Научно-практический электронный журнал
Оригинальные исследования (ОРИС), *ISSN 2222-0402*
www.ores.su
original-research.ru

УДК 004.896

ПРИМЕНЕНИЕ АУГМЕНТАЦИИ ДАННЫХ ДЛЯ ПОВЫШЕНИЯ ТОЧНОСТИ КЛАССИФИКАЦИИ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ МОДЕЛИ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ

Камольцев Даниил Алексеевич,
студент,
Московский государственный технический университет
имени Н.Э. Баумана.
kamoltsev97@mail.ru

Шаброва Анна Сергеевна,
студент,
Московский государственный технический университет
имени Н.Э. Баумана.
shabrova.anna.2410@list.ru

Аннотация

В статье рассматривается подход к классификации вредоносного программного обеспечения на основе сверточной нейронной сети с применением аугментации данных. В качестве средства расширения обучающей выборки используется добавление шумовых искажений различного типа (гауссов, пуассонов, лапласов шум) к изображениям, полученным в результате преобразования бинарных представлений исполняемых файлов. Исходные бинарные данные конвертируются в трехканальные изображения с использованием техники блочного преобразования, что позволяет фиксировать структуру файла в визуальной форме. Аугментация направлена на моделирование разнообразных искажений входных данных, характерных для метаморфных семейств вредоносных программ, и служит для повышения устойчивости и обобщающей способности модели. Разработанная система включает 3 ключевых компонента: генерацию изображений из исполняемых файлов, создание их искажённых копий и последующую классификацию с помощью архитектуры сверточной нейронной сети. Такой подход позволяет применять методы компьютерного зрения для задачи обнаружения семейств вредоносного программного обеспечения при высокой изменчивости их структуры.

Ключевые слова: сверточные нейронные сети, классификация вредоносного ПО, аугментация данных, предобработка данных, преобразование бинарных файлов.

IMPROVING MALWARE CLASSIFICATION ACCURACY VIA DATA AUGMENTATION USING A CONVOLUTIONAL NEURAL NETWORK

Kamoltsev Daniil Alekseevich,
student,

Bauman Moscow State Technical University.

Shabrova Anna Sergeevna,
student,
Bauman Moscow State Technical University.

ABSTRACT

The article explores an approach to malware classification based on a convolutional neural network with the use of data augmentation techniques. To expand the training dataset, various types of noise distortions (Gaussian, Poisson, and Laplacian noise) are applied to images generated from the binary representations of executable files. The original binary data is converted into three-channel images using a block-based transformation technique, which captures the structural characteristics of the file in visual form. Augmentation is aimed at simulating diverse input distortions typical for metamorphic malware families and serves to enhance the model's robustness and generalization capability. The developed system includes three key components: image generation from executable files, creation of their distorted copies, and subsequent classification using a convolutional neural network architecture. This approach enables the application of computer vision methods to the task of detecting malware families under conditions of high structural variability.

Keywords: convolutional neural networks, malicious software classification, data augmentation, data preprocessing, conversion of binary data.

Актуальность исследования

Современные компьютерные атаки становятся все более сложными и изощренными, создавая серьезную угрозу информационной безопасности как для компаний, так и для пользователей устройств, подключенных к сети. Вредоносное программное обеспечение (ВПО) эволюционирует, используя сложные методы сокрытия и мутации кода, что затрудняет его выявление антивирусными программами и системами защиты.

Одним из ключевых методов, применяемых вредоносными программами для обхода систем обнаружения, является метаморфизм. Данный процесс заключается в модификации внутреннего содержимого исполняемого файла при его распространении [1], что приводит к созданию множества различных хеш- значений для одной и той же вредоносной программы. Такой механизм делает стандартные методы обнаружения, основанные на сигнатурах, малоэффективными, поскольку каждое новое представление ВПО воспринимается как уникальный файл. В связи с этим возникает необходимость в применении более продвинутых методов анализа, основанных на машинном обучении и, в частности, сверточных нейронных сетях (CNN), способных распознавать вредоносные программы по их структурным признакам.

Перспективным является подход, заключающийся в преобразовании исполняемых файлов программ с целью их анализа методами компьютерного зрения. Однако эффективность нейросетевых моделей напрямую зависит от объема и разнообразия обучающей выборки. В области кибербезопасности получение достаточного количества размеченных данных может быть затруднено, так как ВПО постоянно изменяется, а доступ к реальным образцам обычно ограничен. В связи с этим, в настоящее время актуальным становится вопрос улучшения качества классификации вредоносных программ

нейронными сетями с применением аугментации данных, особенно в условиях ограниченного набора данных на входе. Таким образом возможно искусственно расширить обучающую выборку с помощью трансформаций.

Аугментация данных позволяет повысить обобщающую способность модели, делая ее более устойчивой к изменчивости входных данных. Этот процесс может включать в себя различные модификации изображений, такие как добавление шумов, цветовые и геометрические преобразования и другие методы. Подобное расширение обучающей выборки способствует повышению точности классификации ВПО и, следовательно, снижению риска возникновения ложных срабатываний.

Настоящее исследование направлено на повышение точности классификации вредоносных программ с использованием модели сверточной нейронной сети посредством аугментации данных на основе шумовых искажений.

Архитектура модели классификации

Предлагаемый метод классификации ВПО основан на преобразовании программных файлов в изображения, их последующем анализе с помощью CNN и аугментации обучающей выборки. Это обеспечивает устойчивость модели к вариативности входных данных и повышает качество классификации.

На рисунке 1 представлена поэтапная схема модели классификации вредоносного ПО, построенная на основе предложенного в работе [2] подхода. На ней отображены ключевые стадии обработки данных: от загрузки бинарного файла и его преобразования в изображение до аугментации данных с добавлением различных типов шумов и последующего обучения сверточной нейросети.

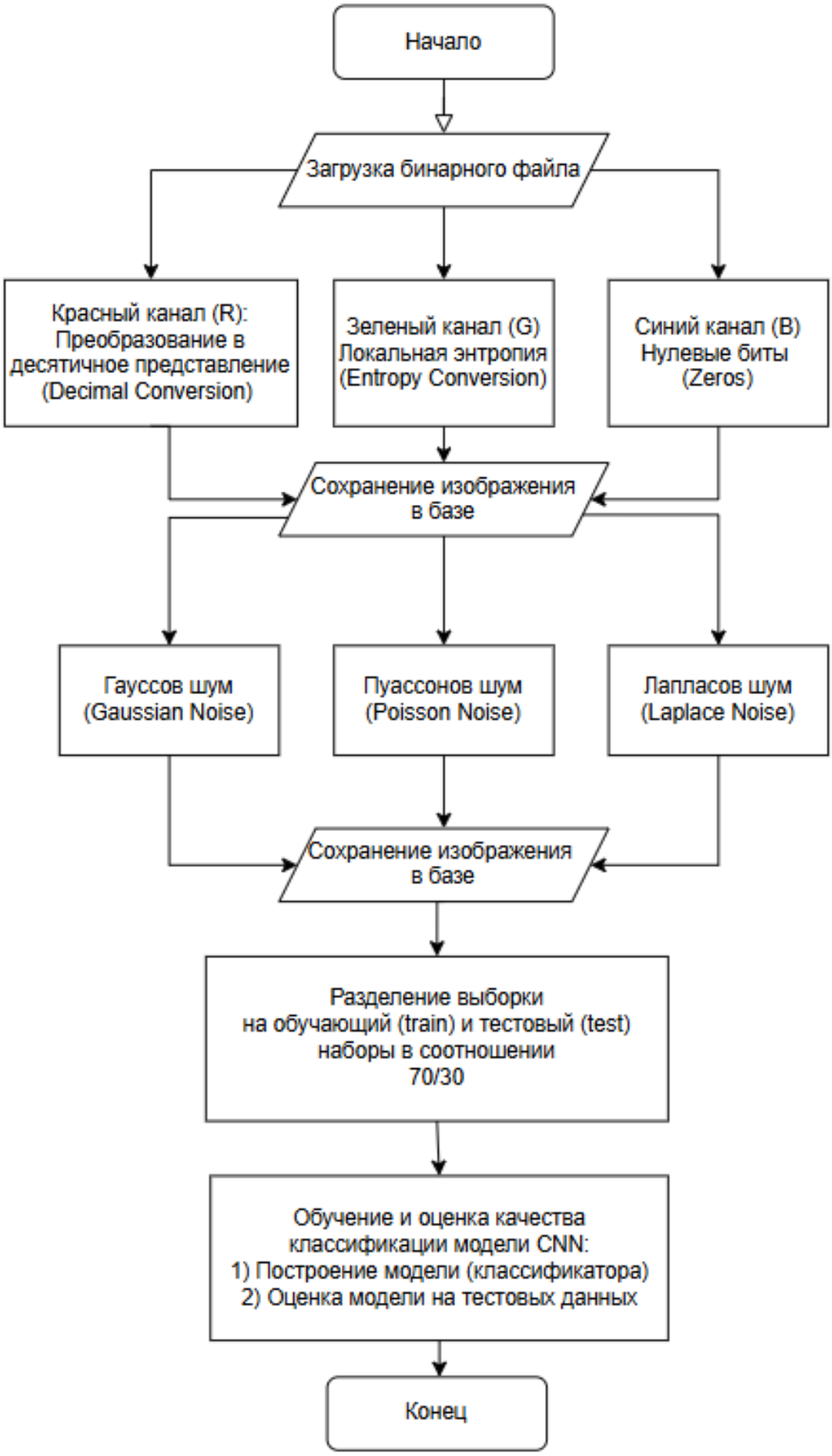


Рис. 1. Схема модели классификации вредоносного программного обеспечения.

Алгоритм преобразования файлов в изображения

Преобразование файлов в изображение происходит по следующему алгоритму:

Файл открывается в бинарном режиме и читается в байтовый массив;

Файл преобразуется в массив целых чисел (от 0 до 255);

Формируются цветовые R-G-B каналы:

красный (R): содержит значения исходных байтов;

зеленый (G): рассчитывается как локальная энтропия в пределах скользящих окон размером 256 байт;

синий (B): инициализируется нулями.

Формируется трехканальное изображение:

при недостаточном объеме данных массив дополняется нулями;

на основе подготовленных каналов формируется RGB-изображение.

Полученное изображение сохраняется в формате PNG для последующего анализа и обработки.

Формирование зеленого канала (G) изображения из бинарного файла на основе локальной энтропии предполагает численное представление информационной сложности в каждом участке файла. Сначала файл последовательно разбивается на неперекрывающиеся блоки по 256 байт: $B_1, B_2, \dots, B_N$, где каждый $B_k \in \{0,1, \dots,255\}^{256}$. Далее происходит подсчет частот появления байтов:

для каждого блока B_k вычисляется гистограмма частот появления каждого байта (1):

$$count_i = \text{количество вхождений байта } i \text{ в блоке } B_k, i \in \{0, \dots,255\} \quad (1)$$

полученные частоты нормализуются до вероятностей (2):

$$p_i = \frac{count_i}{\sum_{j=0}^{255} count_j} = \frac{count_i}{256}, \quad (2)$$

так как в каждом блоке всегда ровно 256 байт.

Для каждого блока B_k вычисляется энтропия (3) по формуле Шеннона [3]:

$$H_k = -\sum_{i=0}^{255} p_i \cdot \log_2(p_i). \quad 3)$$

Полученное значение энтропии масштабируется:

максимально возможное значение энтропии (4) при равномерном распределении (все p_i=1/256):

$$H_{\max} = \log_2(256) = 8. \quad 4)$$

энтропия нормализуется до диапазона (0, 255) для отображения в виде значения компонента цвета (5):

$$G_k = \left[\frac{H_k}{8} \cdot 255 \right]. \quad 5)$$

Каждому пикселю будущего изображения, соответствующему блоку B_k , присваивается значение G_k. Таким образом, зеленый канал — это карта нормализованных локальных энтропий всех 256-байтных блоков файла.

Пример преобразованного в изображение вредоносного файла (Adware) представлен на рисунке 2.

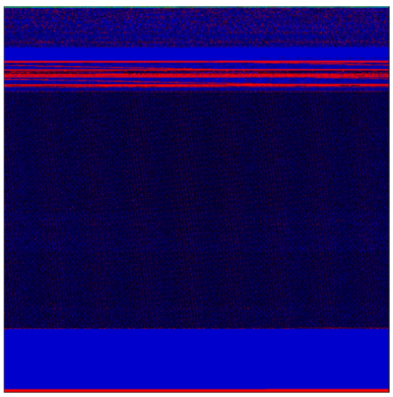


Рис. 2. Преобразованный в изображение вредоносный файл Adware:Win32/Lollipop (рекламное ПО).

Добавление шумов на изображения

Для решения проблемы недостатка данных в обучающей выборке применялась аугментация (data augmentation) – процесс создания дополнительного набора данных из имеющегося набора [4]. Одним из наиболее популярных способов аугментации, позволяющим повысить устойчивость моделей машинного обучения и увеличить вариативность обучающей выборки, является добавление шумов. Для улучшения качества классификации объектов на изображениях были выбраны следующие типы шумов:

Гауссов шум (additive Gaussian noise);

Лапласов шум (additive Laplace noise);

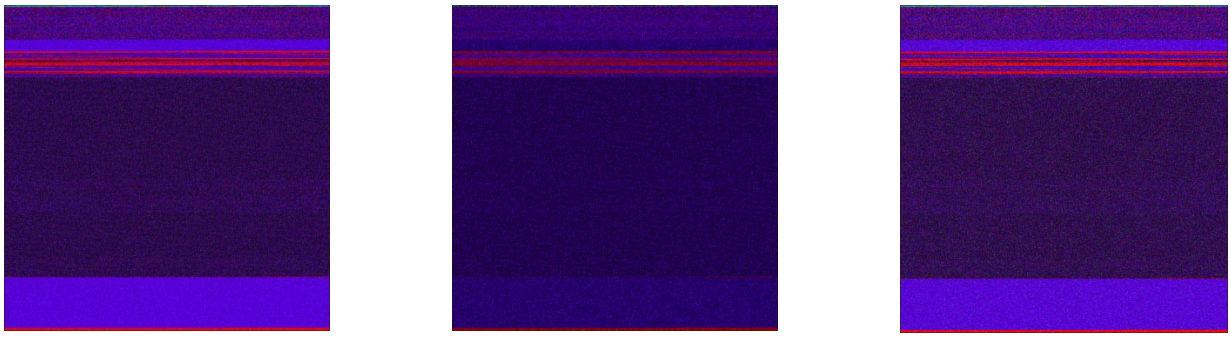
Пуассонов (дробовой) шум (additive Poisson noise).

Гауссов шум используется для имитации воздействия множества случайных процессов, происходящих в природе [5]. Это помогает сделать модель более устойчивой к небольшим вариациям данных.

Пуассонов шум возникает в системах, где сигналы являются счетными процессами, т.е. числом событий, произошедших за фиксированный промежуток времени [6]. Такой тип шумовых искажений используется для моделирования случайных флуктуаций, возникающих в дискретных стохастических процессах.

Лапласов шум — это вид аддитивного шума, моделируемый с помощью распределения Лапласа, при котором к каждому пикселю изображения добавляется случайная величина. Такой шум подчеркивает резкие изменения яркости и подавляет однородные участки изображения, поскольку оператор Лапласа чувствителен к скачкам интенсивности и игнорирует области с плавными градиентами [7].

Примеры изображений с наложенными шумами разных типов представлены на рисунке 3.



а) б) в)

Рис. 3. Наложение шумов на изображение ВПО Adware:Win32/Lollipop:
а) Гауссов шум; б) Пуассонов шум; в) Лапласов шум.

Модель сверточной нейронной сети (CNN) и исходные данные

Была построена модель сверточной нейронной сети (CNN) для классификации изображений, полученных из бинарных файлов. В качестве основы выбрана архитектура модели VGG19, состоящей из последовательности сверточных блоков, каждый из которых включает два или более сверточных слоев с ядрами 3×3 и функцией активации ReLU [8], за которыми следует слой подвыборки (MaxPooling). После сверточных блоков следуют полносвязные слои (Dense), завершающиеся выходным Softmax-слоем (рис. 4, [9]).

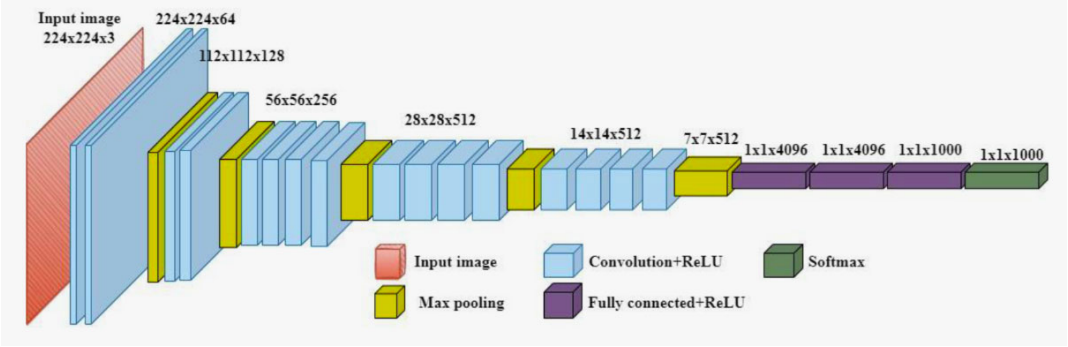


Рис. 4. Архитектура модели CNN VGG19, [9]

Модель включает следующие слои:

2 сверточных слоя (Conv2D, 32 фильтра, 3×3, ReLU);

2 MaxPooling слоя подвыборки (2×2);

Сверточный слой (Conv2D, 64 фильтра, 3×3, ReLU);

Flatten слой (для преобразования карт признаков, полученных от свёрточных слоёв, в одномерный вектор);

Полносвязный слой (Dense, 128 нейронов, ReLU);

Выходной слой (Dense, 9 классов, Softmax).

Классификация ВПО выполнялась на основе 9 классов, представленных в датасете Microsoft Malware Classification Challenge (BIG 2015) [10]. Датасет содержит множество известных вредоносных файлов, принадлежащих к девяти семействам (табл. 1) и *включает* 10868 образцов ВПО в формате .byte.

Каждому файлу соответствует уникальный идентификатор (Id), представленный в виде 20-символьного хэша, а также метка класса (Class), обозначающая принадлежность к одному из семейств. Исходные данные включают шестнадцатеричное представление бинарного содержимого файла без PE- заголовка.

Таблица 1 – Семейства вредоносного ПО в наборе данных Microsoft Malware Classification Challenge

Название семейства	Число образцов в обучающей выборке	Тип ВПО
Ramnit	1541	Червь (Worm)
Lollipop	2478	Рекламное ПО (Adware)
Kelihos_ver3	2942	Бэкдор (Backdoor)
Vundo	475	Троян (Trojan)
Simda	42	Бэкдор (Backdoor)
Tracur	751	Троян-загрузчик (Downloader)
Kelihos_ver1	398	Бэкдор (Backdoor)
Obfuscator.ACY	1228	(Obfuscated malware)
Gatak	1013	Бэкдор (Backdoor)

Метрики оценки качества классификации

С целью оптимизации обучения и тестирования модели на исходных и искусственно «зашумленных» изображениях обучающий набор данных был сокращен до 2000 образцов.

Для оценки качества классификации вредоносного ПО с помощью построенной модели использовались следующие метрики:

Точность классификации (Accuracy) (6):

$$Accuracy = \frac{\sum_{i=1}^{C} TP_i}{N}, \quad (6)$$

где:

C — количество классов,

TP_i — количество правильно классифицированных объектов класса i,

N - общее количество всех объектов.

Логарифмическая функция потерь (Logloss) (7):

$$Logloss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)), \quad (7)$$

где N — общее количество примеров, y_i — истинная метка i-го примера (0 или 1), p_i — предсказанная моделью вероятность того, что i-й пример принадлежит положительному классу.

Точность (Precision) (8):

$$Precision = \frac{TP}{TP + FP}, \quad (8)$$

где TP обозначает количество образцов, правильно классифицированных в свои семейства, а FP — количество образцов, ошибочно отнесенных к этому семейству.

Полнота (Recall), также известная как чувствительность (9):

$$Recall = \frac{TP}{TP + FN}, \quad (9)$$

где FN — количество образцов, которые были ошибочно классифицированы как принадлежащие другим семействам.

F1-мера (F1-score) (10):

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \quad (10)$$

Результаты обучения и тестирования модели классификации

Данные были распределены в соотношении 70/30: 70% на обучение и 30% на тестирование (валидацию). Такой подход обеспечивает объективную оценку качества модели и позволяет избежать переобучения (overfitting) на гиперпараметрах или утечки информации (information leaking).

На рисунке 5 показаны графики зависимости точности классификации (accuracy) и значения функции потерь (loss function) от числа эпох обучения модели на исходных данных. Как видно, точность на тестовой выборке начала расти с 6-й эпохи и достигла максимального значения на 16-й. Функция потерь сначала демонстрировала скачкообразный рост до 5-й эпохи, после чего начала резко снижаться, достигнув минимума также на 16-й эпохе.

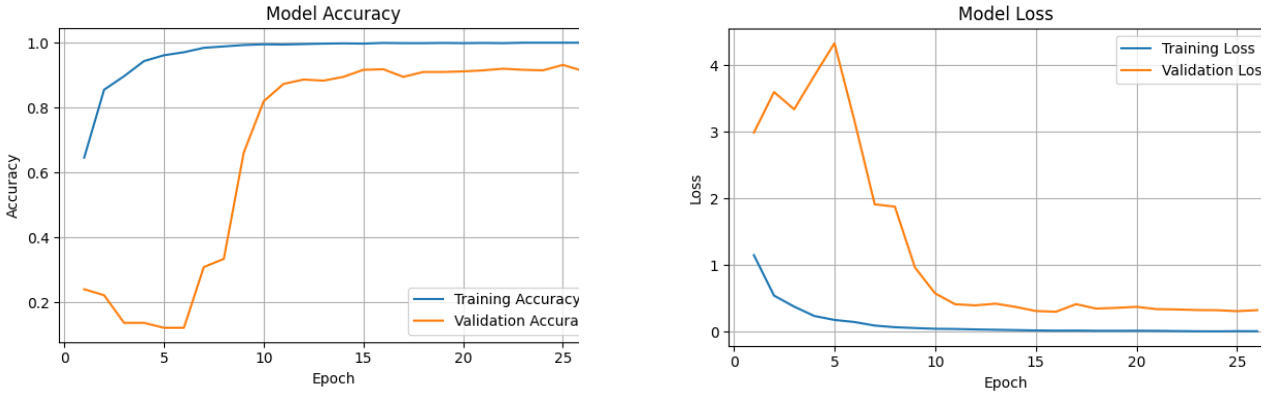


Рис. 5. Графики зависимости точности классификации и значений функции потерь от числа эпох обучения для исходного набора данных.

На рисунке 6 представлена матрица ошибок (confusion matrix), показывающая количество правильно и неправильно классифицированных образцов в выборке. По горизонтали указаны предсказанные классы, по вертикали — истинные. Каждая ячейка содержит значение, представляющее собой отношение числа предсказаний алгоритма для определённого класса к общему количеству образцов этого класса, то есть действительное число в интервале [0, 1]. Видно, что наихудший результат получен для семейства ВПО Simda — 0,31. Это обусловлено малым количеством исходных образцов: всего 42 файла в датасете, что после разделения на обучающую и тестовую выборки дало лишь 13 изображений для валидации.

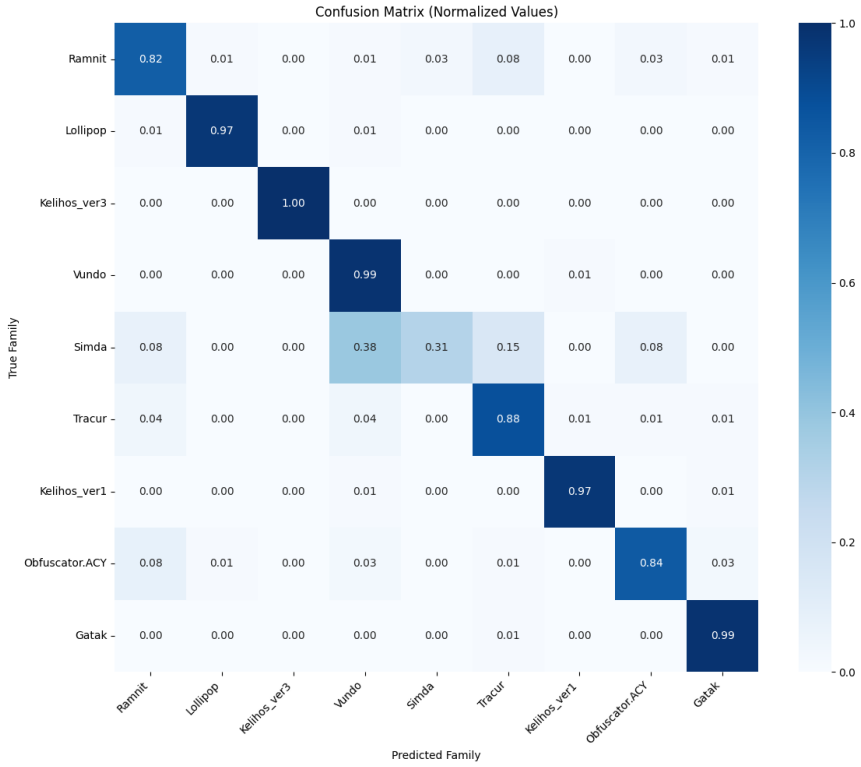


Рис. 6. Матрица ошибок классификации модели CNN для исходной обучающей выборки.

В таблице 2 приведены значения метрик, используемых для оценки качества классификации модели по каждому из классов, после обучения на исходном наборе данных. Как видно из таблицы, наихудшие результаты были также получены для семейства ВПО Simda.

Таблица 2 – Значения метрик оценки качества модели CNN для каждого семейства ВПО без аугментации

Название семейства	Точность (Precision)	Полнота/Чувствительность (Recall)	F_1-score
Ramnit	0.8451	0.8219	0.8333
Lollipop	0.9726	0.9726	0.9726
Kelihos_ver3	1.0000	1.0000	1.0000
Vundo	0.8488	0.9865	0.9125
Simda	0.6667	0.3077	0.4211
Tracur	0.8649	0.8767	0.8707
Kelihos_ver1	0.9726	0.9726	0.9726
Obfuscator.ACY	0.9394	0.8378	0.8857
Gatak	0.9351	0.9863	0.9600

Далее обучающая выборка была расширена добавлением «зашумленных» образцов. Для того чтобы добиться минимального разброса значений между классами было решено произвести аугментацию с получением одинакового числа изображений для каждого класса в результирующем наборе - по 300 образцов в семействе.

На рисунке 7 показаны графики зависимости точности классификации (accuracy) и значения функции потерь (loss function) от числа эпох обучения модели на расширенном наборе данных. Как видно, уже к 12-й эпохе были достигнуты наилучшие значения этих показателей. На рисунке 8 представлена матрица ошибок, отражающая результаты обучения и валидации модели на аугментированном наборе данных. Для семейства Simda показатель отношения числа предсказанных и истинных образцов стабилизировался и достиг значения 1.00. Для остальных классов также наблюдается высокая точность классификации, что свидетельствует о положительном эффекте аугментации на обобщающую способность модели.

Рис. 7. Графики зависимости точности классификации и значений функции потерь от числа эпох обучения после аугментации.

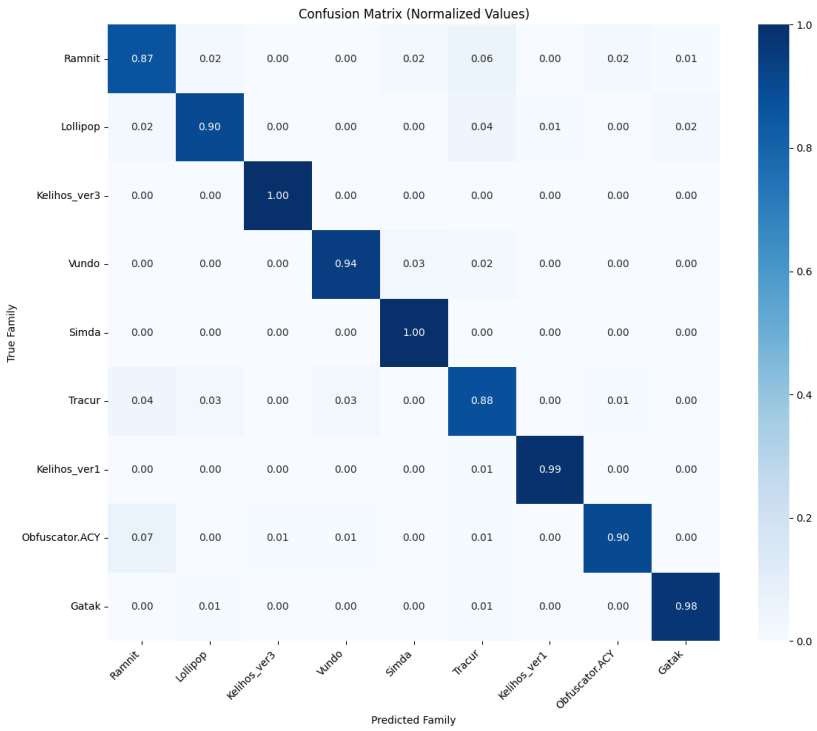


Рис. 8. Матрица ошибок классификации модели CNN для расширенной обучающей выборки.

В таблице 3 приведены значения метрик оценки качества классификации модели для каждого из классов после обучения на расширенном наборе данных. Как видно, для семейства ВПО Simda показатели также выровнялись.

Таблица 3 – Значения метрик оценки качества модели CNN для каждого семейства ВПО с аугментацией обучающей выборки

Название семейства	Точность (Precision)	Полнота/Чувствительность (Recall)	F_1-score
Ramnit	0.8667	0.8667	0.8667
Lollipop	0.9310	0.9000	0.9153
Kelihos_ver3	0.9890	1.0000	0.9945
Vundo	0.9551	0.9444	0.9497
Simda	0.9474	1.0000	0.9730
Tracur	0.8495	0.8778	0.8634
Kelihos_ver1	0.9889	0.9889	0.9889
Obfuscator.ACY	0.9643	0.9000	0.9310
Gatak	0.9670	0.9778	0.9724

Таким образом, аугментация данных позволила существенно улучшить результаты классификации для класса с изначально малым числом образцов, недостаточным для обучения модели CNN. Улучшение метрик и стабилизация показателей на графиках и в матрице ошибок подтверждают повышение обобщающей способности модели после расширения обучающей выборки.

Заключение

Результаты экспериментов показали, что использование шумов для аугментации положительно влияет на способность модели обобщать данные, особенно в случае классов с небольшим количеством исходных образцов. Модель, обученная на расширенной

выборке, продемонстрировала лучшую точность по сравнению с вариантом, обученным только на исходных данных.

Таким образом, была подтверждена целесообразность использования аугментации как эффективного инструмента повышения точности классификации в задачах анализа вредоносного программного обеспечения.

Список литературы:

1. ReasonLabs: What is Metamorphism? [Электронный ресурс]. – URL: https://cyberpedia.reasonlabs.com/EN/metamorphism.html (дата обращения: 16.05.2025)
2. Catak F.O., Ahmed J., Sahinbas K., Khand Z.H. Data augmentation based malware detection using convolutional neural networks // PeerJ Computer Science, 2021. Vol. 7. - DOI: https://doi.org/10.7717/peerj-cs.346
3. Шеннон К. Работы по теории информации и кибернетике // Издательство иностранной литературы. – М., 1963. – 832 с.
4. ГОСТ Р 59898-2021 Оценка качества систем искусственного интеллекта. Общие положения. – М.: Российский институт стандартизации, 2021. – 19 с.
5. Selesnick I.W. The Estimation of Laplace Random Vectors in Additive White Gaussian Noise // IEEE Transactions on Signal Processing. – 2008. – С. 3482- 3496. – DOI: 10.1109/TSP.2008.920488.
6. Wojtkiewicz S.F., Johnson E.A., Bergman L.A., Grigoriu M., Spencer B.F.: Response of stochastic dynamical systems driven by additive Gaussian and Poisson white noise: Solution of a forward generalized Kolmogorov equation by a spectral finite difference method // Computer Methods in Applied Mechanics and Engineering. – 1999. – Т. 168. - С. 73–89. – URL: https://doi.org/10.1016/S0045-7825(98)00098-X
7. Нгок Х.Т.Д. Метод устранения лапласовского шума на изображениях // Естественные и математические науки в современном мире. - 2014.
8. Соснин А. С., Суслова И. А. Функции активации нейросети: сигмоида, линейная, ступенчатая, ReLU, Tahn // «Наука. Информатизация. Технологии. Образование»: материалы XII международной научно-практической конференции. — Екб, 2019. — С. 237–246
9. Nguyen T.-H., Nguyen T.-N., Ngo B.-V. A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease // AgriEngineering. - 2022. – Т. 4. - С. 871-887. - URL: https://doi.org/10.3390/agriengineering4040056
10. Ronen R., Radu M., Feuerstein C., Yom-Tov E., Ahmadi M. Microsoft malware classification challenge // arXiv preprint arXiv:1802.10135. - 2018. - URL: https://arxiv.org/pdf/1802.10135

References:

1. ReasonLabs: What is Metamorphism? [Electronic resource]. – URL: https://cyberpedia.reasonlabs.com/EN/metamorphism.html (accessed: 16.05.2025)
2. Catak F.O., Ahmed J., Sahinbas K., Khand Z.H. Data augmentation based malware detection using convolutional neural networks // PeerJ Computer Science, 2021. Vol. 7. - DOI: https://doi.org/10.7717/peerj-cs.346

3. Shannon C. Works on information theory and cybernetics // Foreign Literature Publishing House. – Moscow, 1963. – 832 p.
4. GOST R 59898-2021 Quality assessment of artificial intelligence systems. General provisions. – Moscow: Russian Institute for Standardization, 2021. – 19 p.
5. Selesnick I.W. The Estimation of Laplace Random Vectors in Additive White Gaussian Noise // IEEE Transactions on Signal Processing. – 2008. – P. 3482-3496. – DOI: 10.1109/TSP.2008.920488.
6. Wojtkiewicz S.F., Johnson E.A., Bergman L.A., Grigoriu M., Spencer B.F.: Response of stochastic dynamical systems driven by additive Gaussian and Poisson white noise: Solution of a forward generalized Kolmogorov equation by a spectral finite difference method // Computer Methods in Applied Mechanics and Engineering. – 1999. – Vol. 168. - P. 73–89. – URL: https://doi.org/10.1016/S0045-7825(98)00098-X
7. Ngoc H.T.D. A method for eliminating Laplacian noise in images // Natural and mathematical sciences in the modern world. - 2014.
8. Sosnin A. S., Suslova I. A. Neural network activation functions: sigmoid, linear, step, ReLU, Tanh // "Science. Informatization. Technologies. Education": materials of the XII international scientific-practical conference. — Yekaterinburg, 2019. — P. 237–246
9. Nguyen T.-H., Nguyen T.-N., Ngo B.-V. A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease // AgriEngineering. - 2022. – Vol. 4. - P. 871-887. - URL: https://doi.org/10.3390/agriengineering4040056
10. Ronen R., Radu M., Feuerstein C., Yom-Tov E., Ahmadi M. Microsoft malware classification challenge // arXiv preprint arXiv:1802.10135. - 2018. - URL: https://arxiv.org/pdf/1802.10135