http://dx.doi.org/10.35596/1729-7648-2021-19-2-91-99

Оригинальная статья
Original paper

УДК 004.492.3

ОБНАРУЖЕНИЕ DGA ДОМЕНОВ И ПРЕДОТВРАЩЕНИЕ BOTNET СРЕДСТВАМИ Q-ОБУЧЕНИЯ ДЛЯ POMDP

Я.В. БУБНОВ, Н.Н. ИВАНОВ

Белорусский государственный университет информатики и электроники
(г. Минск, Республика Беларусь)

Поступила в редакцию 29 января 2021

Аннотация. Предлагается эффективный метод предотвращения эксплуатации узлов компьютерной сети для организации botnet. Под botnet подразумевается совокупность устройств, объединенных через сеть Интернет с целью организации DDoS-атак, кражи данных, рассылки спама и других вредоносных действий. Описанный метод подразумевает детектирование сгенерированных доменных имен в DNS-запросах с помощью нейронной сети с параллельной организацией сверточных и двунаправленных рекуррентных слоев. Эффективность метода базируется на предположении, что для создания botnet используют генерируемые доменные имена для объединения. Эксперименты подтверждают, что предлагаемая нейронная сеть превосходит точность существующих аналогов на наборе данных UMUDGA. Вычисляется оценка качества распознавания сгенерированных доменных имен с помощью ROC-анализа для обученной нейронной сети. В статье также формулируется модель управления детекторами с помощью частично наблюдаемого марковского процесса принятия решений для блокировки зараженных узлов компьютерной сети. Предлагается поиск оптимальной политики для сформулированной модели средствами Q-обучения ценностных агентов. Производится сравнительный анализ по средней, минимальной и максимальной ценности принимаемых агентами действий в процессе взаимодействия с окружением.

Ключевые слова: алгоритм генерирования доменов, защита компьютерных сетей, рекуррентная нейронная сеть, частично наблюдаемый марковский процесс принятия решений, Q-обучение.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Для цитирования. Бубнов Я.В., Иванов Н.Н. Обнаружение DGA доменов и предотвращение botnet средствами Q-обучения для POMDP. Доклады БГУИР. 2021; 19(2): 91-99.

DGA DOMAIN DETECTION AND BOTNET PREVENTION USING Q-LEARNING FOR POMDP

YAKOV V. BUBNOV, NICK N. IVANOV

Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)

Submitted 29 January 2021

Abstract. An effective method for preventing the operation of computer network nodes for organizing a botnet is proposed. A botnet is a collection of devices connected via the Internet for the purpose of organizing DDoS attacks, stealing data, sending spam and other malicious actions. The described method implies the detection of generated domain names in DNS queries using a neural network with parallel organization of convolutional and bidirectional recurrent layers. The effectiveness of the method is based on the assumption that generated domain names are used to create a botnet for merging. Experiments confirm that the proposed neural network is superior to the accuracy of existing counterparts on the UMUDGA dataset. The estimation of the quality of recognition of generated domain names using ROC analysis is calculated for a trained neural network. The article also formulates a model for controlling detectors using a partially observable Markov decision-making process to block infected nodes of a computer network. The search for the optimal policy for the formulated model by means of Q-learning of value agents is proposed. A comparative analysis of the average, minimum and maximum value of actions taken by agents in the process of interacting with the environment is carried out.

Keywords: domain generation algorithm, computer network security, recurrent neural network, partially observable Markov decision process, Q-learning.

Conflict of interests. The authors declare no conflict of interests.

For citation. Bubnov Y.V., Ivanov N.N. DGA domain detection and botnet prevention using Q-learning for POMDP. Doklady BGUIR. 2021; 19(2): 91-99.

Введение

Предотвращение вредоносной активности узлов корпоративной компьютерной сети является одной из важных прикладных задач. Ее решение позволяет устранить целевые кибератаки, направленные как на отказ системы в целом, так и на кражу конфиденциальной информации. Кража данных может носить катастрофические последствия, приводящие к потере доверия у пользователей, утрате конкурентного преимущества на рынке, а также снижению стоимости акций компании.

Как отмечается в работе [1], помимо кражи информации, с помощью туннелирования системы доменных имен узлы компьютерной сети могут использоваться для организации DDoS-атак. В такой схеме зараженный узел получает команды для начала атаки от центрального сервера злоумышленника. Как отмечается в работе [2], стандартной методологией для организации botnet стало использование алгоритмов генерации доменных имен (Domain Name Generation Algorithms, DGAs). То есть вредоносное программное обеспечение использует генератор псевдослучайных чисел для создания доменных имен. Созданные доменные имена в дальнейшем используются для поиска центрального сервера, который может изменять свое месторасположение динамически. По этой причине блокировка центральных серверов на практике затруднена: множество возможных комбинаций огромно.

В данной работе предлагается архитектура сверточно-рекуррентной нейронной сети с параллельным размещением слоев и двунаправленными LSTM-ячейками для детектирования DGA-имен в DNS-запросах. Также в работе рассматривается POMDP-модель оптимального управления DGA-детекторами в компьютерной сети.

Сверточно-рекуррентная нейронная сеть для обнаружения DGA-имен в DNS запросах

Сформулируем задачу обнаружения DGA-имен следующим образом: пусть $\mathbf{X} = \{x_0, x_1, ..., x_n\}$ – запрашиваемое доменное имя, с которым ассоциирована метка $y \in \mathbf{Y}$ с вероятностью p_y. Тогда стоит задача нахождения функции

$$y : \mathbf{X} \to \mathbf{Y}. \tag{1}$$

Так как входы нейронной сети представляют собой вектор действительных чисел, требуется определить отображение символьного множества $\mathbf{X}$ в пространство R^k. Для этого воспользуемся модифицированным механизмом встраивания слов (англ. word embedding), предложенным в [3], где вместо словаря слов используется алфавит символов $\mathbf{A}$. Пусть дана функция $I(\mathbf{X})$, которая задает упорядочение множества $\mathbf{A}$. Тогда встраивание – это преобразование вектора символов в вектор целых чисел, где каждому символу $x \in \mathbf{X}$ в соответствие поставлен индекс из алфавита $\mathbf{A}$:

$$I : \mathbf{A} \to \left\{0, ..., |\mathbf{A}|\right\}; \text{cemb}(\mathbf{X}) = \left[I(x_1), I(x_2), ..., I(x_n)\right]. \tag{2}$$

Для краткости дальнейших выкладок определим полносвязный слой выражением

$$\Delta(\mathbf{h}) = D_{0,5}(a_{relu}(\mathbf{h})), \tag{3}$$

где $D_{0,5}$ – оператор прореживания связей, а a_{relu} – линейная функция ректификации.

В работе [4] рассматривается использование нейронных сетей с двунаправленными LSTM-ячейками для улучшения качества классификации в решении задачи выделения ключевых фраз – фраз, определяющих основную тему анализируемого текстового документа. Авторы отмечают, что при использовании двух рекуррентных слоев, состоящих из LSTM-ячеек, где один слой обучается на последовательности символов в прямом направлении, а второй – в обратном, наблюдается улучшение точности определения контекста фраз в тексте. В общем виде двунаправленный LSTM-слой формируется следующим образом:

$$\begin{aligned} &\vec{\mathbf{h}}^{(t)} = H(\mathbf{W}_{x\vec{h}}\mathbf{x}^{(t)} + \mathbf{W}_{\vec{h}\vec{h}}\vec{\mathbf{h}}^{(t-1)} + \mathbf{b}_{\vec{h}}); \overleftarrow{\mathbf{h}}^{(t)} = H(\mathbf{W}_{x\overleftarrow{h}}\mathbf{x}^{(t)} + \mathbf{W}_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{\mathbf{h}}^{(t-1)} + \mathbf{b}_{\overleftarrow{h}}), \\ &\mathbf{z}^{(t)}{}_{\text{bi-lstm}} = \mathbf{W}_{\vec{h}z}\vec{\mathbf{h}}^{(t)} + \mathbf{W}_{\overleftarrow{h}z}\overleftarrow{\mathbf{h}}^{(t)} + \mathbf{b}_z, \end{aligned} \tag{4}$$

т. е. производится объединение рекуррентных слоев, осуществляющих распространение в прямом и обратном направлениях, где $\mathbf{W}$ – матрица весов, а $\mathbf{b}$ – вектор смещений для каждой пары сопряженных слоев.

Воспользуемся данным подходом для усовершенствования архитектуры нейронной сети для классификации DGA-запросов. Для удобства определим двунаправленный слой, состоящий из N ячеек, следующим образом:

$$\left[a_{\text{bi-lstm}}\right]^N = \bigcup_t^N \mathbf{z}^{(t)}{}_{\text{bi-lstm}}. \tag{5}$$

Выражение (5) описывает композицию двунаправленных LSTM-ячеек, предназначенных для представления доменного имени. Таким образом, архитектура комбинированной сверточно-рекуррентной нейронной сети со 128 двунаправленными LSTM-ячейками будет выглядеть следующим образом:

$$\begin{aligned} &\mathbf{h}^{(1')} = D_{0,5}\left(\bigcup_{k=2}^{6} \text{conv1d}(cemb(\mathbf{X}); k)\right), \\ &\mathbf{h}^{(1'')} = D_{0,5}(\left[a_{\text{bi-lstm}}\right]^{128}(cemb(\mathbf{X})), \\ &y(\mathbf{X}) = a_{\text{softmax}}([\Delta]^3(\mathbf{h}^{(1')} \cup \mathbf{h}^{(1'')})), \end{aligned} \tag{6}$$

где $\text{conv1d}(\mathbf{H}; k)$ – одномерная свертка с ядром размера k.

POMDP-модель для блокировки зараженных узлов

Сформулируем математическую модель POMDP относительно узла компьютерной сети, для которого принимается решение о блокировке. Условимся, что данное решение принимается на основании присутствия DGA-имен в запросах, отправляемых с узла. Для этого на пограничном DNS-сервере производится накопление информации о вредоносной активности. Для обеспечения оптимальной доступности системы DNS проверяются не все проходящие DNS-запросы, а только некоторая их часть. Таким образом, стоит оптимизационная задача обеспечить наиболее эффективное использование ресурсов DNS-сервера для определения узлов, требующих блокировки. Здесь и далее под блокировкой узла будем понимать отключение узла от внешней сети.

Определим POMDP-модель как кортеж $(\mathbf{S},\mathbf{A},T,\mathbf{\Omega},O,R)$, где $\mathbf{S}$ – конечное множество состояний системы; $\mathbf{A}$ – конечное множество действий, доступных из состояния $s \in \mathbf{S}$; $T(s' \mid s,a)$ – функция условных вероятностей перехода из одного состояния в другое; $\mathbf{\Omega}$ – конечное множество наблюдений; $O(o \mid s',a)$ – функция условных вероятностей наблюдений; R – функция вознаграждения.

Введем понятие степени внимания детектора к узлу ρ, и определим его как отношение запросов, анализируемых детектором, к общему числу запросов от узла.

В контексте данной модели для каждого DNS-запроса может быть произведено одно из действий: получение состояния детектора a_{acc}; увеличение степени внимания к детектору a_{inc}; понижение степени внимания к детектору a_{dec}; изоляция узла для предотвращения организации botnet a_{blk}; продолжение функционирования системы без изменений a_{ulk}. Таким образом, конечное множество действий можно задать множеством $\mathbf{A} = \{a_{acc}, a_{inc}, a_{dec}, a_{blk}, a_{ulk}\}$.

Блокировка и разблокировка являются конечными действиями, при которых система завершает работу. Каждый узел сети находится либо в состоянии «заражен» – s_{inf}, либо в состоянии «здоров» – s_{hlt}, множество состояний $\mathbf{S} = \{s_{inf}, s_{hlt}\}$.

Так как состояние POMDP-модели для агента неизвестно, анализировать систему возможно только благодаря наблюдениям, образованным непрерывным векторным пространством: $\mathbf{\Omega} = \mathbf{P} \times \mathbf{M}$, где $\mathbf{P} = [0,1]$ – вектор, определяющий диапазон допустимых значений для степени внимания детектора к узлу ρ, а $\mathbf{M} = [0,1]$ – вектор, определяющий диапазон допустимых значений оценки вредоносной активности $\tilde{\mu}$. Оценка $\tilde{\mu}$ формируется DGA-детектором. То есть наблюдение системы представлено парой значений: $o = \{\rho, \tilde{\mu}\}$.

Функцию условных вероятностей переходов из одного состояния в другое определим так, что при доступе к детектору туннелирования, увеличении или уменьшении степени внимания к детектору система не меняет свое состояние. Изменение состояния происходит только при применении действия по блокировке или разблокировке узла:

$$T(s' \mid s,a) = \begin{cases} 1, & a \in \{a_{blk}, a_{ulk}\} \\ 0, & \textit{иначе} \end{cases}, \tag{7}$$

то есть система детерминирована относительно переходов из состояния в состояние.

Функция условных вероятностей наблюдений определяется исходя из степени точности детектора q. Тогда вероятность наблюдения вредоносной активности в состоянии s_{inf} будет составлять q, а в состоянии s_{hlt}: $1-q$:

$$O(o \mid s',a) = \begin{cases} q, & s' = s_{inf}; a = a_{acc} \\ 1-q, & s' = s_{hlt}; a = a_{acc} \\ 1, & a \in \{a_{inc}, a_{dec}\} \\ 0, & \textit{иначе} \end{cases}. \tag{8}$$

В случае увеличения или уменьшения степени внимания детектора к узлу, наблюдение всегда детерминировано. При выполнении одного из конечных действий окружение не наблюдается.

Функцию вознаграждения определим таким образом, что агент получает негативное вознаграждение за доступ к детектору и изменение степени внимания, т. е. чем больше агент проводит времени за попытками получить дополнительную информацию об узле, тем меньше будет финальное вознаграждение. Также при ошибке системы введем дополнительный штраф. Функцию вознаграждения представим матрицей $\mathbf{R} = \mathbf{S}^T \times \mathbf{A}$, где $\mathbf{R}_{i,j}$ определяет вознаграждение за a_i действие в состоянии s_j. Функция вознаграждения R, таким образом, задается следующим выражением:

$$R(s,a) = \begin{cases} \mathbf{R}_{s,a}, & s \in \mathbf{S}, a \in \mathbf{A} \\ 0 & иначе \end{cases}. \tag{9}$$

Для эксперимента будем рассматривать следующие вознаграждения за действия агента:

$$\mathbf{R} = \begin{bmatrix} -0,2 & -0,8 & -1,0 & 1,0 & -1,0 \\ -0,2 & -0,8 & -1,0 & -1,0 & 1,0 \end{bmatrix}. \tag{10}$$

Агент для поиска оптимальной политики POMDP-модели

Проведем сравнительный анализ моделей поиска оптимальной политики для описанной POMDP-проблемы $(\mathbf{S}, \mathbf{A}, T, \mathbf{\Omega}, O, R)$.

Первая рассматриваемая модель – DQN (англ. Deep Q-Network) – впервые предложена в работе [5] для марковского процесса принятия решения (MDP). Сформулируем данную модель для решения POMDP-проблемы. Исходя из того, что агент взаимодействует с окружением $\xi = (\mathbf{S}, \mathbf{A}, T, \mathbf{\Omega}, O, R)$ путем выполнения действий $a \in \mathbf{A}$, ставится задача нахождения оптимальной политики для максимизации вознаграждения с учетом того, что вознаграждение на каждом шаге дисконтируется величиной $\gamma < 1$:

$$Q^*(s,a) = E_{s' \sim \xi}\left[r + \gamma \max_{a'} Q^*(s',a') \,|\, s,a \right]. \tag{11}$$

Для решения уравнения (11) предлагается использовать глубокую нейронную сеть, обученную для максимизации вознаграждения $Q^*(s,a)$.

Воспользуемся данным подходом для нахождения оптимальной политики действий для выявления зараженных узлов в компьютерной сети. Для этого определим нейронную сеть, которую обучим по методике, предложенной в [5]. На входы $\mathbf{X}$ такой нейронной сети подается множество наблюдений системы $\mathbf{\Omega}$, а на выходе $\mathbf{Y} = y(\mathbf{X})$ формируется действие $\mathbf{A}$, которое должен выполнить агент системы для максимизации вознаграждения. Тогда нейронная сеть описывается следующим образом:

$$y_{dqn}(\mathbf{X}) = [\Delta]^4(\mathbf{X}), \tag{12}$$

где для каждого скрытого полносвязного слоя используется оператор прореживания $D_{0,5}$ для регуляризации нейронной сети.

Вторая рассматриваемая модель – DDQN (англ. Dueling Deep Q-Network) – описана в работе [6], где предлагается подход, позволяющий бороться с чрезмерной оптимистичностью модели DQN-агента. Авторы предлагают использовать две модели: первая обучается функции $V(s;\theta)$, оценивающей текущее состояние агента; вторая обучается функции $A(s,a;\theta)$, оценивающей преимущество принимаемого действия в текущем состоянии.

Опишем нейронную сеть для оценки преимущества действий агента, в которую на вход подается вектор наблюдений $\mathbf{\Omega}$ окружения ξ таким образом, что на последнем слое производится усреднение по количеству выходов:

$$Avg(\mathbf{z}) = \mathbf{z}_{:,0} + \mathbf{z}_{:,1:} - \text{avg}(\mathbf{z}_{:,1:}); y_{ddqn}(\mathbf{X}) = Avg\left([\Delta]^4(\mathbf{X})\right). \tag{13}$$

Для обучения функции $y_{ddqn}(\mathbf{X})$, оценивающей состояние агента, используется ранее сформулированная нейронная сеть (12).

Третья рассматриваемая модель – DDPG (англ. Deep Deterministic Policy Gradient) – предложена в работе [7]. Суть модели заключается в использовании двух нейронных сетей для обучения оптимальной политике: актора и критика. Причем акторная нейронная сеть обучается параметризованной функции $\mu(s,\theta)$, определяющей текущую политику взаимодействия с окружением, путем отображения состояний к конкретным действиям. Результат данной функции используется при обновлении весов акторной нейронной сети в рамках алгоритма градиентного спуска.

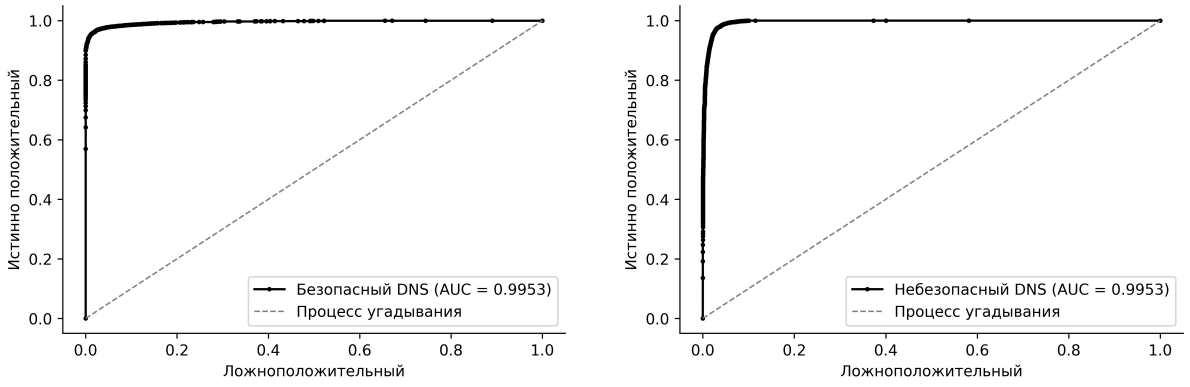
В качестве архитектуры нейронной сети для критика используется модель (12), а для актора используется модель, на вход которой подается объединенный вектор действий и наблюдений. Поскольку модель формулируется относительно непрерывного пространства действий, применим функцию argmax к выходам сети:

$$y^*\left(\mathbf{X}_A \cup \mathbf{X}_\Omega\right) = \arg\max\left([\Delta]^4(\mathbf{X}_A \cup \mathbf{X}_\Omega)\right). \quad (14)$$

Результаты и обсуждение

Все описываемые далее результаты получены в среде Google Colab, где PODMP-модель описана с помощью библиотеки OpenAI Gym, а нейросетевые модели определены средствами библиотеки TensorFlow и Keras-RL.

Обучение предложенной нейросети (6) для обнаружения DGA-имен производится на обучающем наборе данных UMUDGA [8], содержащем сгенерированные доменные, доменные имена популярных сервисов сети Интернет, а также доменные имена, используемые в сети распространения контента. Оценка качества обученной модели осуществляется на тестовом наборе данных UMUDGA, после чего строятся ROC-кривые для класса безопасных и DGA доменных имен. Результаты оценки качества классификации представлены на рис. 1, где диагональной штриховой линией изображен график эфемерной модели, для которой классификация представляет собой процесс угадывания.



a *b*

Рис. 1. ROC-кривые обученной нейронной сети для DNS-запросов: *a* – безопасных; *b* – DGA
Fig. 1. ROC curves of trained neural network for DNS queries: *a* – safe; *b* – DGA

Для оценки качества классификации DGA-имен используется точность, чувствительность, специфичность, а также F_1-метрика. Помимо предложенной модели (6), включим в сравнение существующие модели: рекуррентную нейронной сеть [9]; сверточно-рекуррентную сеть с последовательным размещением слоев [10]; сверточно-рекуррентную сеть с параллельным размещением слоев [11].

Результаты оценки моделей на тестовом наборе UMUDGA приведены в табл 1. Из результатов видно, что предложенная модель превосходит сразу по нескольким параметрам модель, предложенную в [11], считавшуюся лучшим на данный момент способом определения DGA-имен.

Таблица 1. Метрики бинарной классификации определения DGA-имен для набора UMUDGA
Table 1. Binary classification metrics of DGA-name detection for UMUDGA dataset

Модель (model)	Точность (accuracy)	Чувствительность (precision)	Специфичность (recall)	F_1
Woodbridge et al. [9]	0,5000	0,0000	0,0000	0,0000
Vosoughi et al. [10]	0,9599	0,9441	0,9777	0,9606
Highnam et al. [11]	0,9643	0,9486	**0,9818**	0,9645
Предложенная модель (6)	**0,9654**	**0,9581**	0,9734	**0,9657**

Обучение Q-нейросетей агентов: для оптимизации используется алгоритм Adam. Для предотвращения влияния начальных наблюдений окружения на результаты обучения нейронной сети вводится 1000 шагов разогрева, в рамках которых происходит лишь накопление исторических результатов для последующего использования при обучении. Результаты обучения Q-нейросетей представлены на рис. 2. Серым цветом на рисунках обозначен график средней ценности действий $Q(s,a)$, а черным – значение скользящего среднего ошибки обучения с окном в 50 значений.

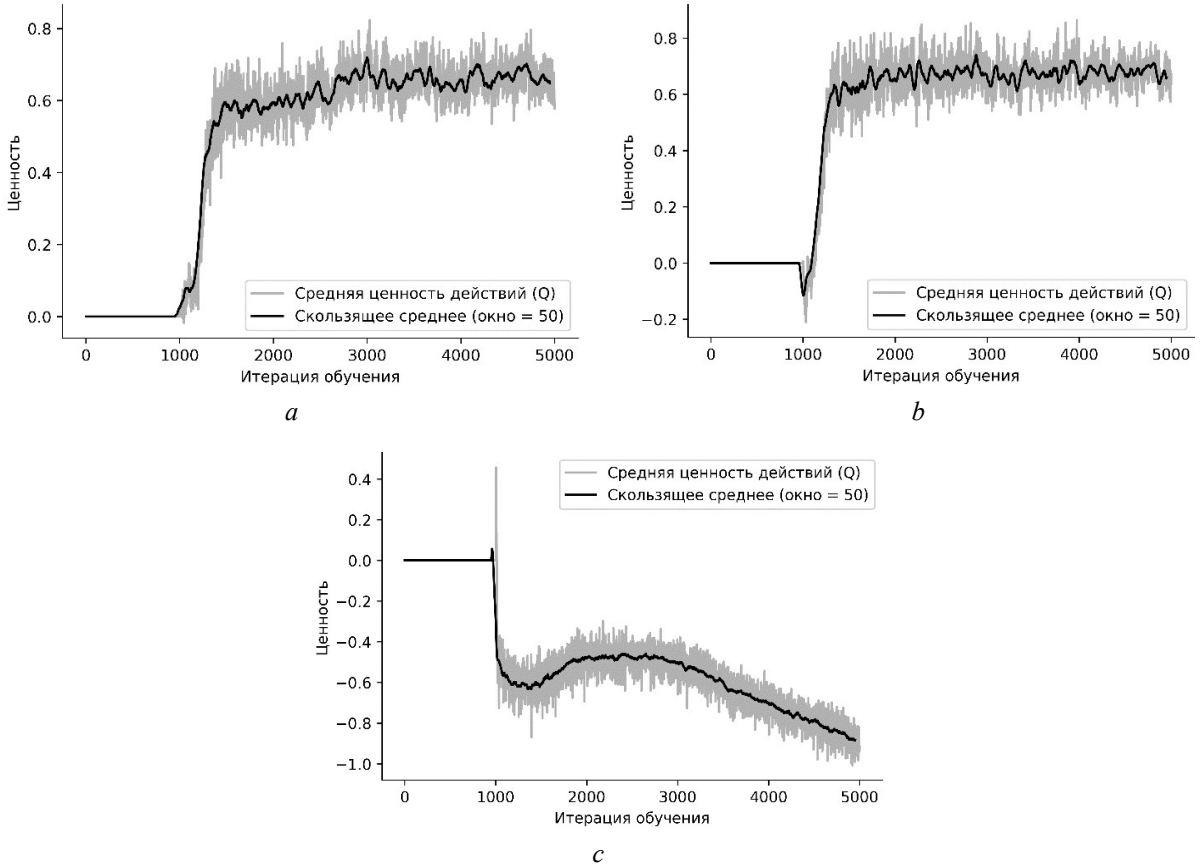


Рис. 2. Ценность действий агентов в процессе обучения: *a* – DQN; *b* – DDQN; *c* – DDPG
Fig. 2. Mean value of actions of agents during training: *a* – DQN; *b* – DDQN; *c* – DDPG

Из рис. $2, a, b$ видно, что для DQN- и DDQN-агента рост средней ценности принимаемых действий $Q(s,a)$ происходит практически сразу после итераций разогрева. Для DDPG-агента (см. рис. $2, c$) наблюдается снижение средней ценности действий по мере обучения нейронной сети.

Для определения оптимальной политики определения зараженных узлов в компьютерной сети сравнивается среднее, минимальное и максимальные вознаграждение. Вознаграждение рассчитывается для 1000 эпизодов взаимодействия с окружением. Результаты сравнения рассмотренных агентов представлены в табл. 2.

Таблица 2. Результаты взаимодействия агентов с окружением
Table 2. Results of interaction of agents with environment

Агент (agent)	Среднее вознаграждение (mean reward)	Минимальное вознаграждение (minimum reward)	Максимальное вознаграждение (maximum reward)
DQN	0,6140	**–1,0000**	**1,0000**
DDQN	**0,6160**	**–1,0000**	**1,0000**
DDPG	–19,9999	–19,9999	–19,9999

Как видно из табл. 2, лучшие результаты демонстрируют агенты DQN и DDQN, тогда как агент DDPG не научился взаимодействовать с окружением, в результате среднее, минимальное и максимальное вознаграждение мало. Исходя из метрик оценки эффективности политик агентов, наиболее эффективным агентом является DDQN, использующий дуэльную архитектуру нейронной сети для выбора последующего действия.

Выводы

В работе представлен метод обнаружения сгенерированных доменных имен с помощью нейронной сети с параллельной организацией слоев свертки и рекуррентных слоев с двунаправленными LSTM-ячейками. Данная архитектура превосходит существующие аналоги по точности детектирования DGA-имени на наборе данных UMUDGA. Также сформулирована POMDP-модель окружения компьютерной сети с детекторами DGA-имен для блокирования зараженных узлов. Для POMDP-модели предложен метод поиска оптимальной политики взаимодействия с окружением агентным подходом. Показано, что DDQN-агент позволяет добиться максимизации ценности принимаемых действий при управлении детекторами.

Список литературы / References

1. Kolias C., Kambourakis G., Stavrou A., Voas J. DDoS in the IoT: Mirai and Other Botnets. *Computer.* 2017;50:80-84.
2. Patsakis C., Casino F., Katos V. Encrypted and covert DNS queries for botnets: Challenges and countermeasures. *Computers & Security.* 2020;88:101614.
3. Watson D., Zalmout N., Habash N. Utilizing Character and Word Embeddings for Text Normalization with Sequence-to-Sequence Models. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* Brussels, Belgium: Association for Computational Linguistics; 2018: 837-843 DOI:10.18653/v1/D18-1097.
4. Basaldella M., Antolli E., Serra G., Tasso C. Bidirectional LSTM Recurrent Neural Network for Keyphrase Extraction. *Digital Libraries and Multimedia Archives* (eds. Serra, G. & Tasso, C.) Cham: Springer International Publishing; 2018: 180-187. DOI:10.1007/978-3-319-73165-0_18.
5. Mnih V., Kavukcuoglu K., Silver D., Rusu A.A., Veness J., Bellemare M.G., Graves A., Riedmiller M., Fidjeland A.K., Ostrovski G., Petersen S., Beattie C., Sadik A., Antonoglou I., King H., Kumaran D., Wierstra D., Legg S., Hassabis D. Human-level control through deep reinforcement learning. *Nature.* 2015;518:529-533.
6. Wang Z., Schaul T., Hessel M., Van Hasselt H., Lanctot M., De Freitas N. Dueling network architectures for deep reinforcement learning. *Proceedings of the 33rd International Conference on International Conference on Machine Learning. Vol. 48.* New York, NY, USA: JMLR.org; 2016: 1995-2003.
7. Lillicrap T.P., Hunt J.J., Pritzel A., Heess N., Erez T., Tassa Y., Silver D., Wierstra D. Continuous control with deep reinforcement learning. *Proceedings of the 4th International Conference on Learning Representations.* San Juan, Puerto Rico; 2016.
8. Zago M., Gil Pérez M., Martínez Pérez G. UMUDGA: A dataset for profiling DGA-based botnet. *Computers & Security.* 2020;92:101719.
9. Woodbridge J., Anderson H.S., Ahuja A., Grant D. Predicting Domain Generation Algorithms with Long Short-Term Memory Networks. *Computing Research Repository.* 2016; 1-13 arXiv:1611.00791.
10. Vosoughi S., Vijayaraghavan P., Roy D. Tweet2Vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder. *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval.* New York, NY, USA: Association for Computing Machinery; 2016: 1041-1044. DOI:10.1145/2911451.2914762.
11. Highnam K., Puzio D., Luo S., Jennings N.R. Real-Time Detection of Dictionary DGA Network Traffic using Deep Learning. *Computing Research Repository.* 2020; 1-12 arXiv:2003.12805.

Вклад авторов

Бубнов Я.В. сформулировал нейросетевую модель детектора, модель частично наблюдаемого марковского процесса принятия решений.

Иванов Н.Н. выполнил постановку научной проблематики и обобщение результатов исследования.

Authors' contribution

Bubnov Y.V. formulated neural network model of the detector, the model of partially observable Markov decision process.

Ivanov N.N. completed the formulation of scientific problems and generalization of the research results.

Сведения об авторах

Бубнов Я.В., магистр технических наук, аспирант кафедры электронных вычислительных машин Белорусского государственного университета информатики и радиоэлектроники.

Иванов Н.Н., к.ф.-м.н., доцент, доцент кафедры электронных вычислительных машин Белорусского государственного университета информатики и радиоэлектроники.

Адрес для корреспонденции

220013, Республика Беларусь,
г. Минск, ул. П. Бровки, 6,
Белорусский государственный университет
информатики и радиоэлектроники;
тел. +375-29-757-28-23;
e-mail: girokompass@gmail.com
Бубнов Яков Васильевич

Information about the authors

Bubnov Y.V., M.Sc., Postgraduate student at the Electronic Computing Machines Department of the Belarusian State University of Informatics and Radioelectronics.

Ivanov N.N., PhD, Associate Professor, Associate Professor at the Electronic Computing Machines Department of the Belarusian State University of Informatics and Radioelectronics.

Address for correspondence

220013, Republic of Belarus,
Minsk, P. Brovka str., 6,
Belarusian State University
of Informatics and Radioelectronics;
tel. +375-29-757-28-23;
e-mail: girokompass@gmail
Bubnov Yakov Vasil'evich