Отдельного внимания заслуживает система PROUD-MAL [11], демонстрирующая преимущества каскада для задач анализа исполняемых файлов. Такие архитектуры позволяют перераспределять ресурсы анализа: быстрые модели обрабатывают большинство образцов, а тяжелые применяются лишь к неоднозначным случаям.

Одной из актуальных задач при построении классификаторов для выявления вредоносного программного обеспечения является выбор порогового значения, разделяющего положительный (вредоносный) и отрицательный (легитимный) классы. При этом оптимизация порога оказывает непосредственное влияние на соотношение между ошибками первого и второго рода, особенно в условиях классовой несбалансированности и асимметричной стоимости ошибок. В ряде работ [12, 13] в качестве рационального критерия выбора порога предлагается использовать индекс Юдена (Youden’s J-statistic), который учитывает одновременно чувствительность (Recall) и специфичность (Specificity), максимизируя разницу между истинноположительной и ложноположительной классификацией.

Анализ эффективности различных метрик качества в задачах бинарной классификации представлен в [14], где рассмотрены F1-мера, точность (Precision), полнота (Recall), а также ROC-кривая, площадь под ROC-кривой (AUC) и показатель осведомленности (Informedness).

С учетом анализа современного состояния исследований в данной области можно сформулировать следующие положения:

- статический анализ PE-файлов остается надежным источником признаков, применимых для машинной классификации;
- избыточное увеличение признакового пространства приводит к существенному росту времени анализа, без гарантированного повышения качества классификации;
- многоуровневые (каскадные) архитектуры позволяют реализовать обработку, при которой вычислительно простые модели отсеивают очевидные случаи, снижая общую нагрузку на систему;
- индекс Юдена и аналогичные пороговые критерии представляют собой эффективный инструмент для калибровки классификаторов.

Методы исследования. Проведенное исследование включает совокупность взаимосвязанных этапов, направленных на реализацию поставленной цели – разработку и экспериментальную верификацию каскадного алгоритма статической классификации PE-файлов [15]. Общая структура методики представлена на рис. 1 и включает следующие этапы.

На 1 этапе для проведения анализа была собрана репрезентативная выборка из 34 026 исполняемых файлов формата PE, включающая как вредоносные, так и легитимные [16]. При разделении на обучающую и тестовую выборки соблюдался классовый баланс, обеспечивающий достоверность оценки обобщающей способности моделей.

2 этап заключался в выборе оптимальных наборов признаков для каскадного классификатора. В качестве признаков использовались характеристики, извлекаемые средствами статического анализа. На основании результатов предварительного анализа важности признаков, выполненного с применением метода Extra Trees, были выделены два подмножества, 10 признаков для начального уровня каскада и расширенное 40 признаков для уточняющей классификации [17].

На 3 этапе был проведен сравнительный анализ времени обработки объектов при использовании моделей, обученных на различных объемах признаков [18]. Измерялись как средние, так и медианные значения времени анализа для вредоносных и легитимных файлов. Особое внимание уделялось выявлению взаимосвязи между размером признакового пространства и вычислительной нагрузкой.

4 этап включал настройку порогов классификации, где на первом уровне каскада порог классификации подбирался эмпирически с учетом минимизации ошибки второго рода (FNR) при допустимом уровне ложноположительных срабатываний [19]. На втором уровне порог определялся по индексу Юдена, обеспечивающему оптимальный баланс между чувствительностью и специфичностью. Дополнительно сравнивались модели, реализующие различные алгоритмы машинного обучения.

Заключительный 5 этап состоял из комплексной верификации разработанной архитектуры по совокупности метрик качества (Precision, Recall, F1), а также по временным характеристикам. Анализ включал расчет относительной экономии вычислительных ресурсов.

Рис. 1. Порядок проводимых исследований

В настоящем исследовании рассматривается построение двухуровневого каскадного алгоритма классификации исполняемых файлов формата Portable Executable (PE), основанного исключительно на признаках, извлекаемых методами статического анализа. Для проведения экспериментальной оценки была использована репрезентативная выборка, сформированная в рамках ранее выполненной работы, включающая 34 026 PE-файлов, из которых 17 992 являются вредоносными, предоставленные ресурсом Virusshare.com [20], а 16 034 – легитимными. Статистические характеристики выборки обеспечивали достаточную полноту и разнообразие наблюдений, что позволило гарантировать корректность как качественной, так и количественной интерпретации результатов.

Формирование признакового пространства основывалось на результатах предварительного этапа оптимизации, в рамках которого исходный массив из 333 признаков, полученных в ходе статического анализа, был подвергнут процедуре снижения размерности с использованием метода главных компонент (PCA) и алгоритма изолирующего леса (Isolation Forest). По итогам оценки информативности и вычислительной стоимости извлечения признаков были выделены два подмножества – из 10 и 40 признаков соответственно, обеспечивающих оптимальное соотношение между точностью классификации и временем анализа.

Первый уровень каскада модели на 10 признаков предназначен для первичной фильтрации и выявления заведомо вредоносных объектов при минимальной ошибке второго рода (False Negative Rate, FNR), что значимо с точки зрения обеспечения надежности системы. Второй уровень модели на 40 признаков осуществляет классификацию файлов, не распознанных на первом уровне, обеспечивая максимально возможную точность при ограниченном увеличении временных затрат.

Каждому уровню каскада соответствовала отдельная модель машинного обучения. Для первого уровня использовался алгоритм Decision Tree Classifier, обладающий высокой скоростью и интерпретируемостью. На втором уровне применялся Random Forest, обеспечивающий устойчивость к шуму и стабильные результаты на расширенном множестве признаков. Обучение моделей проводилось на соответствующих подвыборках, сформированных по стратифицированной схеме. Для второго уровня дополнительно осуществлялась оптимизация порога классификации на основе индекса Юдена, что обеспечивало сбалансированное соотношение между ошибкой первого рода (False Positive Rate, FPR) и второго рода (FNR). На первом уровне порог классификации был зафиксирован эмпирически на уровне 0.16 на основании предварительного анализа ошибки второго рода.

Для оценки вычислительной эффективности были проведены замеры среднего времени обработки одного файла при различной размерности признакового пространства [21]. Отдельно анализировались временные характеристики для легитимных и вредоносных объектов, а также зависимость времени обработки от размера PE-файла. Полученные результаты позволили наглядно продемонстрировать, что предлагаемая каскадная архитектура обеспечивает значительное сокращение времени анализа по сравнению со структурной моделью, использующей полный набор признаков.

Результаты исследования

Анализ вычислительных затрат при классификации PE-файлов

Одним из ключевых аспектов разработки эффективной архитектуры классификации PE-файлов является анализ временных характеристик, возникающих при использовании признаков различной размерности. Поскольку методы статического анализа не предполагают запуск исполняемого кода, они представляют собой приоритетный подход в системах превентивной фильтрации вредоносных объектов. Однако даже в таких условиях критически важным остается параметр времени отклика – особенно в системах, функционирующих в режиме приближенного к реальному времени.

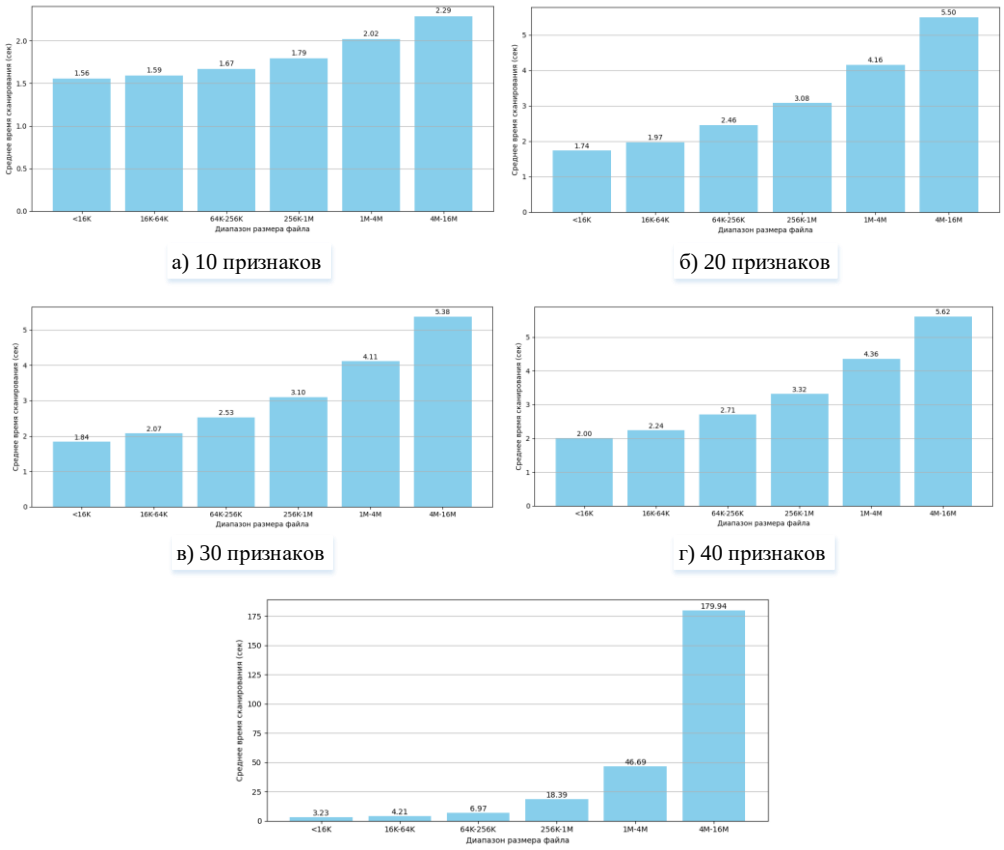
С целью получения воспроизводимых и репрезентативных результатов был разработан специализированный скрипт на языке Python, реализующий многопоточную обработку с точным измерением времени выполнения анализа. Для каждого испытуемого файла выполнялось $N = 5$ независимых измерений, на основе которых вычислялись такие статистики, как среднее значение, медиана и дисперсия времени обработки. Результаты сохранялись в форматах JSON и CSV, обеспечивая прозрачность и последующую возможность визуализации. Такая экспериментальная процедура позволяет оценить устойчивость временных характеристик и их зависимость от архитектуры классификатора.

Экспериментальные измерения проводились на пяти моделях, обученных на подмножествах признаков размерностью 10, 20, 30, 40 и 333 признака (полный набор). Тестирование выполнялось раздельно для двух классов: легитимных и вредоносных PE-файлов. Для повышения точности анализа выборка в каждой из двух групп была дополнительно стратифицирована по шести диапазонам объема файлов меньше 16 КБ, от 16 до 64 КБ, от 64 до 256 КБ, от 256 КБ до 1 МБ, от 1 до 4 МБ, от 4 до 16 МБ.

В каждом интервале рассчитывалось среднее время анализа одного файла соответствующей моделью. Полученные значения легли в основу количественной оценки временной эффективности и были использованы для обоснования выбора каскадной архитектуры.

Результаты, представленные на рис. 2, демонстрируют отчетливую зависимость времени обработки от размерности признакового пространства. В интервале от 10 до 40 признаков наблюдается близкая к линейной динамика роста вычислительных затрат. На-

пример, для крупных файлов объемом 4–16 МБ среднее время анализа составляет 2.29 с при использовании модели на 10 признаках и увеличивается до 5.62 с при переходе к модели на 40 признаках. Существенный рост времени наблюдается при применении модели, содержащей все 333 признака: в данном случае среднее время обработки объектов среднего размера достигает 179.94 с, что делает такую модель непригодной для использования в системах, требующих высокой скорости реагирования.



а) 10 признаков

б) 20 признаков

в) 30 признаков

г) 40 признаков

д) 333 признака

Рис. 2. Среднее время анализа легитимных PE-файлов по диапазонам размера при использовании модели размером 10 признаков а), размером 20 признаков б), размером 30 признаков в), размером 40 признаков г), размером 333 признака д)

Таблица 1

Среднее время сканирования легитимных PE-файлов

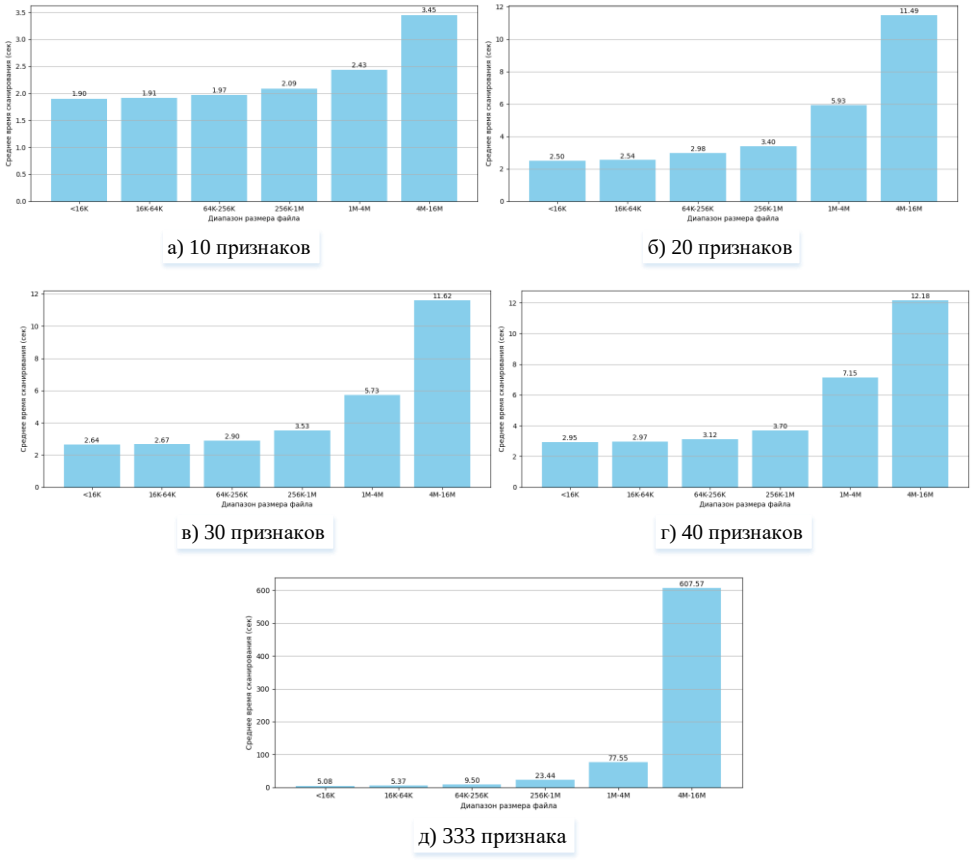
Признаки	Объем файлов					
	<16K	16K-64K	64K-256K	256K-1M	1M-4M	4M-16M
10 признаков	1.56 с	1.59 с	1.67 с	1.79 с	2.02 с	2.29 с
20 признаков	1.74 с	1.97 с	2.46 с	3.08 с	4.16 с	5.5 с
30 признаков	1.84 с	2.07 с	2.53 с	3.1 с	4.11 с	5.38 с
40 признаков	2.0 с	2.24 с	2.71 с	3.32 с	4.36 с	5.62 с
333 признака	3.23 с	4.21 с	6.97 с	18.39 с	46.69 с	179.94 с

Анализ временных затрат при классификации вредоносных PE-файлов выявил закономерности, сходные с результатами, полученными для легитимных объектов, однако с более выраженной зависимостью времени анализа от как количества признаков, так и размера файла. Визуализация результатов, представленных на рис. 3, демонстрирует, что даже при использовании моделей с ограниченным признаковым пространством (10–40 признаков) наблюдается существенный рост времени обработки по мере увеличения объема входных данных.

Так, при применении модели на 40 признаках среднее время анализа PE-файлов в интервале размера 4–16 МБ достигает 12.18 с, что почти в 3.5 раза превышает аналогичный показатель для легитимных объектов аналогичного объема. При этом модели на 10, 20 и 30 признаках также демонстрируют постепенное увеличение затрат времени: от 1.90 с при размере <16 КБ до 11.62 с при размере 4–16 МБ (табл. 2).

Наиболее выраженный рост наблюдается при использовании модели, включающей все 333 признака. В этом случае среднее время анализа вредоносного файла объемом 4–16 МБ составляет 607.57 с, что указывает на экспоненциальный рост вычислительной нагрузки при увеличении объема данных и сложности признакового пространства. Этот результат согласуется с известными особенностями вредоносных объектов: применение упаковки, шифрования, полиморфизма и иных методов обфускации, а также высокая энтропия и структурная сложность, существенно увеличивают затраты на статический анализ.

Полученные данные подчеркивают критическую важность выбора эффективной структуры классификатора и обоснованного подмножества признаков для обеспечения практической применимости моделей в условиях ограниченных вычислительных ресурсов.



а) 10 признаков

б) 20 признаков

в) 30 признаков

г) 40 признаков

д) 333 признака

Рис. 3. Среднее время анализа вредоносных PE-файлов по диапазонам размера при использовании модели размером 10 признаков а), размером 20 признаков б), размером 30 признаков в), размером 40 признаков г), размером 333 признака д)

Таблица 2

Среднее время сканирования вредоносных PE-файлов

Признаки	Объем файлов					
	<16K	16K-64K	64K-256K	256K-1M	1M-4M	4M-16M
10 признаков	1.90 с	1.91 с	1.97 с	2.09 с	2.43 с	3.45 с
20 признаков	2.50 с	2.54 с	2.98 с	3.40 с	5.93 с	11.49 с
30 признаков	2.64 с	2.67 с	2.90 с	3.53 с	5.73 с	11.62 с
40 признаков	2.95 с	2.97 с	3.12 с	3.70 с	7.15 с	12.18 с
333 признака	5.08 с	5.37 с	9.50 с	23.44 с	77.55 с	607.57 с

Для комплексного анализа вычислительных затрат при классификации объектов различной природы была выполнена визуализация распределения времени обработки с использованием диаграмм размаха (boxplot), представленных на рис. 4. Такой подход позволил выявить различия в характеристиках времени анализа между легитимными и вредоносными PE-файлами при различной размерности признакового пространства [22].

Анализ показал, что медианные значения времени обработки вредоносных файлов систематически превышают соответствующие значения для легитимных объектов во всем диапазоне моделей. Например, при использовании модели на 10 признаках медианное время обработки составляет 1.93 с для вредоносных файлов и 1.61 с для легитимных (разница +19.88 %). Для структурной модели на полном наборе признаков (333 признака) соответствующие значения составляют 6.60 с и 3.84 с, что эквивалентно росту на 72.02 % (табл. 3).

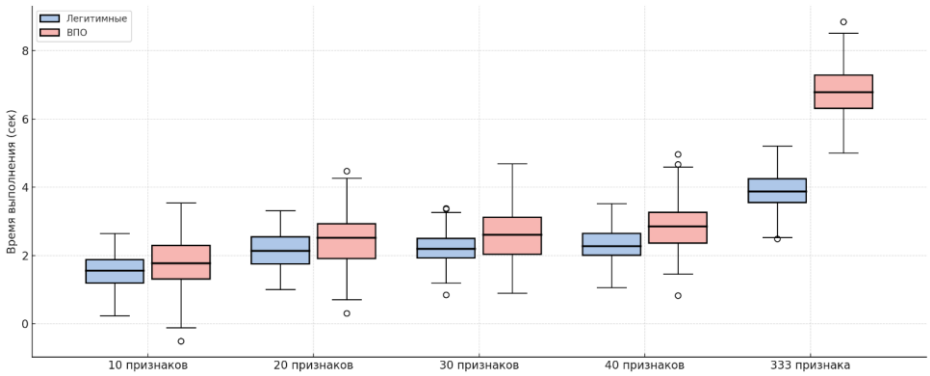


Рис. 4. Распределение времени обработки легитимных и вредоносных PE-файлов в зависимости от количества признаков (визуализировано с помощью диаграмм размаха, boxplot)

Таблица 3

Сравнение медианных времен для легитимных файлов и ВПО

Признаки	Медиана (Легитимные)	Медиана (ВПО)	Медиана по обоим классам	Относительное превышение
10 признаков	1.61	1.93	1.77	+19.88
20 признаков	2.12	2.57	2.35	+21.23
30 признаков	2.23	2.69	2.46	+20.63
40 признаков	2.39	2.90	2.65	+21.34
333 признака	3.84	6.60	5.22	+71.88

Кроме того, вредоносные файлы демонстрируют значительно больший интерквартильный размах, а также увеличенное количество выбросов, что указывает на высокую вариативность времени анализа. Такая дисперсия, как правило, обусловлена различиями в степени обфускации, наличии упаковщиков, нестандартных структурных сегментов и других усложняющих факторов, характерных для ВПО [23].

Анализ распределения временных затрат, представленный на рис. 4, позволяет зафиксировать устойчивое расхождение между легитимными и вредоносными объектами при одинаковых параметрах классификационной модели. Установлено, что вредоносные PE-файлы характеризуются не только более высокими медианными значениями времени анализа, но и значительно большим интерквартильным размахом, что указывает на высокую неоднородность сложности их обработки. Это связано с применением техник обфускации, упаковки и шифрования, а также со структурной сложностью вредоносных исполняемых файлов, приводящей к увеличению времени извлечения и обработки признаков.

Для оценки общей вычислительной нагрузки, связанной с применением моделей различной сложности, была проанализирована зависимость среднего времени анализа от числа признаков и класса объекта. Агрегированные результаты представлены на рис. 5 и в табл. 4.

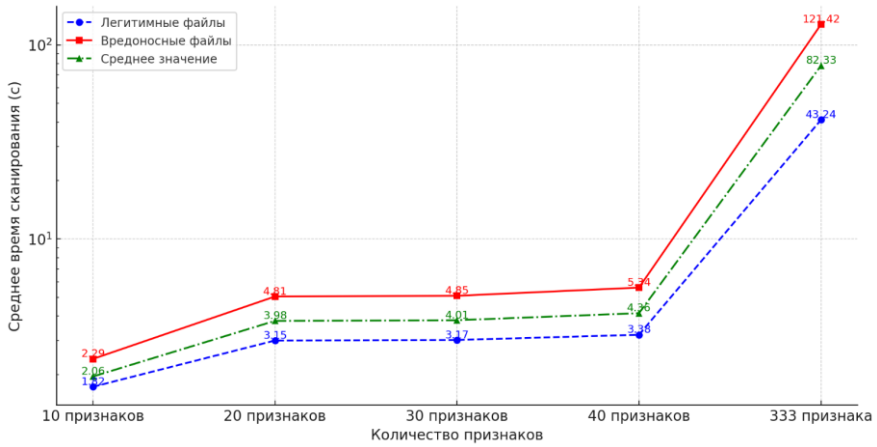


Рис. 5. Зависимость среднего времени анализа легитимных и вредоносных файлов PE-файлов от количества признаков

Таблица 4

Зависимость среднего времени анализа

Признаки	Легитимные файлы (с)	Вредоносные файлы (с)	Среднее значение (с)
10 признаков	1.82	2.29	2.06
20 признаков	3.15	4.81	3.98
30 признаков	3.17	4.85	4.01
40 признаков	3.38	5.34	4.36
333 признака	43.24	121.42	82.33

Анализ зависимости времени обработки от размерности признакового пространства показал, что в диапазоне от 10 до 40 признаков рост вычислительных затрат носит умеренный характер. Так, среднее время анализа легитимных файлов увеличивается с 1.82 до 3.38 с, а вредоносных с 2.29 до 5.34 с. Совокупное среднее значение по обоим классам возрастает с 2.06 до 4.36 с. Такие значения являются приемлемыми для задач первичного анализа в условиях ограниченных ресурсов. В этом диапазоне наблюдается почти линейная зависимость времени от числа признаков, с наибольшим приростом при переходе с

10 на 20 признаков. Увеличение признакового пространства до 30 и 40 признаков приводит к дальнейшему, но более плавному нарастанию затрат, что делает модель на 40 признаках оптимальной по соотношению между качеством классификации и вычислительной нагрузкой [24].

В противоположность этому, использование полной модели на 333 признаках вызывает экспоненциальный рост времени анализа до 43.24 с для легитимных и 121.42 с для вредоносных объектов, при среднем значении 82.33 с. Такая нагрузка делает модель непригодной для применения в реальном времени или в сценариях массовой проверки PE-файлов. Резкое увеличение затрат подтверждает чувствительность процесса статического анализа к размерности признакового пространства и подчёркивает необходимость предварительной фильтрации объектов с целью оптимального распределения ресурсов.

На этом фоне эффективным представляется каскадный подход, объединяющий модели на 10 и 40 признаках. Первая обеспечивает минимальные задержки и используется для быстрого анализа однозначных случаев, в то время как вторая обеспечивает углублённую проверку «сложных» файлов. Подобное распределение нагрузки позволяет достичь баланса между точностью и производительностью, снижая общее время анализа без ущерба для качества классификации.

Настройка порогов классификации каскада

На этапе построения каскадной классификационной архитектуры ключевое значение приобретает задача оптимального выбора пороговых значений (thresholds), определяющих поведение каждого уровня классификатора (рис. 6). В отличие от традиционного подхода, где порог выбирается по максимуму общей точности или F1-меры, в условиях каскадной архитектуры необходимо обеспечить строгий баланс между ошибками первого (ложноположительные) и второго рода (ложноотрицательные) на каждом уровне. Особенно значимой является ошибка второго рода (FNR) на начальном уровне каскада, так как пропущенные вредоносные объекты не будут проверены на последующих стадиях, нарушая надежность всей системы.

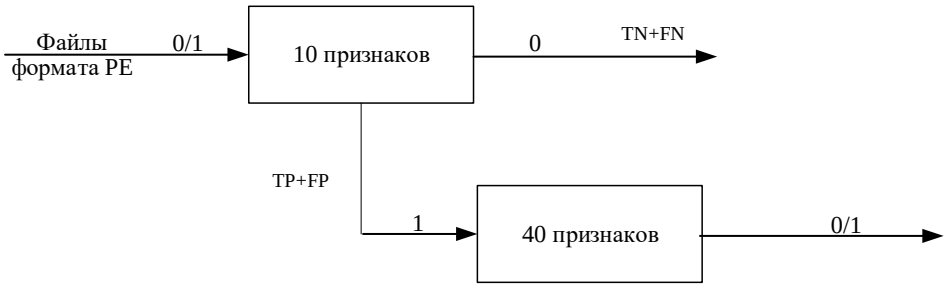


Рис. 6. Структура каскадного классификатора

Ввиду принципиальной значимости контроля ошибок второго рода в каскадной структуре, настройка пороговых значений была начата с анализа параметров второго уровня – модели, построенной на подмножестве из 40 признаков. Эта модель обладает более высокой дискриминационной способностью по сравнению с первым уровнем, что позволяет рассматривать ее в качестве опорной при определении максимально допустимого уровня FNR [25]. Фактически, рассчитанное на втором уровне значение FNR служит порогом, который не должен быть превышен на предыдущих, менее точных этапах, поскольку это приводит к необратимому пропуску вредоносных объектов и подрывает надежность всей системы.

Второй уровень выполняет функцию уточняющей классификации и требует обеспечения сбалансированного соотношения между полнотой (Recall) и точностью (Precision), что необходимо для стабильного функционирования каскада при переходе от грубой фильтрации к более строгому анализу. Для выбора оптимального порогового значения на данном уровне использовался индекс Юдена (Youden’s J-statistic), определяемый формулой:

$$J = TPR + TNR - 1 = Recall + (1 - FPR) - 1, \quad (1)$$

где TPR – истинно положительная классификация (Recall), а TNR – истинно отрицательная классификация (1 − FPR). Максимизация данного критерия позволяет учитывать одновременно оба класса и минимизировать влияние дисбаланса.

В результате анализа зависимости метрик от порога (табл. 5, рис. 7), оптимальным для модели на 40 признаках признано значение порога 0.54. При данном значении достигаются следующие метрики: Precision = 0.988, Recall = 0.986, FPR = 0.012, FNR = 0.013, при максимальном значении Youden_J = 0.973. Таким образом, модель демонстрирует высокую селективность и низкий уровень обоих типов ошибок, что делает ее надежной основой для принятия решений на втором этапе.

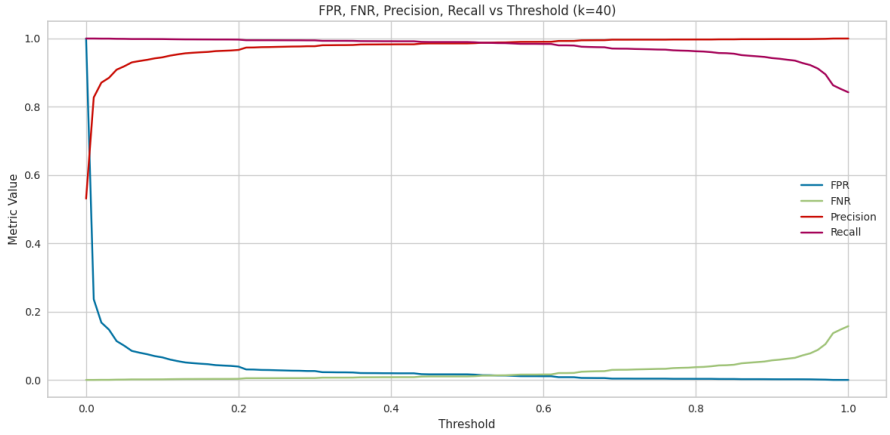


Рис. 7. Графики зависимости метрик от порогового значения для 40 признаков

Таблица 5

Метрики порогов классификации для 40 признаков

№ п/п	Threshold	FPR	FNR	Precision	Recall	Youden_J
1	0.5	0.016	0.010	0.985	0.989	0.973
2	0.51	0.015	0.010	0.986	0.989	0.973
3	0.52	0.013	0.012	0.987	0.987	0.973
4	0.53	0.013	0.012	0.987	0.987	0.973
5	**0.54**	**0.012**	**0.013**	**0.988**	**0.986**	**0.973**
6	0.55	0.012	0.013	0.988	0.986	0.973
7	0.56	0.011	0.014	0.989	0.985	0.973
8	0.57	0.011	0.015	0.990	0.984	0.973
9	0.58	0.010	0.015	0.990	0.984	0.973
10	0.59	0.010	0.015	0.990	0.984	0.973

После фиксации оптимального порогового значения для модели второго уровня, основанной на 40 признаках, следующим этапом стала калибровка первого уровня каскада, использующего модель с 10 признаками. Ключевым требованием при выборе порога на этом этапе являлось обеспечение полноты выявления вредоносных объектов (Recall) не ниже, чем на втором уровне, то есть сохранение ошибки второго рода (FNR) в пределах, допустимых по результатам более точной модели. Это критически важно, поскольку файлы, ошибочно классифицированные как легитимные на первом уровне, не поступают

на дальнейшую проверку, что может привести к пропуску угроз. Таким образом, первый уровень должен гарантировать максимально полное обнаружение, даже ценой увеличения числа ложноположительных срабатываний.

Дополнительно учитывался показатель FPR, поскольку чрезмерно высокий уровень ложных срабатываний приводит к росту нагрузки на модель второго уровня, что снижает эффективность всей каскадной структуры. На основе анализа кривых зависимости показателей Recall, FNR и FPR от порогового значения (табл. 6, рис. 8), было установлено, что оптимальным является порог 0.16. В этой точке достигается следующее соотношение метрик Recall = 0.990, FNR = 0.009, FPR = 0.407, Precision = 0.733, Youden_J = 0.583.

Несмотря на умеренное значение точности, такой режим функционирования полностью соответствует функциональному назначению первого уровня каскада – осуществлять первичную фильтрацию с приоритетом на максимальную полноту выявления. Благодаря этому обеспечивается, что общее качество распознавания, с точки зрения недопущения пропуска вредоносных объектов, не будет ниже, чем на более глубоком уровне каскада, что является критически важным требованием.

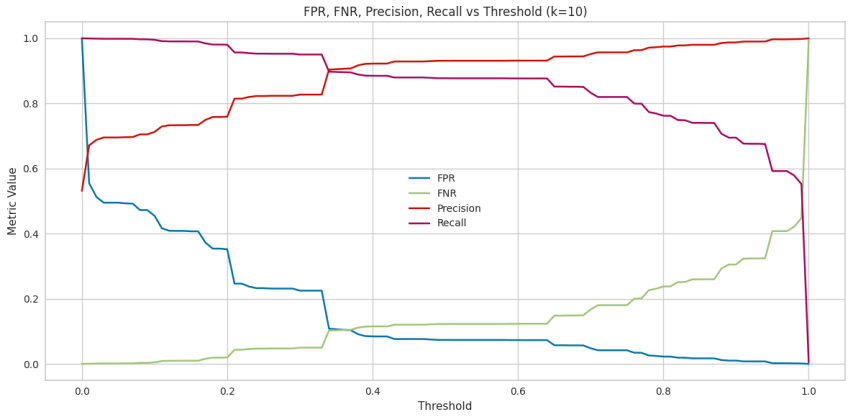


Рис. 8. Графики зависимости метрик от порогового значения для 10 признаков

Таблица 6

Метрики порогов классификации для 10 признаков

№ п/п	Threshold	FPR	FNR	Precision	Recall
1	0.12	0.409	0.009	0.732	0.990
2	0.13	0.408	0.009	0.733	0.990
3	0.14	0.408	0.009	0.733	0.990
4	0.15	0.407	0.009	0.733	0.990
5	**0.16**	**0.407**	**0.009**	**0.733**	**0.990**
6	0.17	0.372	0.015	0.749	0.984
7	0.18	0.354	0.019	0.758	0.980
8	0.19	0.354	0.019	0.758	0.980
9	0.2	0.352	0.019	0.759	0.980
10	0.21	0.246	0.043	0.814	0.956

Таким образом, итоговая схема порогов определяется как:

- Уровень 1 (10 признаков) порог = 0.16 высокий Recall, допустимый FPR.
- Уровень 2 (40 признаков) порог = 0.54 сбалансированная точность и полнота.

Для реализации каждого уровня были выбраны соответствующие модели на основе анализа их производительности. На первом уровне применен Decision Tree Classifier, обеспечивающий минимальные вычислительные затраты и высокую интерпретируемость. На втором уровне – Random Forest Classifier, демонстрирующий высокую устойчивость и стабильные показатели качества. Сводные характеристики моделей представлены в табл. 7.

Таблица 7

Результаты классификации моделей

Признаки	Модель	Threshold	Accuracy	Precision	Recall	F1-мера	Ошибка I рода (α)	Ошибка II рода (β)
10	DecisionTreeClassifier	0.16	0.803	0.733	0.990	0.842	0.407	0.009
40	RandomForestClassifier	0.54	0.986	0.988	0.986	0.987	0.012	0.013

Выбор порогов и моделей был подтвержден дополнительными экспериментами и визуализацией зависимости метрик от значения порога (рис. 7, 8). Такая архитектура позволяет реализовать адаптивную стратегию анализа, в которой легковесная модель, использующая 10 признаков быстро фильтрует очевидные случаи, а более точная модель на 40 признаках уточняет классификацию для спорных объектов.

Подобный подход обеспечивает не только высокое качество классификации, но и рациональное распределение вычислительной нагрузки, позволяя эффективно обрабатывать большие объемы PE-файлов в системах статического анализа. Полученные результаты формируют обоснование для перехода к следующему этапу – оценке интегральной временной эффективности и анализу распределения нагрузки между уровнями каскада.

Расчет временной эффективности и границы применимости каскадного классификатора

Для объективной количественной оценки эффективности предложенной каскадной архитектуры был проведен анализ временных затрат на обработку PE-файлов при прохождении через различные уровни классификации. Основная цель расчета заключалась в сравнении предложенного каскадного подхода с моделью, использующей расширенный набор из 40 признаков, при сохранении сопоставимого уровня качества классификации.

В качестве исходных параметров были приняты следующие экспериментальные значения среднее время анализа одного файла на первом уровне каскада составляет $T_{\text{ср}}^{10}$ 2.06 сек, на втором уровне $T_{\text{ср}}^{40}$ 4.36 сек. На первом уровне классификации установлены значения ошибок первого и второго рода FPR_{10} 0.407 и FNR_{10} 0.0097 соответственно. Это означает, что 40.7 % легитимных файлов ошибочно классифицируются как вредоносные и направляются на второй уровень каскада. Далее на втором уровне, использующем более ресурсоемкую модель с 40 признаками, они будут повторно классифицированы.

Для обоснования применимости каскадной архитектуры с точки зрения временной эффективности был разработан аналитический подход, учитывающий не только характеристики ошибок, но и априорные вероятности принадлежности объекта к тому или иному классу. Обозначив долю вредоносных файлов во входном потоке как P_M, а долю легитимных как $P_L = 1 - P_m$ можно выразить ожидаемое среднее время анализа одного файла через взвешенные доли объектов, проходящих каждый уровень:

$$T_{\text{каскад}} = T_{\text{ср}}^{10} + T_{\text{ср}}^{40} \cdot (P_M \cdot TPR + P_L \cdot FPR), \quad (2)$$

где $T_{\text{каскад}}$ – время анализа одного файла каскадным классификатором,

TPR – доля корректно распознанных вредоносных объектов $(1 - FPR)$.

Сравнение приведенного значения $T_{\text{каскад}}$ с временем анализа $T_{\text{ср}}^{40}$, соответствующим модели на 40 признаках, позволяет определить границу применимости каскадного подхода. Необходимым условием его эффективности является выполнение неравенства, $T_{\text{каскад}} < T_{\text{ср}}^{40}$ подставляя численные значения в выражение:

$$T_{\text{каскад}} = 2.06 + 4.36 \cdot \left(0.991P_M + 0.407 \cdot (1 - P_M)\right) = 3.85 + 2.53P_M, \quad (3)$$

решая уравнение $T_{\text{каскад}} = T_{\text{ср}}^{40}$, получаем $P_M = 0{,}206$, что соответствует критической доле вредоносных объектов во входном потоке, равной 20,6 %. Таким образом, при $P_M < 20{,}6\%$ каскадная архитектура обеспечивает выигрыш по времени в сравнении с моделью на 40 признаках. При превышении данного порога каскад становится менее эффективным с точки зрения временных затрат.

Относительный выигрыш по времени определим формуле:

$$G(P_M) = \left(1 - \frac{T_{\text{каскад}}}{T_{\text{ср}}^{40}}\right) \cdot 100\%, \quad (4)$$

Ниже приведены конкретные значения, полученные при различных соотношениях классов. График сравнения времени сканирования каскадной модели и модели с 40 признаками в зависимости от доли ВПО (рис. 9, табл. 8):

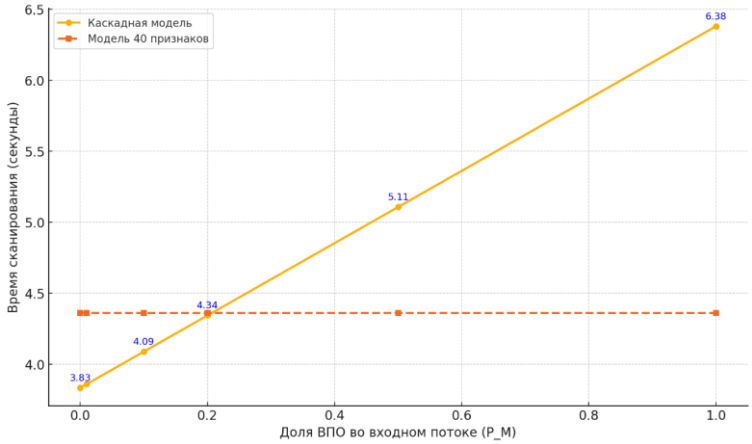


Рис. 9. График зависимость времени сканирования от доли ВПО

Таблица 8

Результаты эффективности каскадной архитектуры

Доля ВПО, P_M	Доля легитимных файлов, P_L	Время каскада, $T_{\text{каскад}}$ секунд	Выигрыш по времени, (%)
0.00	1.00	3.834	12.05
0.01	0.99	3.860	11.47
0.10	0.90	4.089	6.21
0.20	0.80	4.344	0.37
0.50	0.50	5.108	-17.15
1.00	0.00	6.381	–46.35

Проведённый анализ показал, что временная эффективность каскадной архитектуры определяется соотношением классов в анализируемом потоке. При доле вредоносных объектов менее 20.6 % каскад демонстрирует выигрыш по времени, достигающий до 12 % по сравнению с моделью, использующей 40 признаков, при этом сохраняя сопоставимые значения основных метрик качества [26]. С увеличением доли вредоносных файлов нагрузка на второй уровень возрастает, что снижает эффективность каскада и в предельном случае может привести к превышению временных затрат по сравнению с моделью на 40 признаках [27]. Таким образом, значение $P_M = 0{,}206$ может рассматриваться как эмпирически обоснованная граница применимости каскадной схемы с точки зрения её временной рациональности.

На основании полученных результатов можно утверждать, что каскадная архитектура целесообразна для использования в системах статического анализа, ориентированных на обработку потоков с преобладанием легитимного трафика. В таких условиях она обеспечивает снижение вычислительных затрат за счёт ранней фильтрации простых случаев при сохранении требуемого уровня точности. При превышении доли вредоносных объектов рациональность использования каскада должна определяться дополнительными факторами, включая специфику системы, допустимую задержку и критичность ошибок классификации. Таким образом, предложенная модель наиболее эффективна в сценариях массового предварительного анализа с преобладанием доверенного содержимого.

Обсуждение. Разработанная двухуровневая каскадная архитектура статической классификации PE-файлов продемонстрировала устойчивую результативность как по качественным, так и по вычислительным показателям. На первом уровне каскада применяется дерево решений, обученное на десяти наиболее информативных признаках; его основная задача максимально быстрое отсечение потенциально вредоносных объектов при минимальном риске пропуска. При эмпирически подобранном пороге 0,16 модель обеспечивает высокую полноту 0,990 при умеренной точности 0,734, что допустимо для каскадной структуры, в которой все сомнительные экземпляры перенаправляются на второй уровень анализа.

Второй уровень реализован на ансамбле, выполняющем уточняющую классификацию с порогом 0,54. Данная комбинация обеспечивает итоговую F1-меру 0,987 при среднем времени анализа одного PE-файла около 4 с. В совокупности каскадная схема демонстрирует оптимальный баланс между скоростью и достоверностью классификации в условиях преобладания легитимного трафика.

В сравнении с одноуровневыми моделями каскадный подход обеспечивает эквивалентное качество распознавания при снижении среднего времени обработки на 5–12 % в диапазоне долей вредоносных объектов до 20 %. Таким образом, при сохранении высокой F1-меры достигается существенное повышение производительности без ухудшения качества детекции. Это подтверждает целесообразность использования двухуровневой организации классификатора в условиях ограниченных вычислительных ресурсов и высокой интенсивности потоков PE-файлов.

Ближайшим архитектурным аналогом предложенного решения является каскадная модель MDCML, ранее рассмотренная в обзоре литературы. В отличие от PE-Cascade, где используются структурные PE-индикаторы и бинарная классификация с калиброванными порогами решений, MDCML опирается на TF-IDF-признаки, извлекаемые из последовательностей байтов и опкодов и реализует мультиклассовую постановку на датасете Microsoft BIG-2015. Различия в исходных данных, целевой функции и системе метрик не позволяют выполнять прямое сопоставление по времени анализа и ошибкам I/II рода в рамках текущего датасета. Поэтому обсуждение ограничено сравнением архитектурных принципов и уровня достигаемых показателей по порядку величин, без воспроизведения конкретных числовых значений. Для корректного сравнения требуется единый протокол испытаний, включающий идентичный набор данных, унифицированные метрики такие как FPR, FNR и согласованную методику измерения задержки при проверке файлов.

Заключение. В настоящем исследовании, относящемся к области информационной безопасности и машинного обучения для статического обнаружения ВПО, разработан двухуровневый каскадный классификатор PE-файлов. Его конструкция основана на рациональном разграничении признакового пространства, где первичное решение принимает модель на 10 статических признаках, тогда как углубленная верификация выполняется моделью на 40 признаках. Такой подход дополняется формализованной процедурой настройки порогов по индексу Юдена, что обеспечивает требуемое соотношение ошибок первого и второго рода.

Практический эффект выразился в ускорении обработки без заметного ухудшения обнаружения при характерной для прикладных систем доле вредоносного трафика. Каскад сокращает среднее время анализа одного файла на 5–12 %, одновременно сохраняя значения F1-меры на уровне 0,987. Тем самым подтверждена возможность сочетать высокую полноту выявления с приемлемым временем анализа, что важно для шлюзовых и конечных средств защиты.

Полученный подход подтвердил, что каскадный классификатор позволяет выиграть в ресурсах при адаптации его параметров и масштабировать статическое сканирование под реальные нагрузки. Перспективами дальнейших работ можно выделить автоматическую корректировку порогов под изменяющееся соотношение классов и использование динамических признаков, что позволит еще более повысить надежность обнаружения при сохранении достигнутой производительности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Schultz M.G., Eskin E., Zadok E., Stolfo S.J.* Data mining methods for detection of new malicious executables // Proc. IEEE Symp. Security and Privacy (S&P). – 2001. – P. 38-49.
2. *Kuang H., Wang J., Li R., Feng C., & Zhang X.* Automated Data-Processing Function Identification Using Deep Neural Network // IEEE Access. – 2020. – Vol. 8. – P. 55411-55423. – doi: 10.1109/ACCESS.2020.2981537.
3. *Ghanem K., Kherbache Z., Ourdighi O.* Enhancing Adversarial Examples for Evading Malware Detection Systems: A Memetic Algorithm Approach // IJCNIS. – 2025. – Vol. 17, No. 1. – P. 1-16. – DOI: 10.5815/ijcnis.2025.01.01.
4. Microsoft. Microsoft Portable Executable and Common Object File Format Specification. – Режим доступа: https://learn.microsoft.com/ru-ru/windows/win32/ debug/pe-format (дата обращения: 26.05.2025).
5. *Козачок А.В., Матовых С.С.* Структурная модель файлов формата Portable Executable содержащих вредоносный код // Проблемы информационной безопасности. Компьютерные системы. – 2025. – № 2. – С. 41-59. – DOI: 10.48612/jisp/pdu2-fvxz-g5d3.
6. *Rúa E.A., Bulut I.* Machine Learning-Based Secure Malware Detection Using Features from Binary Executable Headers // European Symposium on Research in Computer Security. – Springer, 2025. – P. 204-216. – DOI: 10.1007/978-3-031-82362-6_12.
7. *Al Balawi M., Alnabhan M.* Generative AI for Advanced Malware Detection // 4th Intelligent Systems Conference (IntelliSys). – IEEE, 2024. – P. 204-216. – DOI: 10.1109/ICSC63108.2024. 10895965.
8. *Petrean D.E., Potolea R., Oprisa C.* Packed Code Detection Using Shannon Entropy and Homomorphic Encrypted Executables // Proceedings of the 20th International Conference on Intelligent Computer Communication and Processing. – IEEE, 2024. – P. 01-08. – DOI: 10.1109/ICCP63557. 2024.10793050.
9. *Mahato A., Majumdar R., Ghosh S.K.* Feature-Driven Malware Detection using Cascade Machine Learning Models // SN Computer Science. – 2025. – Vol. 6, No. 7. – P. 794. – https://doi.org/10.1007/s42979-025-04342-1.
10. *Alizada Adil and Ragab Hassen Hani*. Pextract: A Light-Weight Static Feature Extractor for Windows Portable Executable Files // SSRN. – 2025. – Режим доступа: https://ssrn.com/abstract=5165659 (дата обращения: 26.05.2025).
11. *Kumar S.S., Shetty J.* Malicious PE File Detection Using Machine Learning: An Analysis of Header Features // COSMIC. – IEEE, 2024. – P. 66-71. – DOI: 10.1109/ COSMIC63293.2024.10871898.
12. *Rizwan M., Ali E., Batool N.* Assessing Concept Drift in Malware: A Comprehensive Review and Analysis // IBCAST. – IEEE, 2024. – P. 564-569 – DOI: 10.1109/IBCAST61650. 2024.10876901.
13. *Schubert Kabban C.M., Graham S.R.* Malware Classification through Abstract Syntax Trees and L-moments // Computers & Security. – 2025. – Vol. 133. – Article ID: 104082. – DOI: 10.1016/j.cose.2024.104082.
14. *Canbek G., Temizel T.T., Sagiroglu S.* PToPI: A comprehensive review, analysis, and knowledge representation of binary classification performance measures/metrics // SN Computer Science. – 2022. – Vol. 4, Article No. 13. – DOI: 10.1007/s42979-022-01409-1.
15. A survey of machine learning methods and challenges for Windows malware classification. – Режим доступа: https://arxiv.org/abs/2006.09271 (дата обращения: 28.05.2025).
16. *Jusoh R., Firdaus A., Anwar S., Osman M.Z.* Malware detection using static analysis in Android: a review of FeCO (features, classification, and obfuscation) // PeerJ Computer Science. – 2021. – Vol. 7. – Article ID: e522. – DOI 10.7717/peerj-cs.522.
17. *Kumar S., Janet B., Neelakantan S*. Identification of malware families using stacking of textural features and machine learning // Expert Systems with Applications. – 2022. – Vol. 204. – Article ID: 117635. – https://doi.org/10.1016/j.eswa.2022.118073.
18. *Lad S.S., Adamuthe A.C*. Improved deep learning model for static PE files malware detection and classification // International Journal of Computer Network and Information Security. – 2022. – Vol. 14, No. 2. – P. 14-26.
19. *Ravindra Babu S., Leisha R., Medows K.J.* Unveiling Powerful Machine Learning Strategies for Detecting Malware in Modern Digital Environment // Lecture Notes on Intelligent Computing and Data Science. – Springer, 2024. – Vol. 874. – P. 277-286. – ISBN978-3-031-50886-8. – DOI: 10.1007/978-3-031-50887-5_28.

20. VirusShare.com. A collection of malware samples for research purposes. – Режим доступа: https://virusshare.com/ (дата обращения: 26.05.2025).
21. *Cohen A., Nissim N., Rokach L., Elovici Y.* SFEM: Structural feature extraction methodology for the detection of malicious office documents using machine learning methods // Expert Systems with Applications. – 2016. – Vol. 64. – P. 324-338. – https://doi.org/10.1016/j.eswa.2016.07.010.
22. *Damaševičius R., Venčkauskas A., Toldinas J.* Ensemble-based classification using neural networks and machine learning models for Windows PE malware detection // Electronics. – 2021. – Vol. 10, No. 4. – Art. 485. – https://doi.org/10.3390/electronics10040485.
23. *Muralidharan T., Cohen A., Gerson N., Alazab M.* File packing from the malware perspective: Techniques, analysis approaches, and directions for enhancements // ACM Computing Surveys. – Vol. 55, No. 5. – Article 108. – https://doi.org/10.1145/3530810.
24. *Nie S., Zhu X., Xiong F., Zhang N.* Network learning and propagation dynamics analysis // Frontiers in Physics. – 2025. – Vol. 13. – Article ID: 1609957. – DOI: 10.3389/fphy.2025.1609957.
25. *Saxe J., Berlin K.* Deep neural network-based malware detection using two-dimensional binary program features // 10th International Conference on Malicious and Unwanted Software (MALWARE). – IEEE, 2015. – P. 11-20. – DOI: 10.1109/MALWARE.2015.7413680.
26. *Sahs J., Khan L.* A machine learning approach to Android malware detection // Published in 2012 European Intelligence and Security Informatics Conference. – IEEE, 2012. – P. 141-147. – DOI: 10.1109/EISIC.2012.34.
27. *Ucci D., Aniello L., Baldoni R.* Survey of machine learning techniques for malware analysis // Computers & Security. – 2019. – Vol. 81. – P. 123-147. – doi.org/10.1016/j.cose.2018.11.001.

REFERENCES

1. *Schultz M.G., Eskin E., Zadok E., Stolfo S.J.* Data mining methods for detection of new malicious executables, *Proc. IEEE Symp. Security and Privacy (S&P)*, 2001, pp. 38-49.
2. *Kuang H., Wang J., Li R., Feng C., & Zhang X.* Automated Data-Processing Function Identification Using Deep Neural Network, *IEEE Access*, 2020, Vol. 8, pp. 55411-55423. doi: 10.1109/ACCESS.2020.2981537.
3. *Ghanem K., Kherbache Z., Ourdighi O.* Enhancing Adversarial Examples for Evading Malware Detection Systems: A Memetic Algorithm Approach, *IJCNIS*, 2025, Vol. 17, No. 1, pp. 1-16. DOI: 10.5815/ijcnis.2025.01.01.
4. Microsoft. Microsoft Portable Executable and Common Object File Format Specification. Available at: https://learn.microsoft.com/ru-ru/windows/win32/ debug/pe-format (accessed 26 May 2025).
5. *Kozachok A.V., Matovykh S.S.* Strukturnaya model' faylov formata Portable Executable soderzhashchikh vredonosnyy kod [Structural model of Portable Executable files containing malicious code], *Problemy informatsionnoy bezopasnosti. Komp'yuternye sistemy* [Problems of information security. Computer systems], 2025, No. 2, pp. 41-59. DOI: 10.48612/jisp/pdu2-fvxz-g5d3.
6. *Rúa E.A., Bulut I.* Machine Learning-Based Secure Malware Detection Using Features from Binary Executable Headers, *European Symposium on Research in Computer Security*. Springer, 2025, pp. 204-216. DOI: 10.1007/978-3-031-82362-6_12.
7. *Al Balawi M., Alnabhan M.* Generative AI for Advanced Malware Detection, *4th Intelligent Systems Conference (IntelliSys)*. IEEE, 2024, pp. 204-216. DOI: 10.1109/ICSC63108.2024. 10895965.
8. *Petrean D.E., Potolea R., Oprisa C.* Packed Code Detection Using Shannon Entropy and Homomorphic Encrypted Executables, *Proceedings of the 20th International Conference on Intelligent Computer Communication and Processing*. IEEE, 2024, pp. 01-08. DOI: 10.1109/ICCP63557. 2024.10793050.
9. *Mahato A., Majumdar R., Ghosh S.K.* Feature-Driven Malware Detection using Cascade Machine Learning Models, *SN Computer Science*, 2025, Vol. 6, No. 7, pp. 794. Available at: https://doi.org/10.1007/s42979-025-04342-1.
10. *Alizada Adil and Ragab Hassen Hani*. Pextract: A Light-Weight Static Feature Extractor for Windows Portable Executable Files, *SSRN*, 2025. Available at: https://ssrn.com/abstract=5165659 (accessed 26 May 2025).
11. *Kumar S.S., Shetty J.* Malicious PE File Detection Using Machine Learning: An Analysis of Header Features, *COSMIC*. IEEE, 2024, pp. 66-71. DOI: 10.1109/ COSMIC63293.2024.10871898.
12. *Rizwan M., Ali E., Batool N.* Assessing Concept Drift in Malware: A Comprehensive Review and Analysis, *IBCAST*. IEEE, 2024, pp. 564-569. DOI: 10.1109/IBCAST61650. 2024.10876901.
13. *Schubert Kabban C.M., Graham S.R.* Malware Classification through Abstract Syntax Trees and L-moments, *Computers & Security*, 2025, Vol. 133, Article ID: 104082. DOI: 10.1016/j.cose.2024.104082.
14. *Canbek G., Temizel T.T., Sagiroglu S.* PToPI: A comprehensive review, analysis, and knowledge representation of binary classification performance measures/metrics, *SN Computer Science*, 2022, Vol. 4, Article No. 13. DOI: 10.1007/s42979-022-01409-1.

15. A survey of machine learning methods and challenges for Windows malware classification. Available at: https://arxiv.org/abs/2006.09271 (accessed 28 May 2025).
16. *Jusoh R., Firdaus A., Anwar S., Osman M.Z.* Malware detection using static analysis in Android: a review of FeCO (features, classification, and obfuscation), *PeerJ Computer Science*, 2021, Vol. 7, Article ID: e522. DOI 10.7717/peerj-cs.522.
17. *Kumar S., Janet B., Neelakantan S*. Identification of malware families using stacking of textural features and machine learning, *Expert Systems with Applications*, 2022, Vol. 204, Article ID: 117635. Available at: https://doi.org/10.1016/j.eswa.2022.118073.
18. *Lad S.S., Adamuthe A.C*. Improved deep learning model for static PE files malware detection and classification, *International Journal of Computer Network and Information Security*, 2022, Vol. 14, No. 2, pp. 14-26.
19. *Ravindra Babu S., Leisha R., Medows K.J.* Unveiling Powerful Machine Learning Strategies for Detecting Malware in Modern Digital Environment, *Lecture Notes on Intelligent Computing and Data Science*. Springer, 2024, Vol. 874, pp. 277-286. ISBN978-3-031-50886-8. DOI: 10.1007/978-3-031-50887-5_28.
20. VirusShare.com. A collection of malware samples for research purposes. Available at: https://virusshare.com/ (accessed 26 May 2025).
21. *Cohen A., Nissim N., Rokach L., Elovici Y.* SFEM: Structural feature extraction methodology for the detection of malicious office documents using machine learning methods, *Expert Systems with Applications*, 2016, Vol. 64, pp. 324-338. Available at: https://doi.org/10.1016/j.eswa.2016.07.010.
22. *Damaševičius R., Venčkauskas A., Toldinas J*. Ensemble-based classification using neural networks and machine learning models for Windows PE malware detection, *Electronics*, 2021, Vol. 10, No. 4, Art. 485. Available at: https://doi.org/10.3390/electronics10040485.
23. *Muralidharan T., Cohen A., Gerson N., Alazab M*. File packing from the malware perspective: Techniques, analysis approaches, and directions for enhancements, *ACM Computing Surveys*, Vol. 55, No. 5, Article 108. Available at: https://doi.org/10.1145/3530810.
24. *Nie S., Zhu X., Xiong F., Zhang N.* Network learning and propagation dynamics analysis, *Frontiers in Physics*, 2025, Vol. 13, Article ID: 1609957. DOI: 10.3389/fphy.2025.1609957.
25. *Saxe J., Berlin K.* Deep neural network-based malware detection using two-dimensional binary program features, *10th International Conference on Malicious and Unwanted Software (MALWARE)*. IEEE, 2015, pp. 11-20. DOI: 10.1109/MALWARE.2015.7413680.
26. *Sahs J., Khan L.* A machine learning approach to Android malware detection, *Published in 2012 European Intelligence and Security Informatics Conference*. IEEE, 2012, pp. 141-147. DOI: 10.1109/EISIC.2012.34.
27. *Ucci D., Aniello L., Baldoni R.* Survey of machine learning techniques for malware analysis, *Computers & Security*, 2019, Vol. 81, pp. 123-147. doi.org/10.1016/j.cose.2018.11.001.

Козачок Александр Васильевич – МИРЭА – Российский технологический университет; e-mail: totrin@mail.ru; г. Москва, Россия; д.т.н.; доцент; https://orcid.org/0000-0002-6501-2008.

Козачок Андрей Васильевич – МИРЭА – Российский технологический университет; e-mail: kozachok@mirea.ru; г. Москва, Россия; к.т.н.

Матовых Сергей Сергеевич – Академия Федеральной службы охраны Российской Федерации; e-mail: coolt88@gmail.com; г. Орёл, Россия; сотрудник; https://orcid.org/0009-0005-9693-3861.

Kozachok Alexander Vasilevich – MIREA – Russian Technological University; e-mail: totrin@mail.ru; Moscow, Russia; dr. of eng. sc.; associate professor; https://orcid.org/0000-0002-6501-2008.

Kozachok Andrey Vasilevich – MIREA – Russian Technological University; e-mail: kozachok@mirea.ru; Moscow, Russia; cand. of eng. sc.

Matovykh Sergei Sergeevich – The Academy of Federal Security Guard Service of the Russian Federation; e-mail: coolt88@gmail.com; Oryol, Russia; employee; https://orcid.org/0009-0005-9693-3861.