ARTICLE HISTORY: Received: April 13, 2022 Accepted: June 6, 2022 Published: June 13, 2022

УДК 004.056

ИСПОЛЬЗОВАНИЕ МЕТОДОВ МУЛЬТИКЛАССОВОЙ КЛАССИФИКАЦИИ ПРИ АНАЛИЗЕ ВРЕДОНОСНЫХ ПРОГРАММ

Крицулин Артур Андреевич
доцент кафедры,
Нефедов Владимир Сергеевич
доцент кафедры
МИРЭА – Российский Технологический университет
(Москва, Россия)

USING MULTI-CLASSIFICATION METHODS IN MALWARE ANALYSIS

Kriulin Artur Andreevich
associate professor,
Nefedov Vladimir Sergeevich
associate professor
MIREA - Russian Technological University
(Moscow, Russia)

Аннотация. В докладе приведен один из подходов анализа вредоносных программ с использованием алгоритмов машинного обучения и мультиклассовой классификации.

Abstract. The report presents one of the approaches to malware analysis using machine learning algorithms and multiclass classification.

Ключевые слова: вредоносные программы, машинное обучение, мультиклассовая классификация.

Keywords: malware, machine learning, multiclass classification.

Введение

Компьютерные атаки все больше принимают таргетированную форму, представляя собой совокупность различных действий в информационно-вычислительной сети с применением программных средств, в том числе вредоносных. Развитие современных методов интеллектуального анализа данных, алгоритмов машинного обучения позволяет уверенно совершенствовать различные системы поддержки принятия решений, распознавания образов, компьютерного моделирования и т.д., и в частности, систем обнаружения и предупреждения компьютерных атак. Совершенствование превентивных мер по распознаванию вредоносного программного обеспечения (ВПО) с определением класса таких программ должно способствовать максимально эффективному реагированию на инциденты компьютерной безопасности. В статье рассматривается подход к анализу ВПО с использованием мультиклассовой классификации алгоритмами машинного обучения с целью точного определения его класса. Подход отличается от существующих способом формирования классов вредоносных программ с применением синтаксического анализа результатов сканирования антивирусными средствами. Новизной подхода также является использование алгоритмов машинного обучения при определении класса вредоносный программы, что позволяет систематизировать и интегрировать полученные новые знания в процесс обеспечения компьютерной безопасности. Определение класса вредоносной программы позволит прогнозировать поведение таких программ в компьютерной сети, сужать места проверки и контроля, а также применять адекватные меры противодействия компьютерным атакам.

Выборка вредоносного программного обеспечения

Для проведения экспериментальных исследований использовалась выборка вредоносного программного обеспечения с публичного ресурса GitHub [1]. Из выборки были убраны все файлы, не являющиеся исполняемыми для операционной системы Windows. В результате анализу и классификации подвергались только файлы формата PE (Portable Executable) под платформы x86 и x64. Для определения класса вредоносных программ использовался сервис VirusTotal. Вредоносной программе рассчитывалась хэш-сумма и с использованием API сервиса VirusTotal были получены результаты сканирования этих программ средствами антивирусной защиты. Результаты сканирования от сервиса VirusTotal представляют собой ответы средств антивирусной защиты в формате JSON. В результате синтаксического анализа результатов сканирования были сформированы 11 классов вредоносных программ (рисунок 1).

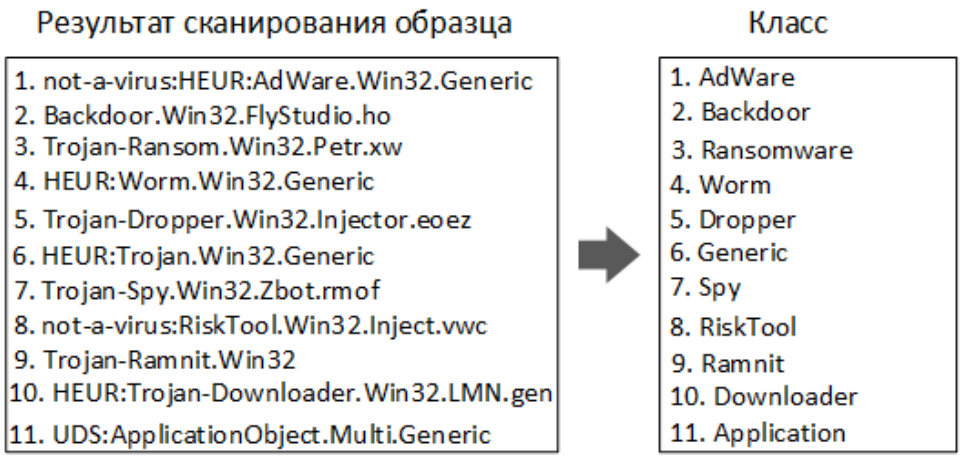

Рисунок 1 – Сопоставление образца вредоносной программы с классом

Отбор признаков для классификации вредоносного программного обеспечения

Обозначим признаком f для каждого экземпляра из всей выборки X вредоносных программ как характеристику исполняемого файла $f: X \to D_{<f>}$, где $D_{<f>}$ – вектор признаков. При этом элементы в векторе $D_{<f>}$ могут быть бинарными и принимать значения $\{0,1\}$ или количественными $\{0, +\infty\}$.

К количественным признакам относятся: размер исполняемого файла – f_1, число секций – f_2, версия компоновщика – f_3, размер образа в памяти операционной системы – f_4, размер исполняемого кода – f_5, размер инициализированных данных – f_6, число исполняемых секций – f_7, число секций с нестандартными именами – f_8, число импортируемых (экспортируемых) функций – f_9, число импортируемых библиотек – f_{10}, значение энтропии всего исполняемого файла – f_{11}.

Совокупность признаковых описаний [3] всех объектов выборки X вредоносных программ будет может быть представлена в матрицы F размера $m \times n$:

$$F = \|f_i(x_i)\|_{m \times n} = \begin{pmatrix} f_1(x_1) & f_n(x_1) \\ \dots & \dots\dots \\ f_1(x_m) & f_n(x_m) \end{pmatrix}, \quad (1)$$

которая отражает исходные данные.

На рисунке 2 представлены диаграммы размаха для разных классов программ по признаку f_1 размера файла из матрицы признаков F (1). На оси Y находятся значения размеров исполняемых файлов в байтах, на оси X классы вредоносных программ. На диаграммах в графическом виде отражаются такие показатели, как медиана, нижний и верхний квартили, минимальное и максимальное значение выборки и выбросы.

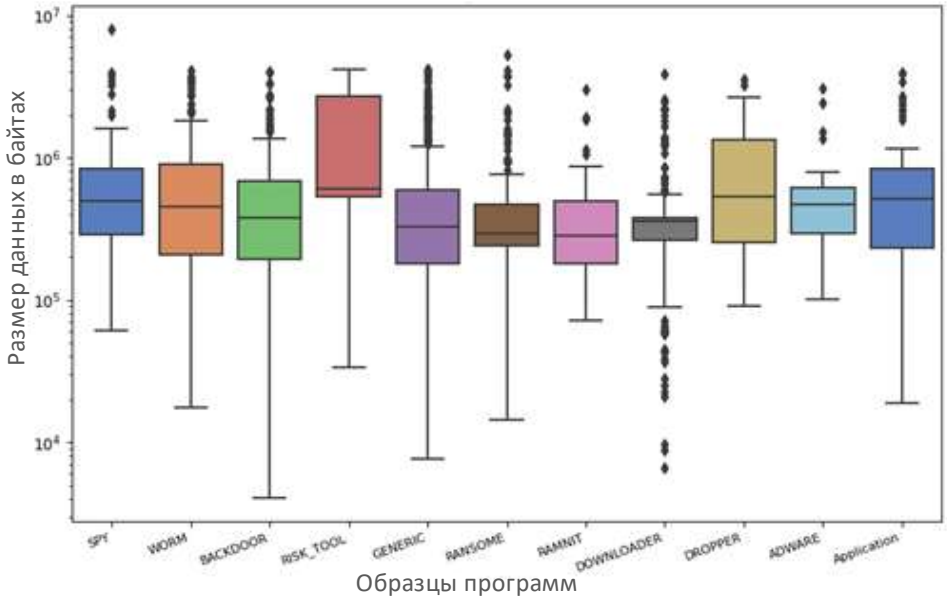

Рисунок 2 – Сравнение классов вредоносных программ по размеру

К бинарным признакам относятся: наличие упаковщика – f_{12}, тип исполняемого файла (библиотека динамического линкования, системный драйвер) – f_{13}, использование платформы .NET – f_{14}, наличие отладочной информации – f_{15}, тип запуск программы (консольный, графический) – f_{16}, наличие секции ресурсов – f_{17}, наличие ресурсов с типом RC_DATA – f_{18}, разработка на языке Delphi – f_{19}.

Порядок разработки мультиклассового классификатора при анализе вредоносного программного обеспечения

Непосредственно самому этапу разработки классификатора предшествует этап подготовки данных из выборки вредоносных программ. Первым шагом в подготовке данных выполняется факторизация. Признаки, имеющие строковые значения, например тип упаковщика или название компилятора кодируются числовыми эквивалентами. Затем выборка разбивается на обучающую и тестовую, как правило в соотношении 80 на 20, соответственно. Многие алгоритмы чувствительны к шкале данных. В тех случаях, когда размерности не сопоставимы, данные из обучающей и тестовой выборки шкалируют, благодаря чему каждая размерность имеет нулевое среднее значение и стандартное отклонение, равное 1. Преобразование каждой размерности в «стандартные отклонения от среднего значения» позволяет фактически избавиться от единиц измерения.

При решении задачи мультиклассовой классификации необходимо построить алгоритм, который будет предсказывать к какому классу, число которых *K*, относится каждый объект из выборки и оценить его точность. Для этого, как правило задача мультиклассовой классификации сводится к задаче бинарной классификации. Используя подход OneVsRest (один против всех) построим *K* классификаторов, каждый из которых будет отделять один класс от остальных. При этом для *i*-го класса обучающей выборкой будет служить вся выборка вредоносных программ, $X^i = X^l$. Если объект относится к классу *i*, ответ будет «1», если объект относится к одному из других классов ответ будет «0», $y_j^i = [y_j = i]$. Построим классификатор, который будет давать оценку принадлежности объекта к классу 1, причем ответами будут вещественные числа $b_i(x) \in \mathbb{R}$. После построения *K* классификаторов вычисляем вещественные оценки принадлежности, находим максимальную из них (2) и этот класс возвращаем в качестве ответа:

$$y(x) = argmax\ b_k(\mathrm{x}) \qquad (2)$$

$$k \in \{1, \ldots, K\}$$

В качестве метрик в задачах классификации используется матрица ошибок, которая будет содержать следующие ответы:

- истинно-положительный, *TP* – образец относится к конкретному классу, например, к классу WORM и алгоритм его классифицировал к классу WORM;

- ложно-отрицательный, *FN* – образец относится к классу WORM и алгоритм его классифицировал к другому классу;

- ложно-положительный, *FP* – образец не относится к классу WORM и алгоритм его классифицировал к классу WORM;

- истинно-отрицательный, *TN* – образец не относится к классу WORM и алгоритм его классифицировал к другому классу.

Для оценки качества работы алгоритма на каждом из классов по отдельности вводятся метрики точности и полноты:

$$P = \frac{TP}{TP + FP}; \quad R = \frac{TP}{TP + FN}. \tag{3}$$

Показатель *P* отражает долю положительно классифицированных вредоносных программ конкретного класса, а показатель *R* показывает какую долю вредоносных программ конкретного класса из всех программ этого класса нашел алгоритм.

Для оценки работы модели в целом, может использоваться также значение площади под кривой ошибок – ROC-AUC. ROC-кривая представляет собой линию со значениями от (0;0) до (0;1) в координатах *TPR* истинно-положительного результата и *FPR* ложно-положительного результата [2]:

- истинно-положительный результат, *TPR* – отношение числа *TP* (истинно-положительных) ответов от алгоритма к общему числу образцов класса WORM в выборке. Данный показатель отражает способность алгоритма распознавать образцы класса WORM при анализе вредоносных образцов различного класса.

- ложно-положительный результат, *FPR* – отношение числа *FP* (ложно-положительных) ответов от алгоритма к общему числу образцов других классов в выборке.

Для решения *K* задач бинарной классификации результирующая точность алгоритма может быть усреднена и оцениваться с использованием макро и микро среднего показателей:

$$P_{macro} = \frac{\sum_{k=1}^{N} P_k}{N}; \quad P_{micro} = \frac{\sum_{k=1}^{N} TP}{\sum_{k=1}^{N} (TP + FP)}. \tag{4}$$

При микроусреднении в каждой задаче бинарной классификации вычисляются базовые показатели: *TP, FN, FP, TN*, затем они усредняются также по каждой задаче и вычисляется результирующая метрика, например точность. При макроусреднении сначала выполняется вычисление результирующей метрики для каждой бинарной задачи, а затем производится усреднение.

Экспериментальные результаты мультиклассовой классификации вредоносного программного обеспечения

С целью более наглядной оценки качества классификации вредоносных программ использовались несколько алгоритмов, – случайный лес [4] и метод опорных векторов [5]. Также из предположения, что с увеличением числа классов вредоносных программ качество классификации должно снижаться, использовалось две выборки, состоящие из четырех классов и из одиннадцати.

В таблицах 1 и 2 представлены результаты классификации вредоносных программ алгоритмом случайный лес с использованием метрик *TP, FN, FP, TN*, а также рассчитанные на основе этих метрик показатели (3) точности, полноты и значения истинно-положительного и ложно-положительного результата.

Таблица 1

Показатели классификации алгоритма случайный лес для 4 классов ВПО

№ п.п	Класс	TP	FN	FP	TN	TPR	FPR	P
1.	WORM	138	10	10	43	0.81	0.19	0.81
2.	BACKDOOR	148	29	5	19	0.40	0.21	0.79
3.	RANSOME	143	22	3	33	0.60	0.08	0.92
4.	DOWNLOADER	153	10	3	35	0.78	0.08	0.92

Таблица 2

Показатели классификации алгоритма случайный лес для 11 классов ВПО

№ п.п	Класс	TP	FN	FP	TN	TPR	FPR	P
1.	SPY	456	13	1	2	0.13	0.33	0.67
2.	WORM	403	25	4	40	0.62	0.09	0.91
3.	BACKDOOR	417	44	5	6	0.12	0.45	0.55
4.	RISK_TOOL	458	11	1	2	0.15	0.33	0.67
5.	GENERIC	223	57	60	132	0.70	0.31	0.69
6.	RANSOME	410	29	8	25	0.46	0.24	0.76
7.	RAMNIT	462	8	0	2	0.20	0.00	1.00
8.	DOWNLOADER	411	19	8	34	0.64	0.19	0.81
9.	DROPPER	467	4	0	1	0.20	0.00	1.00
10.	ADWARE	464	4	3	1	0.20	0.75	0.25
11.	APPLICATION	457	11	2	2	0.15	0.50	0.50

Из сравнения значений точности классификации в таблицах 1 и 2 можно заметить снижение этого показателя для некоторых классов с увеличением общего числа исследуемых классов.

С использованием полученных показателей определим значения (4) микро и макро среднего показателей (таблица 3).

Таблица 3

Микро и макро средние показатели классификации ВПО

№ п.п	Алгоритм	Число классов	Микро среднее	Макро среднее
1.	Случайный лес	4	0,80	0,70
2.	Метод опорных векторов	4	0,66	0,66
3.	Случайный лес	11	0,75	0,65
4.	Метод опорных векторов	11	0,69	0,57

С использованием показателей *TPR* истинно-положительного и *FPR* ложно-положительного результата выполним оценку качества классификации обоих алгоритмов построив графики ROC-кривых для четырех и одиннадцати классов вредоносных программ.

На рисунке 3 (вверху) изображен график ROC-кривых для четырех классов вредоносных программ, классифицированных с использованием алгоритма случайный лес. Микросреднее и макросреднее значения для мультиклассовой классификации с использованием случайного леса получились 0,80 и 0,80 соответственно.

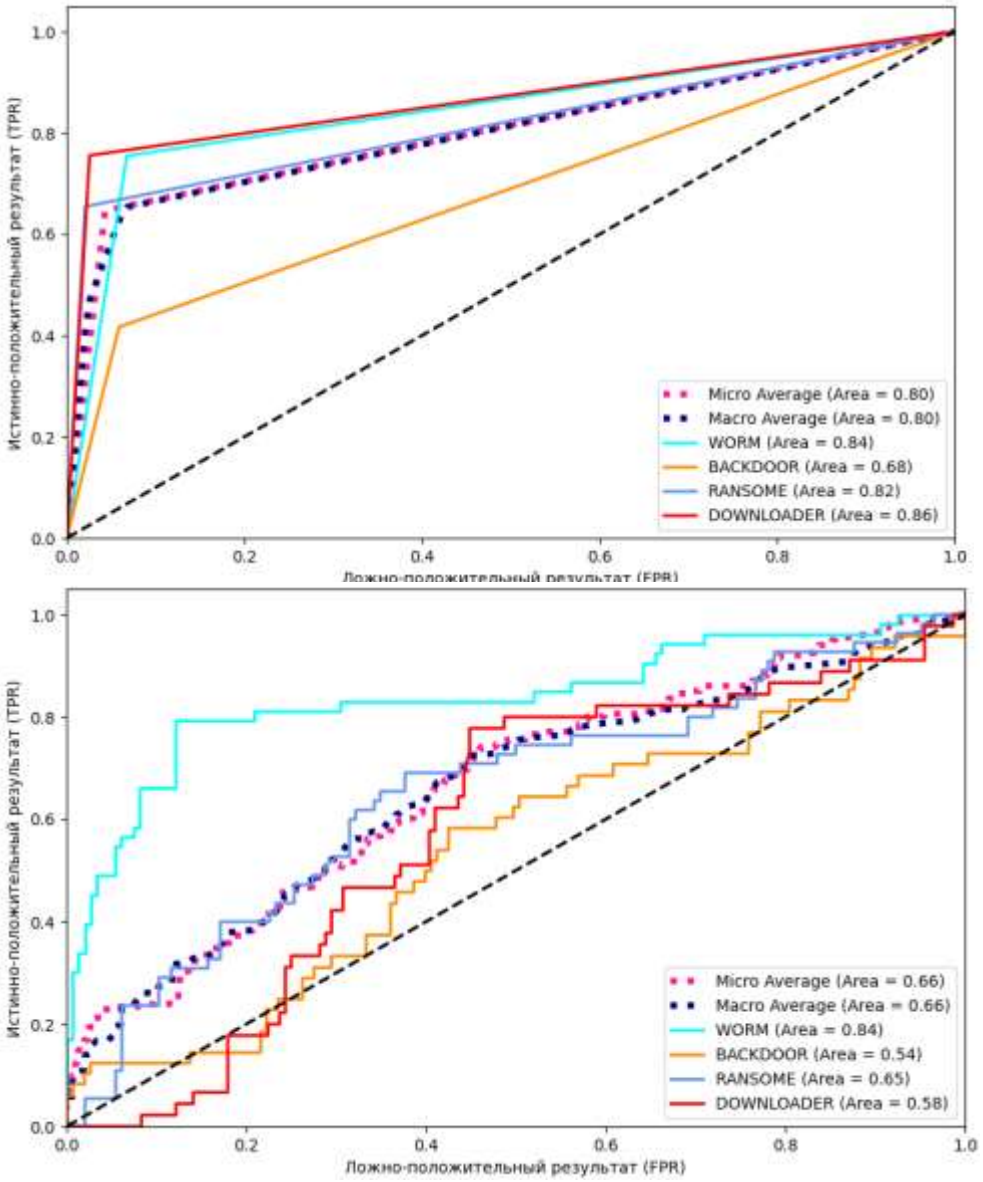

Рисунок 3 – Сравнение результатов мультиклассовой классификации

На рисунке 3 (внизу) изображен график ROC-кривых для четырех классов вредоносных программ, классифицированных с использованием метода опорных векторов. Микросреднее и макросреднее значения для мультиклассовой классификации с использованием метода опорных векторов получились 0,66 и 0,66 соответственно.

Стоит отметить, что результаты качества классификации для обоих алгоритмов оказались довольно высокими, усредненные показатели практически одинаковы. Интересным результатом оказался факт, что алгоритм случайный лес лучшие результаты показал на классах WORM (сетевые черви) и DOWNLOADER (программы, скачивающие нагрузку из интернета). Метод опорных векторов лучше распознает классы BACKDOOR (бэкдоры) и RANSOME (шифровальщики или уничтожители данных).

На рисунке 4 (вверху) изображен график ROC-кривых для всех имеющихся классов вредоносных программ, классифицированных с использованием алгоритма случайный лес. Микросреднее и макросреднее значения для мультиклассовой классификации с использованием случайного леса получились 0,75 и 0,65 соответственно.

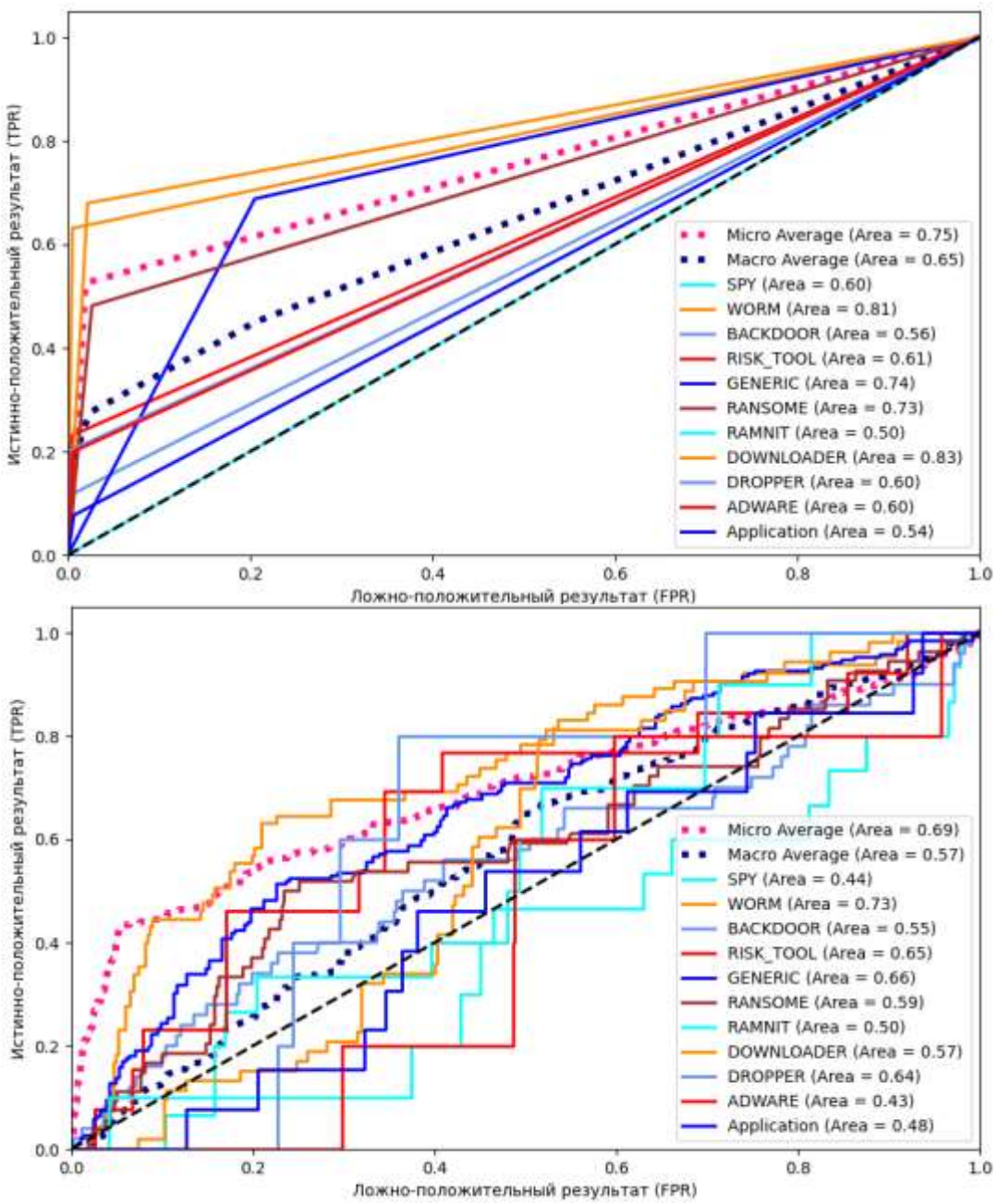

Рисунок 4 – Сравнение результатов мультиклассовой классификации

На рисунке 4 (внизу) изображен график ROC-кривых для всех имеющихся классов вредоносных программ, классифицированных с использованием метода опорных векторов. Микросреднее и макросреднее значения для мультиклассовой классификации с использованием метода опорных векторов получились 0,69 и 0,57 соответственно.

Все полученные результаты классификации одиннадцати классов вредоносных программ оказались ниже, чем для четырех классов, при этом алгоритм случайный лес практически по всем классам показал более высокую точность чем метод опорных векторов.

Заключение

Таким образом, в статье исследован подход к анализу вредоносных программ разных классов с использованием алгоритмов машинного обучения. В отличии от существующих подход использует усредненные макро и микро показатели, позволяющие в оценить точность работы алгоритмов для разных классов вредоносных программ. Полученные результаты отражают специфичность параметров вредоносных программ разных классов, что подтверждается точностью классификации алгоритмов машинного обучения.

Стоит отметить, что результаты классификации получены из анализа параметров и характеристик вредоносных программ не относящихся к динамическим и не учитывают поведение таких программ в операционной системе. С учетом того, что не всегда имеется возможность использования специальных песочниц для анализа программ, рассмотренный в статье подход может быть использован в общей системе предупреждения компьютерных атак в совокупности с другими подходами.

Дальнейшим направлением для развития предложенного подхода к анализу вредоносных программ с использованием мультиклассовой классификации может быть расширение области признакового пространства с учетом поведенческих особенностей. Также представляет интерес применение подхода при анализе вредоносных программ различных таргетированных угроз APT (Advanced Persistent Threat) с целью их идентификации при расследовании компьютерных инцидентов.

Список литературы

1. Репозиторий с образцами вредоносных программ
[Электронный ресурс]. – Режим доступа: https://github.com/ytisf/theZoo. (дата обращения: 18.10.2020).

2. Машинное обучение. [Электронный ресурс]. – Режим доступа: http://www.machinelearning.ru/wiki/images/f/fc/Voron-ML-Intro-slides.pdf. (дата обращения: 18.10.2020).

3. Применение методов машинного обучения к задачам обнаружения вредоносного программного обеспечения / И. В. Абашева, М. А. Еремеев, А. А. Криулин [и др.] // Труды Военно-космической академии имени А.Ф.Можайского. – 2020. – № 675. – С. 164-171.

4. Лекции по логическим алгоритмам классификации. [Электронный ресурс]. – Режим доступа: http://www.machinelearning.ru/wiki/images/3/3e/Voron-ML-Logic.pdf. (дата обращения: 18.10.2020).

4. Линейные методы классификации и регрессии: метод опорных векторов [Электронный ресурс]. – Режим доступа: http://www.machinelearning.ru/wiki/images/a/a0/Voron-ML-Lin-SVM.pdf. (дата обращения: 18.10.2020).

Сведение об авторах

Криулин Артур Андреевич – кандидат технических наук. Доцент кафедры прикладных информационных технологий. МИРЭА – Российский технологический университет (г. Москва). Область научных интересов: машинное обучение, исполняемые файлы, компьютерная криминалистика, безопасность операционных систем, обнаружение вредоносных программ.

Адрес: 107076, Россия, г. Москва, ул. Стромынка, д.20.

Нефедов Владимир Сергеевич – кандидат технических наук. Доцент кафедры прикладных информационных технологий. МИРЭА – Российский технологический университет (г. Москва). Область научных интересов: сетевая безопасность операционных систем, обнаружение вредоносных программ.

Адрес: 107076, Россия, г. Москва, ул. Стромынка, д.20.