ГЕНЕРАЦИЯ СИНТЕТИЧЕСКИХ ДАННЫХ ДЛЯ СИСТЕМ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА В ЗАДАЧЕ ОБНАРУЖЕНИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Стародубов М. И.[1], Боршевников А. Е.[2], Селин Н. А.[3]

DOI: 10.21681/2311-3456-2025-2-105-113

Цель работы: *разработка методики генерации синтетических данных для систем обнаружения вредоносного программного обеспечения.*

Метод исследования: *генерация синтетических данных при помощи методов обработки естественного языка (трансформера T5 и большой языковой модели Mistral 7b), изначально предназначенных для работы с текстовыми задачами.*

Полученный результат: *для проведения исследований различных объектов требуются большие наборы данных. Однако для сбора реальных данных требуется большое количество ресурсов. Один из способов решить указанную проблему – это генерация синтетических данных со схожими свойствами с реальными данными. В рамках данной работы исследовано применение архитектур нейронных сетей T5, Mistral 7b и их комбинации в задаче генерации синтетических образцов ПО. Статистически оценено, что генерируемые синтетические образцы в рамках посимвольной и биграммной модели ПО, соответствуют свойствам естественных объектов ПО. В рамках оценки триграммной модели установлено несоответствие свойств синтетических и естественных данных. Выдвинуто предположение о несоответствии статистических свойств n-граммных моделей ПО как следствие более сложной структуры исследуемых объектов. Практические эксперименты показывают, что синтетических данных, полученных при помощи T5 и Mistral 7b отдельно, недостаточно для использования в качестве обучающей выборки (наблюдается сильное снижение показателя F1-мера с 0,98 до 0,83). Использование объединённых данных приводит к результату, близкому к реальным данным (0,98 и 0,97).*

Практическая ценность *состоит в определении возможности использования синтетических данных для обучения моделей глубокого обучения, что позволяет расширить область нормального и аномального поведения в системах обнаружения аномалий.*

Ключевые слова: *вредоносное программное обеспечение, глубокое обучение, синтетические данные, T5, Mistral 7b, трансформеры, большая языковая модель, Ransomware, компьютерные атаки.*

Введение

Согласно отчёту компании Лаборатория Касперского[4] наибольшее число вредоносных программ направлено на финансы пользователей (Trojan-Spy), использование их вычислительных мощностей (Cryptojacking) и использование их данных с целью получения выкупа (Trojan-Ransom, программы-вымогатели). Наибольший рост модификаций вредоносного программного обеспечения наблюдается в последней категории вредоносных программ. В 2023 году раз в 8 дней появлялось новое семейство программ-вымогателей, а каждые 22 минуты появлялась модификация уже известных семейств[5].

Классических методов обнаружения, сигнатурного [1-2] и эвристического [3], недостаточно для обнаружения новых, ранее неизвестных вредоносных программ. Совершенно другим подходом является метод обнаружения, основанный на аномалиях, заключающийся в проверке вредоносности программы путем обнаружения разницы между поведением нормальной и аномальной программы [4-5]. Подход в обнаружении аномалий заключается в определении области, представляющей нормальное поведение, и объявления аномалией любого наблюдения, не соответствующего области нормального поведения.

1 Стародубов Максим Игоревич, аспирант ФГАОУ ВО «Дальневосточный федеральный университет» (ДВФУ), г. Владивосток, Россия. E-mail: starodubov.mi@dvfu.ru
2 Боршевников Алексей Евгеньевич, доцент департамента Информационной безопасности ФГАОУ ВО «Дальневосточный федеральный университет» (ДВФУ), г. Владивосток, Россия. E-mail: borshevnikov.ae@dvfu.ru
3 Селин Никита Александрович, студент департамента Информационной безопасности ФГАОУ ВО «Дальневосточный федеральный университет» (ДВФУ), г. Владивосток, Россия. E-mail: selin.na@dvfu.ru

4 Kaspersky Security Bulletin 2023. Statistics // Securelist by Kaspersky URL: https://securelist.com/ksb-2023-statistics/111156/
5 Kaspersky Security Bulletin 2023. Statistics // Kaspersky URL: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2023/11/28102415/KSB_statistics_2023_en.pdf

Этот подход кажется достаточно простым, однако, существует несколько факторов, значительно его усложняющих [6]:

1) сложность в определении нормальной области, охватывающей все возможные варианты нормального поведения;
2) злоумышленники часто маскируют аномальные наблюдения так, чтобы они выглядели нормальными;
3) во многих областях нормальное поведение объекта постоянно изменяется и развивается;
4) доступность помеченных данных для обучения/валидации моделей;
5) наличие шума.

Из-за этих факторов задачу обнаружения аномалий в общем виде решить непросто. Большинство существующих методов обнаружения аномалий решают конкретную постановку задачи.

Методы машинного обучения [7-8] и различные архитектуры нейронных сетей [9–19] позволяют достичь лучшие результаты эффективности в задаче обнаружения аномалий поведения программного обеспечения. Разбросы эффективности по трём показателям (Accuracy, precision, recall) в большинстве исследований достаточно большие, однако все держатся на уровне 90–100 %. Результаты исследований показывают [20], что глубокое обучение будет продолжать достигать улучшенных результатов по мере увеличения размера набора данных, в то время как другие методы машинного обучения выйдут на плато на каком-то относительно раннем этапе. Конечно, даже если это полностью верно, существуют практические вычислительные ограничения, поскольку для получения большего количества данных требуется больше вычислительной мощности. В связи со всем вышесказанным остро встаёт задача генерации синтетических данных.

Языковая модель ВПО

Наибольшее количество исследований вредоносного программного обеспечения направлено на исследование статических характеристик исполняемых объектов, в то время как методу динамического анализа уделено гораздо меньше внимания, хотя он и кажется более перспективным и эффективным. Метод динамического анализа позволяет рассматривать не только сам объект, но и его поведение. Под поведением подразумевается последовательность действий (из множества всех возможных действий) исполняемого объекта.

Наиболее используемым инструментом, способным описать поведение образца, является статистическая модель, описывающая вероятностное распределение всех возможных действий в поведении объекта. Однако у неё есть ряд недостатков, главными из которых являются невозможность определения важности последовательности действий относительно друг друга, выделения циклов действий и других более сложных концепций. В данной работе предлагается использовать языковую модель, которая позволяет определять сложные языковые конструкции.

Языковая модель – это вероятностная модель того, как генерировать предложения естественного языка, которая показывает, вероятность появления предложения. Для каждого предложения S, где $S = (w_1, \ldots, w_T)$ является последовательностью слов, языковая модель нацелена на оценку совместной вероятности входящих в него слов $P(w_1, \ldots, w_T)$.

$$P(w_1, \ldots, w_T) = \prod_{t=1}^{T} P(w_T | w_T, \ldots, w_{t-1}). \quad (1)$$

Таким образом, это эквивалентно оценке вероятности каждого слова в S с учетом его предыдущих слов, а именно того, какое слово может быть с учетом последовательности предшествующих слов.

Множество U - множество всех возможных программ.

$$U = \{R_1, \ldots, R_k, B_1, \ldots, B_m\}, \quad (2)$$

где $s = k + m$ – количество всех программ; k – количество всех программ, являющихся программой-вымогателем; m – количество программ, доверено не являющихся программой-вымогателем.

Множество U состоит из двух подмножеств $B = \{B_1, \ldots, B_m\}$ и $R = \{R_1, \ldots, R_k\}$.

Пусть множество $Api = \{api_1, \ldots, api_N\}$ – множество всех возможных api вызовов. Тогда каждый элемент элемент U_i из множества U определим следующим образом:

$$U_i = (api_{i,1}, \ldots, api_{i,T}). \quad (3)$$

Несложно заметить, что каждый элемент U_i из множества U можно представить в виде предложения S, если каждый объект w_i является элементом множества Api. Таким образом, любой элемент U_i имеет вид:

$$U_i = (api_{i,1}, \ldots, api_{i,T}) = (w_{i,1}, \ldots, w_{i,T}). \quad (4)$$

Следовательно, в этом случае возникает возможность использования языковой модели для оценки совместной вероятности входящих в него api.

$$P(api_{i,1}, \ldots, api_{i,T}) = \prod_{t=1}^{T} P(api_{i,t} | api_{i,T}, \ldots, api_{i,t-1}). \quad (5)$$

Поскольку $P(api_{i,t} | api_{i,T}, \ldots, api_{i,t-1})$ трудно оценить, часто используют «n-граммные модели» для её аппроксимации. n-грамма определяется как n последовательных слов и в нашем случае означает, что следующее слово api_t зависит только от предыдущих $n - 1$ слов.

$$P(api_{i,t} | api_{i,T}, \ldots, api_{i,t-1}) \cong P(api_{i,t} | api_{i,t-n+1}, \ldots, api_{i,t-1}) \quad (6)$$

Пусть $\varphi(U_i)$: $U_i \longrightarrow \{0,1\}$ – функционал, обозначающий выполнение программы U_i и приводящий либо к безопасному состоянию систем (0), либо не безопасному состоянию (1). Определим область вероятностей вредоносного поведения ПО, как:

$$V = \{U_i: U_i \in U, P(\varphi(U_i)) = 1\} =$$
$$= \{U_i: U_i \in U, P(\varphi(api_{i,t} | api_{i,t-n+1}, \ldots, api_{i,t-1}) = 1\} \quad (7)$$

Элемент $U_i \in U$ является элементом множества R тогда и только тогда, когда $U_i \in V$.

Элемент $U_i \in U$ является элементом множества B тогда и только тогда, когда $U_i \notin V$, т.е. не является элементом области V.

Таким образом, задача генерации элементов множеств $B = \{B_1, \ldots, B_m\}$ и $R = \{R_1, \ldots, R_k\}$ сводится к задаче генерации предложений S из слов множества Api длины n. А задача классификации элементов U_i в множества B и R сводится к задаче разделения вероятностей $P(api_{i,t} | api_{i.T}, \ldots, api_{i,t-1})$.

Набор реальных данных

В качестве набора реальных данных по представленной на рисунке 1 схеме были сформированы 600 000 поведенческих отчётов исполняемых файлов (300 000 вредоносных и 300 000 чистых).

В качестве среды для исследования поведения объектов была выбрана следующая конфигурация – Cuckoo Sandbox, продукт виртуализации Oracle VM VirtualBox с виртуальной машиной. Операционной системой была выбрана Windows 10 с предустановленными библиотеками, необходимыми для исполняемых файлов. Исходя из выбранной конфигурации, множество Api состоит из 4000 слов, соответственно и тексты состоят из 4000 различных токенов.

LLM модели

Основанные на архитектуре трансформеров большие языковые модели (LLM) способствовали значительному прогрессу в области обработки естественного языка. Использование двунаправленных контекстов в модели BERT [21] привело к повышению производительности в широком спектре задач. Появление моделей серии OpenAI GPT позволило пересмотреть подходы к обработке естественного языка. Эти модели демонстрируют замечательное мастерство в задаче генерации высококачественного текста, похожего на человеческий [22], демонстрации возможностей в элементарном мышлении [23], переводе [24], генерации научных данных [25] и генерации кода [26]. Несмотря на активное использование больших языковых моделей, исследований, связанных с генерацией синтетических данных для задачи обнаружения ВПО, проведено недостаточно, и все они не связаны с большими языковыми моделями.

В качестве генератора была выбрана модель «Mistral 7b», имеющая 7,3 миллиарда параметров. С ее помощью было сформировано 600 000 синтетических отчётов (по 300 000 из каждого класса).

Модель преобразования текста в текст T5

Модель преобразования текста в текст (T5) [27] очень похожа на модель кодера-декодера на основе трансформера. Схема модели представлена на рисунке 2. Каждый блок кодировщика состоит из уровня самообучения и нейронной сети с прямой связью. Каждый блок декодера состоит из уровня самообучения, уровня внимания кодировщика-декодировщика и нейронной сети с прямой связью. Однако существуют незначительные различия между T5 и моделью кодер-декодер на основе трансформера. Например, применяется нормализация уровней между

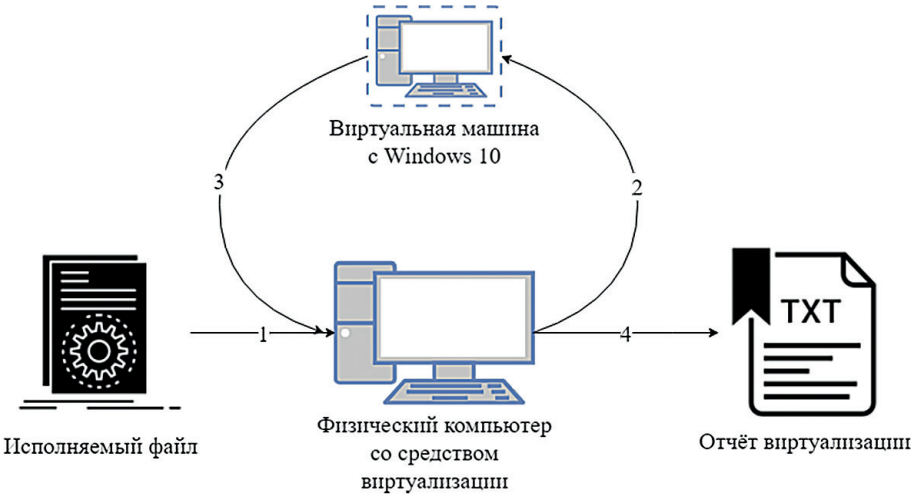

Рис. 1. Получение поведенческих отчётов

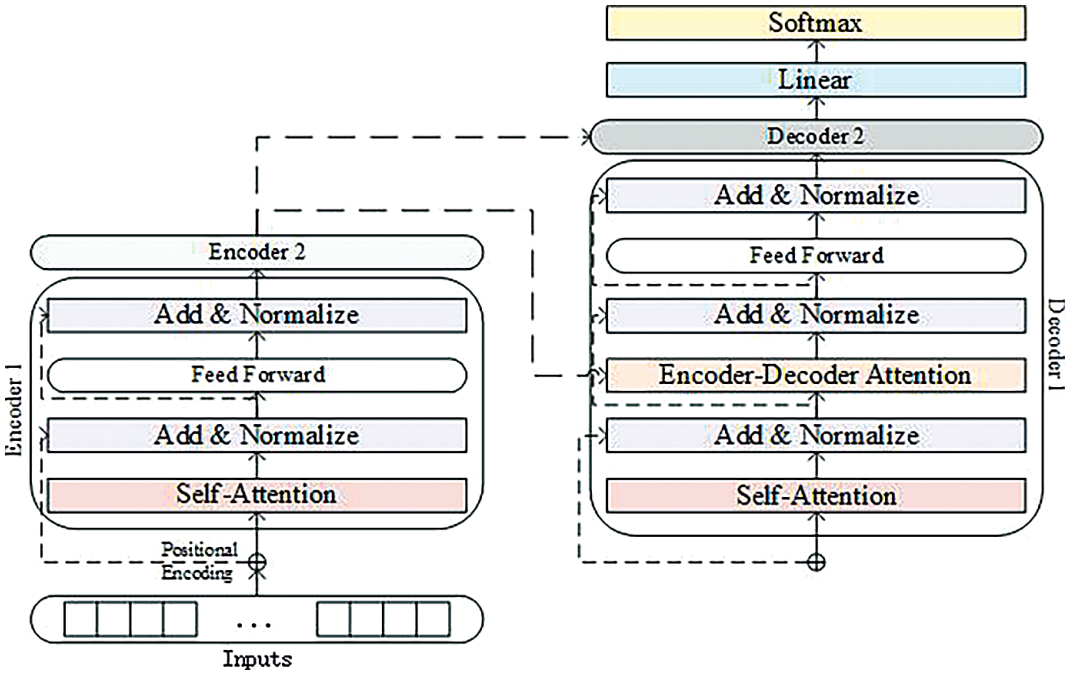

Рис. 2. Схема модели T5

компонентами каждого блока кодера и каждого блока декодера. По сравнению с BERT, добавление блока декодирования позволяет T5 генерировать выходные данные, представляющие собой последовательности текста. T5 предварительно обучается с помощью самоконтроля с помощью учебной задачи.

В качестве данных для обучения были выбраны реальные данные, собранные ранее и размеченные в 2 класса (вредоносный/чистый), а в качестве учебной задачи была поставлена следующая: генерация новых поведенческих отчётов, относящихся к определённому классу.

В результате при помощи T5 было также сформировано 600 000 синтетических отчётов (по 300 000 из каждого класса).

Статистические особенности полученных данных

Перед тем как проводить практические эксперименты, стоит оценить, насколько сгенерированные данные совпадают с реальными. В этом могут помочь статистические методы.

Пусть N_i i-й набор данных. Для каждого набора данных можно провести n-граммный анализ, согласно которому будет построен вектор, компоненты которого являются вероятностями каждой n-граммы: $N = (P(n_1), \ldots, P(n_c))$.

Оценим согласованность двух наборов: естественного набора данных (N_2) и синтетических наборов (N_1), сгенерированных при помощи различных архитектур нейронных сетей, используя двухвыборочный критерий согласия Пирсона:

Таблица 1.

Посимвольный анализ

Набор данных	Рассчитанный критерий согласия Пирсона	Сравнение с табличным значением χ^2_{4000-1}, больше/меньше/равно						
		Уровень значимости						
		0, 001	0,01	0,05	...	0,95	0,99	0,999
T5 (R)	298264987	Больше			...	Больше		
Mistral 7b (R)	41229694							
T5 + Mistral 7b (R)	114891312							
T5 (B)	270394494							
Mistral 7b (B)	30833235							
T5 + Mistral 7b (B)	105949274							

Таблица 2.

Биграммный анализ

Набор данных	Рассчитанный критерий согласия Пирсона	Сравнение с табличным значением $\chi^2_{4000^2-1}$, больше/меньше/равно						
		Уровень значимости						
		0, 001	0,01	0,05	...	0,95	0,99	0,999
T5 (R)	352776870	Больше			...	Больше		
Mistral 7b (R)	83667332							
T5 + Mistral 7b (R)	181681248							
T5 (B)	290970175							
Mistral 7b (B)	54706258							
T5 + Mistral 7b (B)	363502252							

Таблица 3.

Триграммный анализ

Набор данных	Рассчитанный критерий согласия Пирсона	Сравнение с табличным значением $\chi^2_{4000^3-1}$, больше/меньше/равно						
		Уровень значимости						
		0, 001	0,01	0,05	...	0,95	0,99	0,999
T5 (R)	184244596	Меньше			...	Меньше		
Mistral 7b (R)	95309577							
T5 + Mistral 7b (R)	155399099							
T5 (B)	125047216							
Mistral 7b (B)	60833159							
T5 + Mistral 7b (B)	1254157629							

$$\chi^2_{C-1} = mk * \sum_{i=1}^{C} \frac{(P_1(n_{,i}) - P_2(n_{,i}))^2}{|n_{1,i}| + |n_{2,i}|}, \quad (8)$$

где $C = (|Api|)^n$ – число всех возможных n-грамм, $P_j(n_i)$ – вероятность i-й n-граммы в j-м наборе данных, $|n_{j,i}|$ – численное количество i-й n-граммы в j-м наборе данных, m – общее число n-грамм в первом наборе данных, k – общее число n-грамм во втором наборе данных.

Указанные в таблицах 1, 2 и 3 результаты показывают, что синтетические данные, полученные при использовании различных алгоритмов в рамках посимвольной и биграммной модели ВПО, обладают схожими статистическими свойствами со схожими естественными данными с уровнем значимости 0,001. В то же время для триграммной модели (теоретически и для n-граммных моделей, где $n \geq 3$) согласованность естественных и синтетических данных не выполняется, что может говорить о более сложной структуре исследуемых объектов.

Практическое сравнение

В качестве системы обнаружения вредоносного программного обеспечения была выбрана система, схема которой представлена на рисунке 3 [28].

В качестве бинарного классификатора используется искусственная однослойная нейронная сеть (входной слой 768 нейронов, скрытый слой 768 нейронов и выходной слой размером 1 нейрон).

Было проведено 4 разных эксперимента, в которых этапы тонкой настройки выполнялись на разных наборах данных, полученных ранее (600 000 реальных поведенческих отчётов, 600 000 синтетических, полученных при помощи T5, и 600 000 синтетических, полученных при помощи «Mistral 7b», и 1 200 000 объединённых синтетических отчётов). Для эксперимента с реальными данными обучающие, валидационные и тестовые наборы распределяются как 50 %, 20 % и 30 % соответственно. Для экспериментов с синтетическими данными они были разделены на обучающие и валидационные в пропорции 4 к 1,

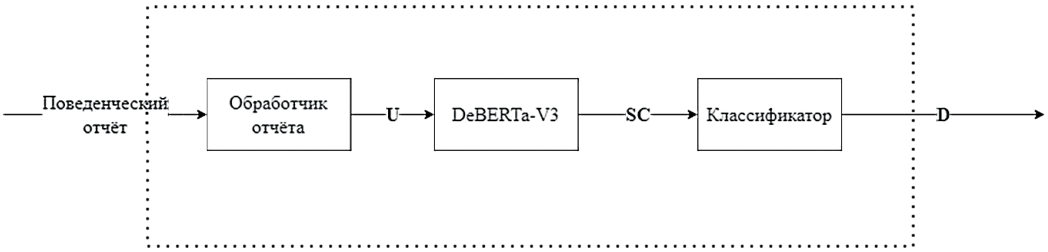

Рис. 3. Схема системы обнаружения ВПО

а в качестве тестового набора был выбран полный набор реальных данных.

Обучающие, валидационные и тестовые наборы стратифицированы, что означает, что каждый набор имеет такое же соотношение вредоносного и чистого ПО, как и весь набор данных.

В качестве параметров обучения были выбраны следующие: 5 эпох, размер пакета (batch size): 32, функция оптимизатора: «Adam» [29], скорость обучения (learning rate): $3e^{-7}$. Все они были тщательно подобраны, чтобы обеспечить наилучшую производительность модели.

Все эксперименты выполнялись на двух графических процессорах NVIDIA GeForce RTX 3090 Ti с суммарным объёмом видеопамяти 48 ГБ. Каждый полный эксперимент выполнялся десять раз с использованием разных исходных данных (т.е. разной последовательности для обучающих/валидационных/тестовых наборов), чтобы получить среднее значение производительности.

Метрики оценки

В качестве мер оценок могут быть выбраны следующие характеристики:

- ✓ TP (True Positives) — количество верных положительных классификаций;
- ✓ FP (False Positives) — количество ложноположительных обнаружений;
- ✓ FN (False Negatives) — количество положительных оценок, которые были ошибочно классифицированы как отрицательные TN;
- ✓ TN (True Negatives) — количество правильных отклонений;
- ✓ Precision – показатель, оценивающий отношение числа верно классифицируемых объектов, как положительных, к общему числу положительно распознанных объектов, правильно и неправильно;

$$\text{Precision} = \frac{TP}{TP + FP} \quad (9)$$

- ✓ Recall – показатель, оценивающий общее отношение числа верно классифицируемых объектов к общему числу объектов в кластере;

$$\text{Recall} = \frac{TP}{TP + FN} \quad (10)$$

- ✓ F1-мера – агрегированный показатель, объединяющий как precision, так и recall;

$$F_1 = 2\,\frac{precision * recall}{precision + recall} \quad (11)$$

Показатель F1-мера позволяет наиболее объективно оценивать результаты и именно он был выбран в качестве меры оценки.

Результаты

Результаты экспериментов, описанных выше, представлены в таблице 4. Как видно, синтетических данных, полученных при помощи T5 и Mistral 7b отдельно, недостаточно для использования в качестве обучающей выборки. Однако использование объединённых данных приводит к результату, близкому к реальным данным, что совпадает со статистическим анализом. При этом возможно использовать только синтетические данные для обучения моделей глубокого обучения, что позволит расширить область нормального и аномального поведения в системах обнаружения аномалий.

Таблица 4.

Результаты экспериментов

Набор	Объём выборки, шт.	F1 мера
Реальные данные	600 000	0.98
T5	600 000	0.83
Mistral 7b	600 000	0.85
T5 + Mistral 7b	1 200 000	0.97

Заключение

В рамках данной работы был собран один набор реальных данных и сгенерировано 3 набора синтетических данных (T5, Mistral 7b, T5 + Mistral 7b). Согласно статистическим особенностям наилучшие результаты при сравнении одного класса достигаются у синтетических данных, созданных при помощи трансформера T5, а наилучшие данные при сравнении классов между собой были сгенерированы большой языковой моделью «Mistral 7b». Однако, согласно практическому эксперименту, синтетических данных, полученных при помощи T5 и Mistral 7b отдельно, недостаточно для использования в качестве обучающей выборки (наблюдается сильное снижение показателя F1-мера с 0,98 до 0,83). Использование объединённых данных приводит к результату, близкому к реальным данным (0,98 и 0,97), что совпадает со статистическим анализом.

Исследование проведено при финансовой поддержке Минобрнауки России («Грант ИБ МТУСИ») № 40469-25-23-К.

Литература

1. Лапсарь А.П., Назарян С.А., Владимирова А.И. Повышение устойчивости объектов критической информационной инфраструктуры к целевым компьютерным атакам // Вопросы кибербезопасности. 2022. №. 2 (48). С. 39–51. DOI:10.21681/2311-3456-2022-2-39-51.
2. Punyasiri D. L.S. Signature & Behavior Based Malware Detection. 2023. DOI:10.13140/RG.2.2.22127.20640.
3. Yunmar R.A. et al. Hybrid Android Malware Detection: A Review of Heuristic-Based Approach // IEEE Access. 2024. Т. 12. С. 41255-41286. DOI:10.1109/ACCESS.2024.3377658.
4. Antić J. et al. Runtime security monitoring by an interplay between rule matching and deep learning-based anomaly detection on logs // 2023 19th International Conference on the Design of Reliable Communication Networks (DRCN). IEEE, 2023. С. 1–5. DOI: 10.1109/DRCN57075.2023.10108105.
5. Mushtaq E., Zameer A., Nasir R. Knacks of a hybrid anomaly detection model using deep auto-encoder driven gated recurrent unit // Computer Networks. 2023. Т. 226. С. 109681. DOI: 10.1016/j.comnet.2023.109681.
6. Sharma P. et al. A comparative analysis of malware anomaly detection // Advances in Computer, Communication and Computational Sciences: Proceedings of IC4S 2019. Springer Singapore, 2021. С. 35–44. DOI:10.1007/978-981-15-4409-5_3.
7. Liu C. et al. MOBIPCR: Efficient, accurate, and strict ML-based mobile malware detection // Future Generation Computer Systems. 2023. Т. 144. С. 140–150. DOI: 10.1016/j.future.2023.02.014.
8. Akhtar M.S., Feng T. Evaluation of machine learning algorithms for malware detection // Sensors. 2023. Т. 23. №. 2. С. 946. DOI: 10.3390/s23020946.
9. Basole S., Di Troia F., Stamp M. Multifamily malware models // Journal of Computer Virology and Hacking Techniques. 2020. Т. 16. С. 79–92. DOI:10.48550/arXiv.2207.00620.
10. Dhanya K.A. et al. Obfuscated Malware Detection in IoT Android Applications Using Markov Images and CNN // IEEE Systems Journal. 2023. Vol. 17, No. 2, pp. 2756–2766. DOI: 10.1109/JSYST.2023.3238678.
11. Ullah F. et al. NMal-Droid: network-based android malware detection system using transfer learning and CNN-BiGRU ensemble // Wireless Networks. 2023. Vol. 30. P. 6177–6198. DOI: 10.1007/s11276-023-03414-5.
12. Jahromi A.N. et al. An improved two-hidden-layer extreme learning machine for malware hunting // Computers & Security. 2020. Т. 89. С. 101655. DOI: 10.1016/j.cose.2019.101655.
13. Reddy V.S. K. et al. MDC-Net: Intelligent Malware Detection and Classification using Extreme Learning Machine // 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS). IEEE, 2023. С. 1590–1594. DOI:10.1109/ICAIS56108.2023.10073874.
14. Bhardwaj S., Dave M. Integrating a Rule-Based Approach to Malware Detection with an LSTM-Based Feature Selection Technique // SN Computer Science. 2023. Т. 4. С. 737. DOI: 10.1007/s42979-023-02177-2.
15. Devi R.A., Arunachalam A.R. Enhancement of IoT device security using an Improved Elliptic Curve Cryptography algorithm and malware detection utilizing deep LSTM // High-Confidence Computing. 2023. Т. 3. №. 2. С. 100117. DOI:10.1016/j.hcc.2023.100117.
16. Al-Khater W., Al-Madeed S. Using 3D-VGG-16 and 3D-Resnet-18 deep learning models and FABEMD techniques in the detection of malware // Alexandria Engineering Journal. 2024. V. 89. P. 39–52. DOI: 10.1016/j.aej.2023.12.061.
17. Камалов Б.Р., Тумбинская М.В. Программное обеспечение обнаружения «скрытых майнеров» в браузерной среде // Прикладная информатика. 2023. Т. 18. №. 1. С. 96–110. DOI: 10.37791/2687-0649-2023-18-1-96-110.
18. Warmsley D. et al. A Survey of Explainable Graph Neural Networks for Cyber Malware Analysis // 2022 IEEE International Conference on Big Data (Big Data). IEEE, 2022. С. 2932–2939. DOI:10.1109/BigData55660.2022.10020943.
19. Hu W., Tan Y. Generating adversarial malware examples for black-box attacks based on GAN // Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, November 21–24, 2022, Proceedings, Part II. Singapore : Springer Nature Singapore, 2023. С. 409–423. DOI: 10.1007/978-981-19-8991-9_29.
20. Козак Е. Обучение нейронных сетей и его значение для развития программной инженерии // Современная наука: актуальные проблемы теории и практики. Серия: естественные и технические науки, 2021, №. 8, С. 68–71. – DOI 10.37882/2223-2966.2021.08.16.
21. Devlin J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding // arXiv preprint arXiv:1810.04805. 2018.
22. Fernandez P. «Through the looking glass: envisioning new library technologies» AI-text generators as explained by ChatGPT // Library hi tech news. 2023. Т. 40. №. 3. С. 11–14. DOI: 10.1108/LHTN-02-2023-0017.

23. Hinton M., Wagemans J.H.M. How persuasive is AI-generated argumentation? An analysis of the quality of an argumentative text produced by the GPT-3 AI text generator // Argument & Computation. 2023. Т. 14. №. 1. С. 59–74. DOI:10.3233/AAC-210026.
24. Колин К.К. и др. Искусственный интеллект в технологиях машинного перевода // Социальные новации и социальные науки. 2021. №. 2 (4). С. 64–80. DOI: 10.31249/snsn/2021.02.05.
25. Ge J., Lai J. C. Artificial intelligence-based text generators in hepatology: ChatGPT is just the beginning // Hepatology communications. 2023. Т. 7. №. 4. С. e0097. DOI: 10.1097/HC9.0000000000000097.
26. Козачок А.В., Спирин А.А., Ерохина Н.С. Метод генерации семантически корректного кода для фаззингетестирования интерпретаторов javascript // Вопросы кибербезопасности. 2023. №. 5. С. 80–88. DOI: 10.21681/2311-3456-2023-5-80-88.
27. Raffel C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer // Journal of machine learning research. 2020. Т. 21. №. 140. С. 1–67. DOI:10.48550/arXiv:1910.10683.
28. Стародубов М.И., Артемьева И.Л., Селин Н.А. Метод обнаружения программ-вымогателей на основе анализа поведенческого отчета исполняемого объекта // Вопросы кибербезопасности. 2024. №. 3. С. 85-89. DOI: 10.21681/2311-3456-2024-3-85-89.
29. Bera S., Shrivastava V.K. Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification //International Journal of Remote Sensing. 2020. Т. 41. №. 7. С. 2664–2683. DOI:10.1080/01431161.2019.1694725.

METHODOLOGY OF GENERATING SYNTHETIC DATA FOR INTELLIGENT ANALYSIS SYSTEMS IN THE PROBLEM OF MALWARE DETECTION

Starodubov M.I.[6], Borshevnikov A.E.[7], Selin N.A.[8]

Keywords: *malware, deep learning, T5, Mistral 7b, transformers, large language model, ransomware, computer attacks.*

The aim of the work *is to develop a methodology for generating synthetic data for malware detection systems.*

The research method *is synthetic data generation using natural language processing methods (T5 transformer and large language model Mistral 7b), originally designed to work with text problems.*

The result obtained: *large datasets are required to expand the range of normal and abnormal behavior. Collecting real data requires a large amount of resources. In this work, one real dataset was collected and 3 synthetic datasets were generated (T5, Mistral 7b, T5 + Mistral 7b). Statistical analysis of the data shows that in most cases the combined dataset (T5 + Mistral 7b) achieves the best results, which is confirmed by a practical experiment. Synthetic data obtained using T5 and Mistral 7b separately are not enough to be used as a training set (a strong decrease in the F1-measure from 0,98 to 0,83 is observed). Using the combined data leads to a result close to the real data (0,98 and 0,97).*

The scientific novelty *consists in determining the possibility of using only synthetic data to train deep learning models, which allows expanding the scope of normal and abnormal behavior in anomaly detection systems.*

References

1. Lapsar' A.P., Nazarjan S.A., Vladimirova A.I. Povyshenie ustojchivosti objektov kriticheskoj informacionnoj infrastruktury k celevym komp'juternym atakam // Voprosy kiberbezopasnosti [Cybersecurity issues], 2022, No 2 (48), pp. 39–51. DOI:10.21681/2311-3456-2022-2-39-51.
2. Punyasiri D.L.S. Signature & Behavior Based Malware Detection, 2023. DOI:10.13140/RG.2.2.22127.20640.
3. Yunmar R.A. et al. Hybrid Android Malware Detection: A Review of Heuristic-Based Approach // IEEE Access, 2024, V. 12, pp. 41255-41286. DOI:10.1109/ACCESS.2024.3377658.
4. Antić J. et al. Runtime security monitoring by an interplay between rule matching and deep learning-based anomaly detection on logs // 2023 19th International Conference on the Design of Reliable Communication Networks (DRCN), IEEE, 2023, pp. 1–5. DOI:10.1109/DRCN57075.2023.10108105.
5. Mushtaq E., Zameer A., Nasir R. Knacks of a hybrid anomaly detection model using deep auto-encoder driven gated recurrent unit // Computer Networks, 2023, V. 226, pp. 109681. DOI: 10.1016/j.comnet.2023.109681.
6. Sharma P. et al. A comparative analysis of malware anomaly detection // Advances in Computer, Communication and Computational Sciences: Proceedings of IC4S 2019. Springer Singapore, 2021, pp. 35–44. DOI:10.1007/978-981-15-4409-5_3.
7. Liu C. et al. MOBIPCR: Efficient, accurate, and strict ML-based mobile malware detection // Future Generation Computer Systems, 2023, V. 144, pp. 140–150. DOI: 10.1016/j.future.2023.02.014.
8. Akhtar M.S., Feng T. Evaluation of machine learning algorithms for malware detection // Sensors. 2023, V. 23, No. 2, pp. 946. DOI: 10.3390/s23020946.

6 Maxim I. Starodubov, Ph.D. student, Far Eastern Federal University (FEFU), Vladivostok, Russia. E-mail: starodubov.mi@dvfu.ru
7 Alexey E. Borshevnikov, Associate Professor of the Department of Information Security of the Far Eastern Federal University (FEFU), Vladivostok, Russia. E-mail: borshevnikov.ae@dvfu.ru
8 Nikita A. Selin, student of the Information Security Department of the Far Eastern Federal University (FEFU), Vladivostok, Russia. E-mail: selin.na@dvfu.ru

9. Basole S., Di Troia F., Stamp M. Multifamily malware models // Journal of Computer Virology and Hacking Techniques, 2020, V. 16, pp. 79–92. DOI:10.48550/arXiv.2207.00620.
10. Dhanya K. A. et al. Obfuscated Malware Detection in IoT Android Applications Using Markov Images and CNN // IEEE Systems Journal, 2023. Vol. 17, No. 2, pp. 2756–2766. DOI: 10.1109/JSYST.2023.3238678.
11. Ullah F. et al. NMal-Droid: network-based android malware detection system using transfer learning and CNN-BiGRU ensemble // Wireless Networks, 2023, Vol. 30. P. 6177–6198. DOI: 10.1007/s11276-023-03414-5.
12. Jahromi A. N. et al. An improved two-hidden-layer extreme learning machine for malware hunting // Computers & Security, 2020, V. 89, pp. 101655. DOI: 10.1016/j.cose.2019.101655.
13. Reddy V. S. K. et al. MDC-Net: Intelligent Malware Detection and Classification using Extreme Learning Machine // 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), IEEE, 2023, pp. 1590–1594. DOI:10.1109/ICAIS56108.2023.10073874.
14. Bhardwaj S., Dave M. Integrating a Rule-Based Approach to Malware Detection with an LSTM-Based Feature Selection Technique // SN Computer Science, 2023, V. 4, pp. 737. DOI: 10.1007/s42979-023-02177-2.
15. Devi R. A., Arunachalam A. R. Enhancement of IoT device security using an Improved Elliptic Curve Cryptography algorithm and malware detection utilizing deep LSTM // High-Confidence Computing, 2023, V. 3, No 2, pp. 100117. DOI:10.1016/j.hcc.2023.100117.
16. Al-Khater W., Al-Madeed S. Using 3D-VGG-16 and 3D-Resnet-18 deep learning models and FABEMD techniques in the detection of malware // Alexandria Engineering Journal, 2024, V. 89, P. 39–52. DOI: 10.1016/j.aej.2023.12.061.
17. Kamalov B. R., Tumbinskaja M. V. Programmnoe obespechenie obnaruzhenija «skrytyh majnerov» v brauzernoj srede // Prikladnaja informatika [Applied Computer Science], 2023, V. 18, No 1, pp. 96–110. DOI: 10.37791/2687-0649-2023-18-1-96-110.
18. Warmsley D. et al. A Survey of Explainable Graph Neural Networks for Cyber Malware Analysis // 2022 IEEE International Conference on Big Data (Big Data), IEEE, 2022, pp. 2932–2939. DOI:10.1109/BigData55660.2022.10020943.
19. Hu W., Tan Y. Generating adversarial malware examples for black-box attacks based on GAN // Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, November 21–24, 2022, Proceedings, Part II. Singapore: Springer Nature Singapore, 2023. pp. 409–423. DOI: 10.1007/978-981-19-8991-9_29.
20. Kozak E. Obuchenie nejronnyh setej i ego znachenie dlja razvitija programmnoj inzhenerii // Sovremennaja nauka: aktual'nye problemy teorii i praktiki. Serija: estestvennye i tehnicheskie nauki [Modern science: actual problems of theory and practice. Series: Natural and technical sciences], 2021, No 8, pp. 68–71. – DOI 10.37882/2223-2966.2021.08.16.
21. Devlin J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding // arXiv preprint arXiv:1810.04805, 2018.
22. Fernandez P. «Through the looking glass: envisioning new library technologies» AI-text generators as explained by ChatGPT // Library hi tech news, 2023, V. 40, No 3, pp. 11–14. DOI: 10.1108/LHTN-02-2023-0017.
23. Hinton M., Wagemans J. H. M. How persuasive is AI-generated argumentation? An analysis of the quality of an argumentative text produced by the GPT-3 AI text generator // Argument & Computation. 2023, V. 14, No 1, pp. 59–74. DOI:10.3233/AAC-210026.
24. Kolin K. K. i dr. Iskusstvennyj intellekt v tehnologijah mashinnogo perevoda // Social'nye novacii i social'nye nauki [Social innovation and social sciences], 2021, No 2 (4), pp. 64–80. DOI 10.31249/snsn/2021.02.05.
25. Ge J., Lai J. C. Artificial intelligence-based text generators in hepatology: ChatGPT is just the beginning // Hepatology communications, 2023, V. 7, No 4, pp. e0097. DOI: 10.1097/HC9.0000000000000097.
26. Kozachok A. V., Spirin A. A., Erohina N. S. Metod generacii semanticheski korrektnogo koda dlja fazzingetestirovanija interpretatorov javascript // Voprosy kiberbezopasnosti [Cybersecurity issues], 2023, No 5, pp. 80–88. DOI: 10.21681/2311-3456-2023-5-80-88.
27. Raffel C. et al. Exploring the limits of transfer learning with a unified text-to-text transformer // Journal of machine learning research, 2020, V. 21, No 140, pp. 1–67. DOI:10.48550/arXiv:1910.10683.
28. Starodubov M. I., Artem'eva I. L., Selin N. A. Metod obnaruzhenija programm-vymogatelej na osnove analiza povedencheskogo otcheta ispolnjaemogo objekta // Voprosy kiberbezopasnosti [Cybersecurity issues], 2024, No 3, pp. 85–89. DOI: 10.21681/2311-3456-2024-3-85-89.
29. Bera S., Shrivastava V. K. Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification // International Journal of Remote Sensing, 2020, V. 41, No 7, pp. 2664–2683. DOI:10.1080/01431161.2019.1694725.