КЛАССИФИКАЦИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПО СПОСОБУ ВНЕДРЕНИЯ

Царегородцев Д. В.
Кафедра интеллектуальных информационных технологий, Белорусский государственный университет информатики и радиоэлектороники
Минск, Республика Беларусь
E-mail: tsaregorotsev.d@gmail.com

В представленной работе рассматривается классификация вредоносного программного обеспечения по способу внедрения с помощью древовидного крассификатора, описаны способы улучшения качества классификации и способы предотвращения переобучения классификатора

Введение

Ежедневно в мире создаётся большое количество новых либо модифицированных экземпляров вредоносных программ, каждую из которых необходимо исследовать для того, чтобы узнать, какие вредоносные действия она совершает, разработать защиту от неё и восстановить систему после заражения, если это возможно.

Очевидно, что подобный анализ каждого вредоносного файла вряд ли является возможным. Таким образом, возникает такая задача, как автоматическая классификация вредоносного программного обеспечения.

В данной работе будет рассмотрена классификация вредоносных файлов по способу внедрения. По способу внедрения мы будем разделять файлы на пять классов: failed_analysis, clear, trojan, warm, downloader.

I. Получение данных

Получение данных для данной работы осуществлялось с помощью системы автоматического анализа вредоносных файлов Cuckoo Sandbox[1]. Преимуществом данной системы являются генерация отчётов в формате json, который крайне удобен для дальнейшего использования и наличие легко расширяемого набора сигнатур.

Для разработки классификатора было проанализировано порядка 4000 исполняемых файлов с помощью системы автоматического анализа Cuckoo Sandbox.

II. Конструирование признаков

Изначально полученные данные для классификации представлены в виде времени работы виртуальной среды, в которой исполнялся файл, которое выглядит как строка вида «ЧЧ:ММ:СС» и бинарных сигнатур, которые сигнализируют об определённом поведении анализируемого исполняемого файла.

В таком виде данные не подходят для построения модели, поэтому они должны быть преобразованы в подходящий вид. Для начала время было переведено из строковой переменной в числовую, путём перевода в количество секунд, затраченных на анализ.

Существует некоторый набор сигнатур, которые сигнализируют об одном и том же поведении файла, но делают это разными способами (таким образом увеличивается точность результата, так как при не срабатывании одной сигнатуры по какой-либо причине, её будет дублировать другая). Использовать все такие сигнатуры для классификации будет избыточно, более того, они могут создавать ненужный шум, так что было решено объединять подобные наборы сигнатур в один признак, значение которого будет принимать положительное значение при срабатывании хотя бы одной сигнатуры из его набора.

Были выделены следующие наборы сигнатур:

1. сигнализирующие о скачивании файла из сети;
2. сигнализирующие о распаковке других файлов из анализируемого.

Так же из обучающей выборки были удалены все сигнатуры, энтропия которых равна нулю, то есть все, которые не сработали ни на одном файле и все, которые сработали сразу на всех.

III. Выбор алгоритма

Для классификации был выбран алгоритм машинного обучения «дерево принятия решений». В данном случае дерево будет являться деревом для классификации, так как предсказываемый результат является классом, к которому принадлежат данные [2].

Дерево классификации было построено по алгоритму ID3[3]:

1. Взять все неиспользованные признаки и посчитать их энтропию относительно тестовых образцов;
2. выбрать признак, для которого энтропия минимальна (а информационная выгода соответственно максимальна);
3. сделать узел дерева, содержащий этот признак.

Выбор атрибута был проведён на основании прироста информации(IG), который был рассчи-

тан по формуле (1)[4].

$$IG(Q) = S_0 - \sum_{i=1}^{q} N_i/N S_i, (1)$$

где Q – признак, S_0 – энтропия до деления, N – количество экземпляров до деления, S_i – энтропия i-той части после деления, N_i – количество экземпляров в i-той группе после деления, а q – количество возможных значений переменной Q.

Сама энтропия была рассчитана по формуле (2).

$$S = -\sum_{i=1}^{N} p_i \log_2 p_i, (2)$$

где N – количество возможных классов, p_i – вероятность нахождения экземпляра в классе i.

Дерево классификации останавливает своё деление при достижении глубины равной 3. При такой глубине дерево даёт верный ответ для 70,3% экземпляров, что является довольно низким результатом.

IV. Кросс-валидация

Для того, чтобы снизить вероятность переобучения дерева на обучающей выборке, для оценки качества был использован метод перекрёстной проверки[5].

Вся обучающая выборка была разделена на пять частей, после чего дерево тренировалось на 4 частях, а качество классификации проверялось на пятой. Данный процесс повторялся пять раз, после чего общим результатом качества классификации принималось среднее значение качества для всех результатов.

V. Нахождение оптимальных параметров

Для нахождения оптимального параметра глубины дерева дерево было обучено много раз с глубиной от 2 до 10, после чего было рассчитано качество классификации для каждого варианта (Рис. 1).

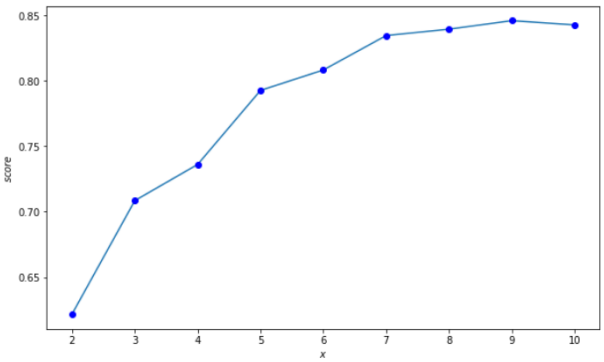

Рис. 1 – Зависимость качества классификации от глубины дерева

Таким образом, оптимальной глубиной для дерева является 9, которая даёт качество классификации 84,5%.

Далее была построена матрица ошибок дерева классификации, которая представлена на таблице 1.

Таблица 1 – Матрица ошибок дерева классификации

	clear	trojan	failed	d-loader	worm
clear	217	0	212	0	0
trojan	0	83	0	0	1
failed	16	0	364	0	0
d-loader	1	0	1	415	0
worm	0	0	0	0	10

При анализе ошибок дерева классификации можно заметить, что абсолютное большинство ошибок происходит между двумя классами – невредоносным программным обеспечением и программным обеспечением, которое не было проанализировано корректно. Улучшение качества классификации между этими двумя классами будет проработано в будущем, так как для поставленной задачи ошибки между этими классами не критичны, потому что классы вредоносного программного обеспечения были классифицированы с высокой степенью точности, что и являлось основной целью данного классификатора.

Список литературы

1. Cuckoo Sandbox – automated malware analysis [Электронный ресурс]. – Режим доступа: https://cuckoosandbox.org/. – Дата доступа: 09.10.2019.
2. Quinlan, J. R. C4.5: Programs for Machine Learning / J. R. Quinlan // San Francisco, CA, USA. Morgan Kaufmann Publishers, Inc. – 1993.
3. Паклин, Н. Б. Бизнес-аналитика: от данных к знаниям(+CD): Учебное пособие. 2-е изд.. / Н. Б. Паклин, В. И. Орешков // СПб: Питер, 2013.
4. Mitchell, T. C4.5: Machine Learning / T. Mitchell // McGraw-Hill Science/Engineering/Math. – 1997.
5. Лопез, П. М. Машинное обучение: алгоритмы для бизнеса. / П. М. Лопез // СПб: Питер, 2019.