КЛАССИФИКАЦИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

Круглик К. С.

Белорусский государственный университет, Минск, Беларусь,
e-mail: KruglikKS@bsu.by

Современные подходы к обнаружению и классификации вредоносного программного обеспечения (далее – ПО) обычно опираются на трудоемкие процессы извлечения сигнатур из вредоносного ПО, которые затем используются для классификации. Более того, сигнатуры часто ограничиваются локальными последовательностями внутри данных, игнорируя их контекст по отношению друг к другу и ко всему вредоносному файлу в целом.

В данной работе рассматривается подход классификации вредоносного ПО, основанный на глубоком обучении. Преимуществом использования глубокого обучения является то, что его можно применять к необработанным данным без необходимости проектирования дополнительных функций для извлечения паттернов.

Описание набора данных. Для экспериментов использованы данные о вредоносном ПО из соревнования Microsoft Malware Classification Challenge на Kaggle [1]. Набор данных состоит из 10868 файлов, помеченных одним из девяти классов вредоносных программ, как представлено в таблице 1. Выборка вредоносного ПО состоит из необработанного шестнадцатеричного представления двоичного содержимого файлов без заголовка PE.

Табл. 1. Количество файлов каждого класса вредоносного ПО в выборке

Класс вредоносного ПО	Количество файлов
Ramnit	1541
Lollipop	2478
Kelihos_ver3	2942
Vundo	475
Simda	42
Tracur	751
Kelihos_ver1	398
Obfuscator.ACY	1228
Gatak	1013

Предобработка данных. При использовании глубоко обучения необходимо привести файлы к одному размеру, этот размер должен быть ограничен для удобного хранения данных и эффективной работы с ними. Также необходимо сохранить первоначальную структуру данных насколько это возможно. Поэтому для каждого файла был использован следующий алгоритм обработки данных с потерями.

Байтовое представление файла рассматривается как одномерное изображение и масштабируется до фиксированного размера. По сравнению с подходом представления файла как двумерного изображения, данный подход проще, т.к. не надо выбирать высоту и ширину изображения. Также преобразование файла в одномерный

поток байтов сохраняет порядок следования двоичного кода в исходном файле, что удобно для применения рекуррентной нейронной сети [2].

Файлы рассматриваемой выборки вредоносного ПО представлены в виде изображения размера 1 на 10000 байтов, т.е. как последовательность 10000 однобайтовых значений. Масштабирование было реализовано при помощи библиотеки OpenCV [3].

Архитектура глубокого обучения. В данной работе рассмотрены три различные архитектуры нейронной сети. Сначала к последовательности байтов применяется несколько сверточных слоев [4]. Сверточные слои инвариантны к сдвигу, данное свойство помогает слоям находить паттерны в любом месте файла. Поверх сверточных слоев рассматриваются два разных подхода.

В первой модели, CNN, после последнего сверточного слоя следует слой выравнивания, полносвязный слой и выходной полносвязный слой с функцией активации Softmax для классификации на 9 классов вредоносного ПО. Этот подход на основе сверточных нейронных сетей классифицирует данные с использованием локальных шаблонов каждого класса вредоносного программного обеспечения.

Для второй и третьей моделей поверх сверточных слоев применены рекуррентные слои нейронной сети – модуль долгой краткосрочной памяти (LSTM) [5]. Т.к. предполагается, что существует зависимость между различными фрагментами байтового кода, то рекуррентный слой должен объединить содержимое всего файла в один вектор признаков, прежде чем подать его на выходной слой.

Во второй модели, CNN-UniLSTM, применяется один прямой LSTM слой поверх сверточного слоя, где соединительное направление ячеек в LSTM идет от начала к концу файл.

В третьей модели, CNN-BiLSTM, учтено, что зависимость между кодом в двоичном файле может быть не только в одном направлении. В данной модели выходы сверточной нейронной сети соединяются как с прямым слоем LSTM, так и с обратным, далее объединяются и подаются на выходной слой.

Все модели имеют три сверточных слоя с функцией активации ReLU, количество фильтров на трех слоях равно 30, 50 и 90. После каждого сверточного слоя следует слой субдискретизации с функцией максимума. Для моделей CNN выходы сверточных слоев соединяются с полносвязным слоем из 256 нейронов, а затем подаются на выходной слой. Для моделей CNN-UniLSTM и CNN-BiLSTM выходы сверточных слоев соединяются с одним или двумя слоями LSTM соответственно, каждый из которых имеет 28 скрытых нейронов; выходы слоев LSTM затем подключаются к выходному слою. Количество параметров для модели CNN равно 1842069, для модели CNN-UniLSTM – 155669, для модели CNN-BiLSTM – 268949.

Результаты вычислительного эксперимента. Выборка из 10868 предобработанных файлов разделена на обучающую для обучения нейронной сети, валидационную для проверки качества на обучении и тестовую для получения окончательных результатов, объем каждой из выборок составил 7281, 717, 2870 файлов соответственно.

Для оценки качества использованы две метрики – точность классификации на всей выборке и усредненная макро F1-оценка, которая вычисляет F1-оценку для каждого класса по отдельности и усредняет. Размер пакета для каждой модели равняется 64, шаг обучения – 0.001.

В таблице 2 представлено время обучения для каждой из рассмотренных нами моделей, а также результаты на обучении. В таблице 3 показаны значения метрик на тестовой выборке данных.

Табл. 2. Результаты обучения моделей

Модель	Количество эпох обучения	Общее время обучения, мин	Среднее время обучения за эпоху, с	Точность на тренировочной выборке, %	Точность на валидационной выборке, %
CNN	80	6.12	4.59	98.38	91.77
CNN-UniLSTM	80	63.56	47.66	96.74	90.38
CNN-BiLSTM	60	90.45	90.45	97.48	94.00

Табл. 3. Результаты обучения моделей

Модель	Точность, %	Макро F1-оценка, %	Время, потраченное на классификацию тестовой выборки, с
CNN	91.01	81.26	0.83
CNN-UniLSTM	91.46	83.93	3.07
CNN-BiLSTM	93.52	84.89	5.56

Эксперименты показывают, что одномерное представление необработанного двоичного файла является хорошим представлением для задачи классификации вредоносного программного обеспечения. Наилучшее качество достигнуто при обучении CNN-BiLSTM. Добавление обратного направления зависимости помогает улучшить качество модели глубокого обучения.

Из-за последовательной зависимости при вычислении ячеек в LSTM-слое CNN-BiLSTM долго обучается, модель CNN обучается в 15 раз быстрее. С другой стороны, модель CNN-BiLSTM использует 268949 параметров, а модель CNN – 1.84 миллиона. Модель CNN-UniLSTM является хорошим компромиссом: ее обучение занимает меньше времени, чем обучение модели CNN-BiLSTM, но данная модель показывает результаты лучше, чем CNN.

Время классификации новых файлов также является плюсом. Предсказание класса для нового предобработанного файла занимает в среднем 0.001 секунды. Еще одним преимуществом подхода глубокого обучения является то, что при получении новых размеченных файлов вредоносного ПО, нашу модель можно дообучить для повышения ее качества, а не обучать новую модель заново.

Литература

1. Kaggle Competition [Electronic resource] – Mode of access: https://www.kaggle.com/c/malware-classification. – Date of access: 15.03.2021.

2. Deep learning at the shallow end: Malware classification for non-domain experts / Q. Le [et al.] // Digital Investigation. – 2018. – Vol. 26. – P. 118-126.

3. Bradski, G. Learning OpenCV: Computer vision with the OpenCV library. / G. Bradski, A. Kaehler – "O'Reilly Media, Inc.", 2008.

4. Convolutional networks for images, speech, and time series / Y. LeCun [et al.] // The handbook of brain theory and neural networks. – 1995. – Vol. 3361, № 10. – P. 1995.

5. Hochreiter, S. LSTM can solve hard long time lag problems / S. Hochreiter, J. Schmidhuber // Advances in neural information processing systems. – 1997. – P. 473-479.