DOI: 10.24143/2072-9502-2020-1-41-49
УДК 004.056.57

МЕТОД ОБНАРУЖЕНИЯ ВИРУСОВ-ШИФРОВАЛЬЩИКОВ В КОМПЬЮТЕРНОЙ СИСТЕМЕ НА ОСНОВЕ АНАЛИЗА ИХ ПОВЕДЕНЧЕСКИХ ПРИЗНАКОВ

А. Б. Калиев, А. Н. Марьенков

Астраханский государственный университет,
Астрахань, Российская Федерация

Показана низкая эффективность существующих методов противодействия вирусам-шифровальщикам (ВШ). Обоснована актуальность разработки новых подходов к выявлению ВШ в компьютерных системах (КС). Рассмотрены методы эвристического анализа в качестве новых подходов к обнаружению ВШ. Представлена новая методика обнаружения ВШ на основе анализа изменений значений параметров КС. Построены модели с использованием методов машинного обучения, позволяющие выявлять начавшуюся атаку ВШ на КС. Целью проведения эксперимента было получение модели, имеющей наиболее высокий процент выявления атак ВШ на КС и наименьшее количество ложных срабатываний. В качестве алгоритмов моделирования были использованы наивный байесовский классификатор, многослойная нейронная сеть, машина опорных векторов, алгоритм градиентного бустинга CatBoost. Для построения моделей использованы программные пакеты, написанные с использованием языка программирования Python. Данные для обучения были собраны в результате экспериментов с наиболее популярными ВШ. В качестве ключевых метрик эффективности моделей машинного обучения выбраны следующие типичные метрики: precision, recall, F1-метрика, accuracy, AUC. В ходе проведенных экспериментов сформированы значения матриц ошибок и получены основные показатели метрик качества моделей. Помимо метрик эффективности классификации приведено среднее время выполнения операций по классификации для каждой из моделей. В процессе анализа результатов обучения моделей было выявлено, что наилучшими показателями по выявлению ВШ в КС обладает модель, построенная на основе алгоритма градиентного бустинга CatBoost. Сделаны выводы о возможности применения данного подхода для выявления атак ВШ на КС.

Ключевые слова: вирус-шифровальщик, выявление вирусов, компьютерная система, программное обеспечение, методы эвристического анализа, машинное обучение.

Для цитирования: *Калиев А. Б., Марьенков А. Н.* Метод обнаружения вирусов-шифровальщиков в компьютерной системе на основе анализа их поведенческих признаков // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2020. № 1. С. 41–49. DOI: 10.24143/2072-9502-2020-1-41-49.

Описание предметной области

Вирусы-шифровальщики (ВШ) остаются одной из главных проблем в области защиты личных данных пользователей компьютерных систем (КС). С одной стороны, последствия заражения системы ВШ в большинстве случаев приводят к частичной или полной утере данных пользователя в силу невозможности расшифровывания этих данных [1], с другой, современные средства защиты от вредоносных программ зачастую не справляются со своей функцией и не защищают пользовательские КС от заражения. Классические методы борьбы с вредоносным программным обеспечением (ПО) остаются малоэффективными при обнаружении вредоносного кода, ранее не известного разработчикам антивирусных программных средств и эксплуатирующего уязвимости нулевого дня.

В качестве возможного решения проблемы обнаружения ВШ в КС могут стать методы эвристического анализа. Так, авторы работы [2] предлагают исследовать особенности поведения ВШ. Суть метода заключается в исследовании свойств поведения ВШ и сравнении этих свойств с поведением штатных программ КС. Авторы выдвигают гипотезу о том, что ВШ и легитимное ПО в части работы с файлами имеют как общие черты, так и различия. Общее между этими классами ПО в том, что они используют одни и те же файловые операции. Различия заключаются в порядке использования операций и их аргументах. Активность вредоносных программ проявляется согласно заранее заданному алгоритму, в то время как работа легитимного ПО зависит

от действий пользователя. Другой способ выявления ВШ основывается на мониторинге подозрительной активности в КС [3]. Суть метода заключается в отслеживании системных событий, описывающих действие программ, и сравнении их с шаблонами опасного поведения.

Отдельно можно отметить способы выявления вредоносного ПО, шифрующего свой код с целью скрытия от анализа сигнатурными методами. Так, в работе [4] предлагается определить наличие шифрования исполняемого файла, исследуя энтропию.

Несмотря на то, что данной проблематике посвящено довольно большое количество работ, проблема эффективности выявления ВШ в КС остается актуальной задачей. Так, за первый и второй квартал 2019 г. выявлено 516 781 инцидентов, связанных с атаками ВШ на системы пользователей [5, 6].

Описание методики выявления вирусов-шифровальщиков в компьютерных системах

Одним из способов выявления ВШ является анализ изменений параметров КС непосредственно во время начавшейся атаки. В работе [7] представлено исследование влияния ВШ на параметры КС, выявлены параметры КС, которые изменяются под воздействием активной фазы работы ВШ.

Суть метода следующая. В исследуемой КС устанавливалось специализированное ПО, позволяющее отслеживать и записывать показатели различных параметров системы в единицу времени, затем проводились измерения показателей в четырех различных режимах загрузки системы:

– низкая загрузка – система эксплуатируется в состоянии простоя, пользователь активные задачи не запускает;

– средняя загрузка – нормальная работа системы, пользователь запускает программы, работает с Интернетом и т. п.;

– высокая загрузка – выполнение ресурсоемких задач, например, архивирование большого объема данных;

– работа ВШ – непосредственная атака КС ВШ (в эксперименте использовались WannaCry, no more ransom и др.).

Полученные данные были проанализированы с использованием статистических тестов для выявления параметров КС, изменяющихся уникальным образом под воздействием ВШ. Если тест показывал значимость изменений значения параметра КС под влиянием ВШ в сравнении с низким, средним и высоким режимами загрузки, то показатели данного параметра можно использовать в дальнейшем для выявления ВШ.

В результате анализа были отобраны следующие группы параметров: «Память», «Проекты сборщика данных», «Процессор», «Система», «Физический диск».

После того как все неинформативные параметры были отсеяны, оставшиеся параметры были проверены на наличие линейной зависимости между параметрами как внутри группы, так и между параметрами из других групп. В результате для дальнейшей работы были отобраны 43 из 123 счетчиков.

Следующим этапом исследования было построение моделей с применением методов машинного обучения, позволяющих выявлять начавшуюся атаку ВШ на КС.

Для моделирования были использованы следующие алгоритмы машинного обучения:

– наивный байесовский классификатор;
– многослойная нейронная сеть;
– машина опорных векторов;
– алгоритм градиентного бустинга CatBoost.

При построении моделей были использованы программные пакеты и модели языка программирования Python. Общий перечень библиотек представлен в табл. 1.

Таблица 1

Алгоритмы и библиотеки машинного обучения

Алгоритм машинного обучения	Библиотека машинного обучения
Наивный байесовский классификатор	Scikit-learn
Машина опорных векторов	
Многослойная нейронная сеть	Keras
Алгоритм градиентного бустинга CatBoost	CatBoost

Генеральный датасет был разделен на обучающую и тестовую выборки в соотношении 80 : 20 %. Итоговый объем обучающей выборки равен 10 900 значений, объем оценивающего множества составляет 2 726 значений. В пределах выборок соотношение объектов, характеризующих нормальное состояние КС и атаки ВШ на КС, является приблизительно равным.

Анализ метрик качества

В качестве ключевых метрик эффективности моделей машинного обучения выбраны следующие типичные метрики: precision, recall, F1-метрика, accuracy, AUC [8]. Данные метрики, за исключением AUC, представляющей собой значение площади под ROC-кривой (кривая ошибок), получаются путем преобразований исходных значений матрицы ошибок. Матрица ошибок содержит первоначальные характеристики оценивания модели машинного обучения. Ошибочные классификации объекта, принадлежащего к показателям состояния КС в период вредоносной деятельности программ-вымогателей, и объекта, являющегося признаковым описанием нормальной работы КС, обозначаются как ошибки первого и второго рода соответственно. ROC-кривая зависит от показателей частоты правильной классификации моделью машинного обучения объектов положительного класса (TPR) и частоты неправильной классификации объектов отрицательного класса (FPR). Далее будут представлены матрицы ошибок, ROC-кривые и значения основных метрик качества для каждой из отобранных моделей. В расчет принимается то, что мощность оценивающего множества данных равняется 2 726 значениям, из которых 1 360 значений относятся к показателям КС в режиме полезной нагрузки, а другие 1 366 значений являются значениями параметров в период вредоносной активности ВШ. Далее представлены формулы, по которым вычисляются значения основных метрик качества моделей машинного обучения:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN};$$

$$precision = \frac{TP}{TP + FP};$$

$$recall = \frac{TP}{TP + FN};$$

$$F_{\beta} = (1 + \beta^2)\frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall},$$

где TP – число правильно классифицированных объектов положительного класса; TN – число правильно классифицированных объектов отрицательного класса; FN – число ошибочно классифицированных объектов положительного класса; FP – число ошибочно классифицированных объектов отрицательного класса; β определяет необходимую при вычислениях степень точности.

Построение и проверка моделей

Первая модель для обучения была основана на наивном байесовском алгоритме (НБА). Для реализации алгоритма была использована библиотека Scikit-learn, позволяющая строить модели с использованием различных модификаций НБА: GaussianNB, MultinomialNB, ComplementNB, BernoulliNB [9]. В табл. 2 приведены значения метрики *accuracy*, полученной в результате оценивания обученных моделей, разработанных на каждой из модификаций НБА.

Таблица 2

Значения метрики *accuracy* для моделей НБА

Модификация НБА	Значение метрики *accuracy*, %
GaussianNB	66
MultinomialNB	92
ComplementNB	89
BernoulliNB	61

Следующая модель была построена на основе машины опорных векторов (SVM). В работе использовалась библиотека Scikit-learn. В табл. 3 представлены значения метрики *accuracy* для моделей, построенных на различных вариантах исполнения вычислительного ядра машины опорных векторов.

Таблица 3

Значения метрики *accuracy* для моделей SVM

Вычислительное ядро машины опорных векторов (SVM)	Значение метрики *accuracy*, %
SVC	68
NuSVC	70
LinearSVC	76

Обучение моделей происходило без значительного изменения параметров модели, которые задаются по умолчанию [10].

Разработка моделей на нейронных сетях была организована с применением библиотеки машинного обучения Keras [11]. Регулируемые параметры конечных конфигураций нейронных сетей, среди которых производился отбор наилучшей, представлены следующим образом:

– задаваемое при построении количество слоев нейронной сети и количество нейронов в них: 2 слоя по 32 нейрона в каждом слое (32–32), 3 слоя по 64 нейрона в каждом слое (64–64–64);

– функции активации отдельных нейронов, присущих конкретному слою нейронной сети: ReLU, Softmax, Sigmoid.

Также для каждого из слоев спроектированных нейронных сетей был выставлен одинаковый уровень Dropout в значении, равном 0,5. Стек функций активации, присущих соответствующему слою нейронной сети, обозначен следующим образом: ReLU-Sigmoid или Sofmax-Softmax в случае построения нейронной сети с двумя скрытыми слоями и ReLU-ReLU-Sigmoid и Sofmax-Softmax-Softmax в случае проектирования нейронной сети с тремя скрытыми слоями. Функция потерь и алгоритм оптимизации целевого функционала взяты из перечня рекомендуемых для задачи классификации: binary_crossentropy и rmsprop соответственно.

В табл. 4 представлены значения метрики *accuracy* при оценивании моделей, соответствующих различным начальным конфигурациям нейронной сети.

Таблица 4

Значения метрики *accuracy* для моделей нейронной сети

Начальная конфигурация нейронной сети	Значение метрики *accuracy*, %
32–32, ReLU-Sigmoid	52
32–32, Sofmax-Softmax	48
64–64–64, ReLU-ReLU-Sigmoid	73
64–64–64, Sofmax-Softmax-Softmax	71

Что касается построения моделей на основе библиотеки градиентного бустинга CatBoost, то для анализа были выбраны реализации моделей, одна из которых удовлетворяла требованию наибольшей корректности классификации объектов, а другая – наименьшего времени обучения модели и вынесении решения о не встречаемых в обучении объектах. В табл. 5 приведены значения метрик для упомянутых реализаций моделей машинного обучения [12].

Таблица 5

Варианты реализации модели на основе алгоритма CatBoost

Параметры модели	Значение метрики *accuracy*, %
iterations=32, learning_rate=0.008, l2_leaf_reg=3, bagging_temperature=1, random_strength=1, one_hot_max_size=2, leaf_estimation_method='Newton'	97
iterations=8, learning_rate=0.07, boosting_type='Plain', bootstrap_type='Bernoulli', subsample=0.5, one_hot_max_size=20, rsm=0.5, leaf_estimation_iterations=5, max_ctr_complexity=1	95

Значение метрики *accuracy*, относящейся к модели, нацеленной на наилучшую точность вынесения вердиктов по объектам, которые не относятся к обучающему датасету, ненамного выше значения метрики *accuracy*, относящейся к модели, нацеленной на скорость принятия решения, что обусловлено хорошим качеством данных обучающего множества.

Итогом этапа стало формирование совокупности конечных реализаций моделей машинного обучения на основании значений метрик качества. В список лучших были включены следующие модели:

- наивный байесовский алгоритм: MultinomialNB;
- машина опорных векторов: LinearSVC;
- многослойная нейронная сеть: 64–64–64, ReLU-ReLU-Sigmoid;
- модель на основе алгоритма CatBoost, нацеленная на наилучшую точность отнесения объекта к тому или иному классу.

Выбранные реализации соответствующих моделей были подвергнуты последующему более детальному анализу, получены следующие результаты.

Для модели на основе алгоритма наивного байесовского классификатора матрица ошибок представлена в табл. 6.

Таблица 6

Матрица ошибок для модели на основе наивного байесовского классификатора

Класс	Actual Positive	Actual Negative	Суммарное количество значений
Predicted Positive	1 272	112	1 384
Predicted Negative	94	1 248	1 342
Суммарное количество значений	1 366	1 360	–

Матрица ошибок, относящаяся к модели, реализующей машину опорных векторов, представлена в табл. 7.

Таблица 7

Матрица ошибок для модели, реализующей машину опорных векторов

Класс	Actual Positive	Actual Negative	Суммарное количество значений
Predicted Positive	1 053	328	1 381
Predicted Negative	313	1 032	1 345
Суммарное количество значений	1 366	1 360	–

Далее рассмотрим первоначальные показатели эффективности модели, построенной на многослойной нейронной сети прямого распространения. Матрица ошибок приведена в табл. 8.

Таблица 8

Матрица ошибок для модели, построенной на многослойной нейронной сети прямого распространения

Класс	Actual Positive	Actual Negative	Суммарное количество значений
Predicted Positive	1 004	368	1 372
Predicted Negative	356	998	1 354
Суммарное количество значений	1360	1366	–

Что касается модели на основе алгоритма градиентного бустинга CatBoost, то результаты анализа первоначальных характеристик качества рассматриваемой модели приведены в табл. 9.

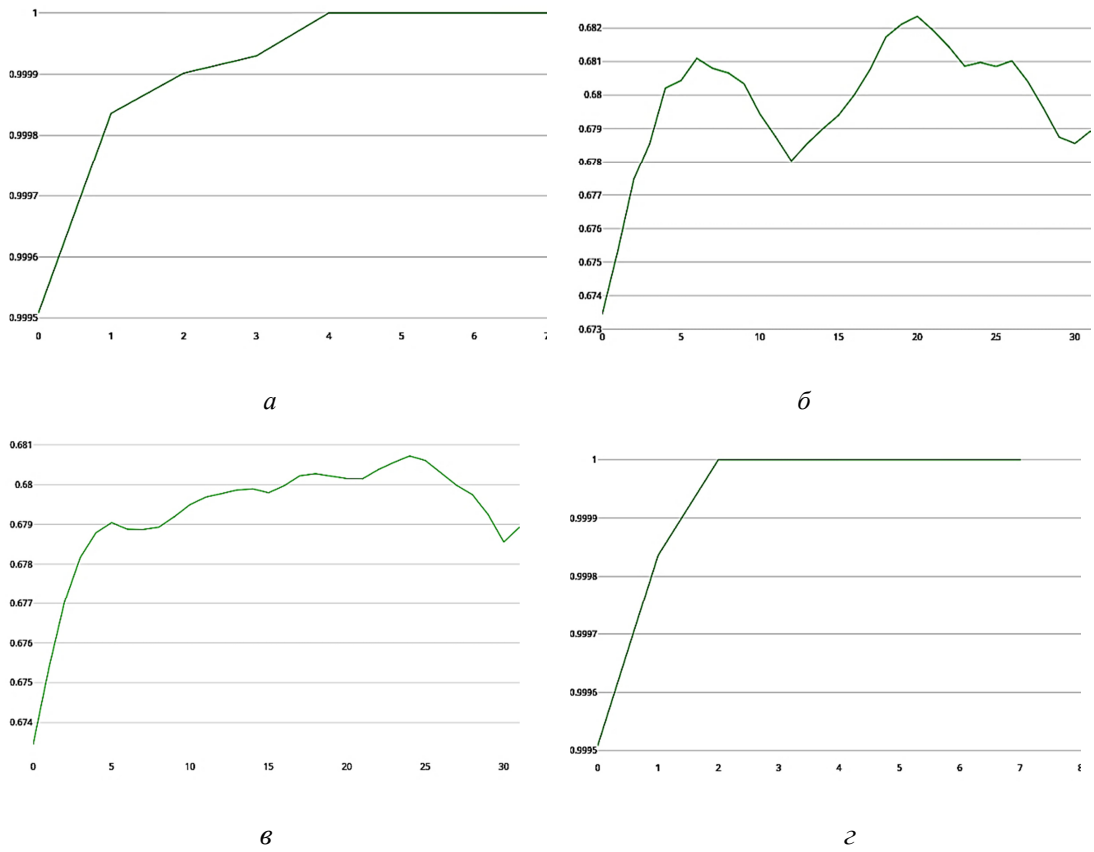
Таблица 9

Матрица ошибок для модели на основе алгоритма градиентного бустинга CatBoost

Класс	Actual Positive	Actual Negative	Суммарное количество значений
Predicted Positive	1 321	31	1 352
Predicted Negative	45	1 329	1 374
Суммарное количество значений	1 366	1 360	–

Как видно, значения *TP* и *TN* для модели на основе алгоритма градиентного бустинга CatBoost представляют собой наилучший результат по сравнению с теми же показателями для других моделей.

ROC-кривые, относящиеся к приведенным матрицам ошибок и соответствующие их значениям, представлены на рис.



а *б*

в *г*

ROC-кривые, относящиеся к приведенным матрицам ошибок:
а – наивный байесовский классификатор; *б* – многослойная нейронная сеть;
в – машина опорных векторов; *г* – алгоритм градиентного бустинга CatBoost

Далее рассмотрим основные показательные метрики качества моделей, сформированных из значений матриц ошибок. Помимо метрик эффективности классификации моделями машинного обучения, в табл. 10 приведено среднее время выполнения операций по классификации объектов тестирующего датасета, равного для всех экспериментальных моделей.

Таблица 10

Метрики качества моделей и среднее время выполнения операций по классификации большой совокупности данных

Алгоритм машинного обучения	Precision, %	Recall, %	F-мера, %	Accuracy, %	AUC	Время выполнения
Наивный байесовский алгоритм	91,9	93	92,4	92	0,919	0,0004992
Многослойная нейронная сеть	72,7	74,5	73,8	73	0,698	0,1283805
Машина опорных векторов	77	76,1	76,6	76	0,764	0,0009968
Алгоритм градиентного бустинга CatBoost	97,7	96,5	97,4	97	0,973	0,0079891

Таким образом, проведенный анализ показателей эффективности классификаций, присущих рассмотренным моделям, и вычислительной эффективности каждой из них показал, что для задачи выявления ВШ в КС наиболее подходящей является модель на основе алгоритма градиентного бустинга CatBoost.

Выводы

Сложность выявления ВШ современными средствами защиты информации, опасность полной или частичной потери данных делают проблемы борьбы с ВШ актуальной задачей современности. На помощь классическим методам борьбы с ВШ приходят методы противодействия, основанные на принципах эвристического анализа. При этом уровень угрозы заражения вирусами-шифровальщиками КС остается одной из главных проблем.

В статье представлен метод выявления ВШ в КС на основе анализа их поведенческих признаков. В результате анализа изменений значений параметров КС в разных режимах работы, в том числе во время атаки ВШ, были выявлены те параметры КС, на которые ВШ оказывает наибольшее влияние. На основании полученных результатов было проведено моделирование с применением различных алгоритмов машинного обучения. Сравнительный анализ моделей показал, что наилучший результат по выявлению ВШ в КС достигнут при использовании алгоритма градиентного бустинга CatBoost.

В целом полученные результаты могут свидетельствовать об эффективности данного метода выявления ВШ и могут быть использованы для противодействия ВШ в момент атаки на КС для предотвращения шифрования данных пользователей.

СПИСОК ЛИТЕРАТУРЫ

1. *Почему* НЕ стоит платить выкуп создателям троянов-вымогателей. URL: https://www.kaspersky.ru/blog/no-no-ransom/13518/ (дата обращения: 03.08.2019).

2. *Смирнов Д. В., Лубкин И. А.* Методика выявления криптовымогателей на основе отличия их поведения от штатных программ // Актуальные проблемы авиации и космонавтики. 2017. Т. 2. № 13. С. 236–238.

3. *Безмалый В.* Мониторинг подозрительной активности // Windows ITPRO. 2017. N. 6. P. 53.

4. *Алексеев И. В., Платонов В. В.* Определение наличия зашифрования исполняемого файла на основе анализа энтропии // Информатика и кибернетика. СПб.: Изд-во Санкт-Петерб. политехн. ун-та Петра Великого, 2016. С. 195–198.

5. *Развитие* информационных угроз в первом квартале 2019 года. Статистика. URL: https://securelist.ru/it-threat-evolution-q1-2019-statistics/94021/ (дата обращения: 03.08.2019).

6. *Развитие* информационных угроз во втором квартале 2019 года. Статистика. URL: https://securelist.ru/it-threat-evolution-q2-2019-statistics/94476/ (дата обращения: 03.08.2019).

7. *Назаров А. В., Марьенков А. Н., Калиев А. Б.* Выявление поведенческих признаков работы вируса-шифровальщика на основе анализа изменений значений параметров компьютерной системы // Прикаспийский журнал: управление и высокие технологии. 2018. № 1. С. 196–204.

8. *Метрики* в задачах машинного обучения. URL: https://habr.com/ru/company/ods/blog/328372/ (дата обращения: 11.05.2019).

9. *1. 9. Naive* Bayes. URL: https://scikit-learn.org/stable/modules/naive_bayes.html (дата обращения: 06.01.2019).

10. *Support* Vector Machines. URL: https://scikit-learn.org/stable/modules/svm.html (дата обращения: 09.01.2019).

11. *Keras:* The Python Deep Learning library. URL: https://keras.io/ (дата обращения: 09.01.2019).

12. *CatBoost* is a high-performance open source library for gradient boosting on decision trees. URL: https://catboost.ai/ (дата обращения: 09.01.2019).

Статья поступила в редакцию 12.09.2019

ИНФОРМАЦИЯ ОБ АВТОРАХ

Калиев Артур Борисович – Россия, 414056, Астрахань; Астраханский государственный университет; студент, направление подготовки «Прикладная математика и информатика. Математическое моделирование»; arthur19970824@gmail.com.

Марьенков Александр Николаевич – Россия, 414056, Астрахань; Астраханский государственный университет; канд. техн. наук, доцент; доцент кафедры информационной безопасности; marenkovan17@gmail.com.

METHOD OF DETECTING VIRUS-ENCODERS IN COMPUTER SYSTEM USING ANALYSIS OF THEIR BEHAVIOR

A. B. Kaliev, A. N. Marenkov

Astrakhan State University,
Astrakhan, Russian Federation

Abstract. The article considers the low efficiency of existing methods of ransomware fighting. The importance of developing new approaches to the ransomware identification in computer systems (CS) is substantiated. Heuristic analysis methods are considered as new approaches to ransomware detecting. A new technique for ransomware detecting is based on the analysis of changes in CS parameters. Using machine-learning methods there have been constructed models, which allow detecting ransomware attacks on the computer system. The aim of the experiment was to obtain a model that has the highest percentage of ransomware attacks detection and the least number of false triggering. The machine learning algorithms used for research are the following: naive Bayes classifier, multilayer neural network, support vector machine, CatBoost gradient boosting algorithm. To build the models training datasets written in Python programming language were used. The raining datasets were collected as a result of experiments with the most popular virus-encoders. The following typical metrics were selected as key metrics for the effectiveness of machine learning models: precision, recall, F1-metric, accuracy, AUC. In the course of experiments, the values of the error matrices were formed and the main indicators of the model quality metrics were obtained. In addition to the classification efficiency metrics, the average time for performing classification operations for each of the models is given. During the process of model training and testing it was revealed that the best model for detecting ransomware is that built on the CatBoost algorithm. The conclusions were drawn about the possibility of applying the approach to detect the ransomware attacks on various computer systems.

Key words: ransomware virus, virus detection, computer system, software, parameters, heuristic analysis methods, machine learning.

For citation: Kaliev A. B., Marenkov A. N. Method of detecting virus-encoders in computer system using analysis of their behavior. *Vestnik of Astrakhan State Technical University. Series: Management, Computer Science and Informatics*. 2020;1:41-49. (In Russ.) DOI: 10.24143/2072-9502-2020-1-41-49.

REFERENCES

1. *Pochemu NE stoit platit' vykup sozdateliam troianov-vymogatelei* [Why you should NOT pay a ransom to the creators of ransomware trojans]. Available at: https://www.kaspersky.ru/blog/no-no-ransom/13518/ (accessed: 03.08.2019).

2. Smirnov D. V., Lubkin I. A. Metodika vyiavleniia kriptovymogatelei na osnove otlichiia ikh povedeniia ot shtatnykh programm [Technique for identifying crypto ransomware based on difference between their behavior and regular programs]. *Aktual'nye problemy aviatsii i kosmonavtiki*, 2017, vol. 2, no. 13, p. 236-238.

3. Bezmalyi V. Monitoring podozritel'noi aktivnosti [Suspicious activity monitoring]. *Windows ITPRO*, 2017, no. 6, p. 53.

4. Alekseev I. V., Platonov V. V. Opredelenie nalichiia zashifrovaniia ispolniaemogo faila na osnove analiza entropii [Detecting encryption of executable file based on entropy analysis]. *Informatika i kibernetika*. Saint-Petersburg, Izd-vo Sankt-Peterb. politekhn. un-ta Petra Velikogo, 2016. Pp. 195-198.

5. *Razvitie informatsionnykh ugroz v pervom kvartale 2019 goda. Statistika* [Development of information threats in first quarter of 2019. Statistics data]. Available at: https://securelist.ru/it-threat-evolution-q1-2019-statistics/94021/ (accessed: 03.08.2019).

6. *Razvitie informatsionnykh ugroz vo vtorom kvartale 2019 goda. Statistika* [Development of information threats in second quarter of 2019. Statistics data]. Available at: https://securelist.ru/it-threat-evolution-q2-2019-statistics/94476/ (accessed: 03.08.2019).

7. Nazarov A. V., Mar'enkov A. N., Kaliev A. B. Vyiavlenie povedencheskikh priznakov raboty virusa-shifroval'shchika na osnove analiza izmenenii znachenii parametrov komp'iuternoi sistemy [Identification of behavioral signs of ransomware based on the analysis of changes in the parameters of computer system]. *Prikaspiiskii zhurnal: upravlenie i vysokie tekhnologii*, 2018, no. 1, pp. 196-204.

8. *Metriki v zadachakh mashinnogo obucheniia* [Metrics in machine learning problems]. Available at: https://habr.com/ru/company/ods/blog/328372/ (accessed: 11.05.2019).

9. *1. 9. Naive Bayes*. Available at: https://scikit-learn.org/stable/modules/naive_bayes.html (accessed: 06.01.2019).

10. *Support Vector Machines*. Available at: https://scikit-learn.org/stable/modules/svm.html (accessed: 09.01.2019).

11. *Keras: The Python Deep Learning library*. Available at: https://keras.io/ (accessed: 09.01.2019).

12. CatBoost is a high-performance open source library for gradient boosting on decision trees. Available at: https://catboost.ai/ (accessed: 09.01.2019).

The article submitted to the editors 12.09.2019

INFORMATION ABOUT THE AUTHORS

Kaliev Arthur Borisovich – Russia, 414056, Astrakhan; Astrakhan State University; Student, training area “Applied mathematics and computer science. Mathematical modeling”; arthur19970824@gmail.com.

Marenkov Alexandr Nikolaevich – Russia, 414056, Astrakhan; Astrakhan State University; Candidate of Technical Sciences, Assistant Professor; Assistant Professor of the Department of Information Security; marenkovan17@gmail.com.