УДК 004.89 DOI: 10.17587/it.25.691-697

А. Б. Менисов, канд. техн. наук, e-mail: men.arty@yandex.ru,
И. А. Шастун, канд. техн. наук, e-mail: shastunivan1982@gmail.com,
Военно-космическая академия имени А. Ф. Можайского,
С. Ю. Капицын, канд. техн. наук, доц., e-mail: wolf76@inbox.ru,
Военная академия Генерального Штаба ВС РФ

Подход к выявлению вредоносных сайтов сети Интернет на основе обработки лексических признаков адресов (URL) и усредненного ансамбля моделей

В настоящее время выявление и блокирование доступа к вредоносным сайтам сети Интернет выполняется, в основном, путем включения URL-адресов в черные списки. Однако черные списки не могут быть исчерпывающими, и с их помощью нет возможности выявлять вновь созданные вредоносные сайты. Целью статьи является разработка нового подхода для выявления вредоносных сайтов на основе усредненного ансамбля моделей, позволяющего улучшить точность выявления. В разработанном подходе по сравнению с современными техническими решениями в сфере информационной безопасности учтены лексические признаки адресной строки (URL) вредоносных сайтов, которую злоумышленники пытаются видоизменить под адрес известного или безопасного сайта.

Также авторы приводят сравнение результатов выявления вредоносных URL-адресов, полученных современными и разработанными подходами. Статья будет полезна не только исследователям в области машинного обучения, но и специалистам в отрасли кибербезопасности.

Ключевые слова: *информационная безопасность, вредоносные веб-сайты, машинное обучение*

Введение

Технологические достижения современного общества сочетаются с новыми угрозами информационной безопасности, такими как вредоносные сайты, распространяющие негативный и экстремистский контент, сайты, через которые реализуются такие атаки, как взлом, социальная инженерия, фишинг, человек посередине, SQL-инъекции, DOS- и DDOS-атаки и многие другие [1], которые в конечном итоге приводят к краже личных данных или даже к установке вредоносных программ в корпоративные системы.

С учетом разнообразия типов атак, проводимых с помощью вредоносных сайтов, становится сложно спроектировать надежные системы для обнаружения нарушений информационной безопасности. Например, большинство фишинговых атак реализуются путем распространения URL-адресов вредоносных сайтов (или распространение таких URL-адресов является одним из этапов атаки) [2].

URL-адрес (Uniform Resource Locator) является глобальным адресом документов и других ресурсов в сети Интернет [3]. URL имеет два основных компонента: 1) идентификатор протокола, который указывает, какой протокол использовать; 2) имя ресурса, указывающего IP-адрес или имя домена, где расположен ресурс.

URL-адреса, используемые для атак, в рамках данной статьи будем называть вредоносными URL-адресами. Отметим, что около трети всех веб-сайтов являются потенциально вредоносными [4].

Наиболее распространенным методом обнаружения вредоносных URL-адресов является метод включения в черные списки. Черные списки — это база URL-адресов, которые в прошлом были определены как вредоносные. Эта база данных формируется с течением времени (часто с помощью краудсорсинговых решений, например, PhishTank [5]). Такой подход очень прост в реализации в связи с простым формированием запросов к базе данных и имеет очень низкий уровень ложноположи-

тельных результатов [6]. Однако невозможно оперативно поддерживать исчерпывающий список вредоносных URL-адресов, тем более что новые URL-адреса создаются каждый день с высокой интенсивностью. Учитывая невозможность пополнения черного списка URL-адресами, сформированными сервисами сокращения ссылок, и с целью замаскировать вредоносный URL-адрес злоумышленники часто изменяют URL-адреса следующими путями [7, 8]: обфускацией хоста IP-адресом, обфускацией хоста другим доменом, обфускацией хоста большим именем и использованием неправильного написания.

Таким образом, метод занесения в черный список имеет серьезные ограничения и не эффективен для выявления вновь созданных URL-адресов. Чтобы преодолеть эти проблемы, в последнее десятилетие исследователи применяли методы машинного обучения для выявления вредоносных URL-адресов [7, 9—16].

1. Постановка задачи

Задано множество URL-адресов U, множество их состояний $Y = \{0, 1\}$, и существует целевая функция $y^* : U \to Y$, значения которой $y_i = y^*(u_i)$ известны только на конечном подмножестве объектов $\{u_1, ..., u_n\} \subset U$, причем при $y_i = 1$ сайт является вредоносным, $y_i = 0$ — безопасным. Пары "объект—состояние" (u_i, y_i) являются прецедентами. Совокупность пар прецедентов $U^l = (u_i, y_i)_{i=1}^l$ является обучающей выборкой для восстановления зависимости y^*.

Задача выявления вредоносных URL-адресов заключается в том, чтобы построить решающую функцию $a : U \to Y$, которая приближала бы целевую функцию $y^*(u)$, причем не только на объектах обучающей выборки, но и на всем множестве U, т. е. была бы способной классифицировать с точки зрения вредоносности произвольный URL-адрес $u \in U$. Вероятность правильной классификации и вероятности ошибок задают средний риск ошибочного выявления вредоносных URL:

$$П = M[C] = p_0 \cdot 0 + p_1 C_1 + ... + p_i C_i, \qquad (1)$$

где C — множество ошибок выявления, $\langle C_1, ..., C_i \rangle$ — ошибки выявления, i — число классов состояний URL-адресов, p_0 — вероятность правильного решения, $\langle p_1, ..., p_i \rangle$ — вероятность ошибок.

Таким образом, задача выявления вредоносных URL-адресов по двум классам (безопасные и вредоносные URL-адреса) заключается в следующем:

$$П = M[C] = p_0 \cdot 0 + p_1 C_1 + ... + p_2 C_2 \to \min, \qquad (2)$$

где p_1 — вероятность ошибки первого рода (когда ошибочно классифицирован безопасный URL-адрес как вредоносный), а p_2 — вероятность ошибки второго рода (когда происходит пропуск вредоносного URL-адреса).

2. Лексические признаки вредоносных URL-адресов

Практически в любой задаче классификации возникают вопросы: какие признаки использовать, а какие нет; нужно ли как-то преобразовывать исходные признаки. Для задачи выявления вредоносных URL-адресов вопрос извлечения лексических признаков из адресной строки (URL) является наиболее актуальным.

Признак f URL-адреса u — это результат измерения некоторой характеристики URL-адреса. Формально признаком называется отображение $f: U \to D_f$, где D_f — множество допустимых значений признака. Лексические признаки — это результат анализа строки URL, которая состоит из совокупности знаков, слов, взаимосвязи слов, характеризующих принадлежность к вредоносному или безопасному URL-адресу.

Наиболее часто используемые лексические признаки включают статистические параметры строки URL, такие как длина URL, длина каждого из компонентов URL (имени хоста, домена верхнего уровня, основного домена и т. д.) [17].

В зависимости от того, что содержит строка URL, существует возможность идентифицировать вредоносную природу URL-адреса. Например, существует метод скрытия, когда вредоносные URL-адреса пытаются выдать за безопасные, имитируя их имена и добавляя к ним незначительные дополнения в строке URL, или злоумышленники добавляют схожие слова в разные части домена.

Для расширения признаков в работе использовались значения TF-IDF имени домена и поддоменов.

TF-IDF — статистическая мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов или корпуса [30]. Вес некоторого слова $z_{i,j} \in Z$ пропорционален частоте употребления i-го слова в j-м документе и обратно пропорционален частоте употребления i-го

слова во всех документах коллекции и вычисляется как

$$z_{i,j} = TF_{i,j} \cdot \log\left(\frac{D}{d_i}\right), \qquad (3)$$

где $TF_{i,j}$ (term frequency — частота i-го слова $0 \leqslant i \leqslant |Z|$) в j-м документе) — отношение числа вхождений некоторого слова к общему числу слов документа, и $\log\left(\frac{D}{d_i}\right)$ — частотная характеристика, с которой некоторое i-е слово встречается в d_i документах коллекции мощностью D.

Такой подход уменьшает вес широкоупотребительных слов. Это актуально, так как все домены можно разделить по географической и специальной принадлежности, и для задачи выявления этот признак является малоинформативным.

Таким образом, извлечение лексических признаков из адресной строки (URL) состоит из нижеописанной последовательности действий.

Шаг 1. Подсчет числа f_1 разделителей (точек) в доменных и поддоменных частях URL-адреса. Является количественным признаком: $f_1 \in \mathbb{R}$.

Шаг 2. Подсчет числа f_2 других разделителей (';' ,'_', '?', ' = ', '&'). Является количественным признаком: $f_2 \in \mathbb{R}$.

Шаг 3. Проверка кодирования URL-адреса. Если в качестве альтернативы имени домена в URL-адресе используется IP-адрес, например, "http://125.98.3.123/fake.html", то это информативный признак вредоносного сайта. Иногда IP-адрес даже преобразуется в шестнадцатеричный код, как показано в следующей ссылке: "http: //0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html". Является бинарным признаком: $f_3 = \{0, 1\}$.

Шаг 4. Проверка на использование символа "@" в URL заставляет браузер игнорировать все, что предшествует символу "@", а реальный адрес часто следует за символом "@". Является бинарным признаком: $f_4 = \{0, 1\}$.

Шаг 5. Проверка на наличие символа "//" в пути URL означает, что пользователь будет перенаправлен на другой веб-сайт. Пример таких URL-адресов: "http://www.legitimate.com//http://www.phishing.com". Является бинарным признаком: $f_5 = \{0, 1\}$.

Шаг 6. Подсчет длины f_6 составляющих URL-адресов. Является количественным признаком: $f_6 \in \mathbb{R}$.

Шаг 7. Вычисление TF-IDF ключевых слов доменов и поддоменов. Являются количественными признаками: $\langle f_7, ..., f_m \rangle \in \mathbb{R}$, где m — число частей адресной строки (URL).

Пример. Если URL-адрес содержит 10 слов, и слово "fake" встречается в нем два раза, то частота слова для слова "fake" в документе будет 0,2 (2/10). Вычислим логарифм отношения числа всех URL-адресов к числу URL-адресов, содержащих слово "fake". Таким образом, если "fake" содержится в 1000 документах из 10 000 000 документов, то IDF будет равной: log(10 000 000/1000) = 4. Для расчета окончательного значения веса слова необходимо TF умножить на IDF. В данном примере TF-IDF-вес для слова "fake" в выбранном документе будет равен: $0{,}2 \cdot 4 = 0{,}8$. Большой вес в TF-IDF получат слова с высокой частотой в пределах конкретного документа и с низкой частотой употреблений в других документах.

Таким образом, полученный вектор признаков $\langle f_1, ..., f_7, ..., f_m \rangle$ является признаковым описанием URL-адреса $u \in U$. Совокупность признаковых описаний всех объектов выборки U^l, записанная в виде таблицы размера $l \times m$, является матрицей объектов-признаков:

$$F = \|f_j(u_i)\|_{l \times m} = \begin{pmatrix} f_1(u_1) & \ldots & f_m(u_1) \\ \ldots & \ldots & \ldots \\ f_1(u_l) & \ldots & f_m(u_l) \end{pmatrix}. \qquad (4)$$

Для множества сайтов значения m различны, и это накладывает определенные ограничения на выбор моделей для обучения.

3. Описание подхода выявления вредоносных URL-адресов на основе усредненного ансамбля моделей

Процесс выявления вредоносных URL-адресов следует разделить на следующие этапы:

Этап 1. Сбор данных: эта фаза направлена на сбор всей доступной информации о URL-адресе, такой как наличие URL-адресов в черном списке; явные признаки URL-адреса, такие как строка URL и информация о хосте; контент веб-сайта, такой как HTML и JavaScript; информация о доступности и т. д.

Этап 2. Предварительная обработка данных: на этом этапе неструктурированная информация о URL-адресе (например, текстовое описание) должна быть преобразована в матрицу объектов-признаков.

Этап 3. Выбор и обучение моделей.

В современной литературе упоминается большое число отдельных алгоритмов машин-

№ п/п	Название модели	Организация	Страна разработки	Год разработки	Значение СКО
1	Gradient Boosting	INRIA	Франция	2010	0,2730239
2	XGBoost	Вашингтонский университет	США	2014	0,2693472
3	LightGBM	Microsoft	США	2016	0,2756339
4	Catboost	Яндекс	РФ	2017	0,2675411
5	**Усредненный ансамбль**	**ВКА имени А. Ф. Можайского**	**РФ**	**2019**	**0,2674244**

ного обучения, которые могут быть применены для выявления вредоносных URL-адресов [21—26].

Однако проблема применения матрицы объектов-признаков, отображающей состояние URL-адреса, заключается в том, что число признаков может быть не фиксировано или не известно заранее. Например, можно рассматривать каждое слово как отдельный признак для текстового обозначения домена и всех поддоменов URL-адреса, который мог встречаться в данных для обучения. Модель выявления вредоносных URL-адресов может быть обучена на этих данных, но при работе с новыми URL-адресами, которые могут содержать слова, которых не было в обучающих данных, точность выявления будет снижаться.

Моделью выявления вредоносных URL-адресов называется параметрическое семейство отображений $A = \{g(u, \theta)|\theta \in \Theta\}$, где g: $U \times \Theta \to Y$ — некоторая фиксированная функция, Θ — множество допустимых значений параметра θ. Для задачи с m признаками f_j: $U \to \mathbb{R}$, $j = 1, ..., m$, используются модели с вектором параметров $\theta = (\theta_1, ..., \theta_m) \in \Theta = \mathbb{R}^m$:

$$g(u, \theta) = \text{sign} \sum_{j=1}^{m} \theta_j f_j(u). \quad (5)$$

Так, для повышения точности выявления вредоносных URL-адресов целесообразно объединить несколько алгоритмов в ансамбль моделей, чтобы повысить точность классификатора и уменьшить дисперсию его работы [27].

Усреднение ансамбля моделей — это отображение μ: $(U \times Y)^l \to A$, которое произвольной конечной выборке $U^l = (u_i, y_i)_{i=1}^l$ ставит в соответствие алгоритм:

$$g_{\text{ср}} = \frac{1}{N} \sum_{k=1}^{N} g_k, \quad g_k \in A. \quad (6)$$

Создание усредненного ансамбля моделей заключается в следующем:

1. Разработка N моделей, каждая со своими значениями точности выявления.

2. Обучение каждой модели отдельно.

3. Объединение моделей и усреднение значений в конечном классификаторе.

Преимущество такого подхода заключается в том, что ансамбль моделей может быть легче обучен на небольших входных наборах данных [28] и часто повышает точность по любому отдельному алгоритму [29].

Чтобы подход имел эффективную программную реализацию, выбор моделей основывался на успешном применении моделей на основе градиентного бустинга (LightGBM [32], Gradient Boosting [33], XGBoost [34] и CatBoost [35]) в соревнованиях по машинному обучению [31].

Для измерения эффективности отдельных моделей и разработанного метода будем использовать метрику среднеквадратичной ошибки, которая рассчитывается как

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - y_i^*)^2}, \quad (7)$$

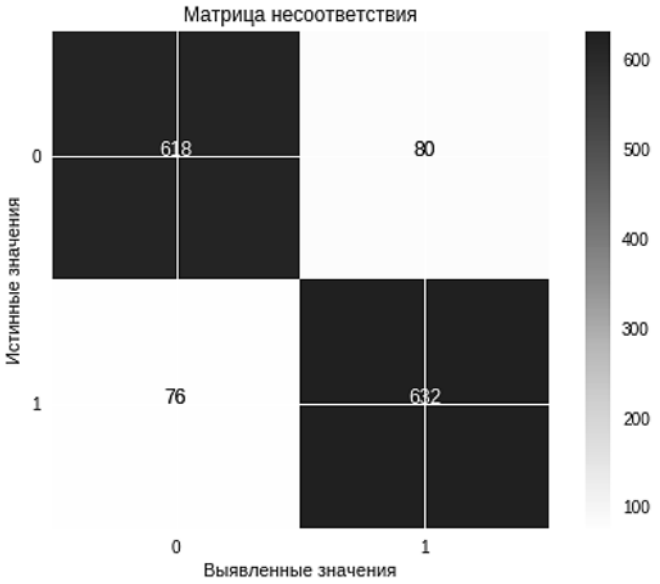
где y_i — результат выявления состояния URL-адреса, а y_i^* — истинное значение. Результаты измерений представлены в таблице.

Этап 4. Определение состояния URL-адреса.

4. Оценивание эффективности выявления вредоносных URL-адресов

Для определения качества разработанной методики будем использовать набор данных [5], который содержит 7030 записей URL-адресов и их значений, включающих 3494 вредоносных и 3536 безопасных URL-адресов. Валидационную часть определим равной 0,2 от общего набора данных, что включает 696 вредоносных и 712 безопасных URL-адресов.

Для оценивания качества выходных данных применим матрицу несоответствия выявления вредоносных URL-адресов, представленную на рисунке.



Матрица несоответствия выявления вредоносных URL-адресов

Матрица несоответствия отображает число верных и ошибочных выявлений по сравнению с фактическими данными. Рассмотрим полученные результаты:

- (0, 0) — правильно выявлены состояния не вредоносных URL-адресов;
- (1, 1) — правильно выявлены состояния вредоносных URL-адресов;
- (0, 1) — URL-адреса безопасны, а принято решение о том, что они вредоносны;
- (1, 0) — URL-адреса вредоносны, а принято решение о том, что они безопасны.

Эти вероятности первого и второго рода можно вычислить как вероятность попадания случайной величины y_i в область допустимых значений классов состояний URL-адресов, т. е. $p_1 = P(0, 1)$ и $p_2 = P(1, 0)$. Подставив эти значения из матрицы несоответствия в формулу (2), получим

$$П = M[C] = \frac{80}{696} \cdot 80 + \frac{76}{712} \cdot 76 \approx 17,3.$$

Сравним средний риск рандомного определения состояний (т. е. с вероятностью 0,5) URL-адресов:

$$П_{\text{ранд}} = M_{\text{ранд}}[C_{\text{ранд}}] = \\ = \frac{348}{696} \cdot 348 + \frac{356}{712} \cdot 356 \approx 350.$$

Таким образом, разработанный ансамбль моделей показал повышение качества выявления вредоносных URL-адресов.

Заключение

Разработка новых подходов к повышению защищенности компаний и пользователей информационных web-систем является постоянной и актуальной задачей.

Элементом новизны разработанного подхода является предложение по применению признаков, определяющих лексическую природу вредоносного URL-адреса. Особенностью данного подхода является учет возрастающей схожести составных частей URL-адресов у злоумышленников, что дополняет существующие подходы по явному выявлению вредоносных сайтов и способствует более точному выявлению вновь созданных в целях их дальнейшего блокирования.

Практическая значимость заключается в возможности применения подхода при обосновании и разработке технических решений информационной безопасности.

Разработанный подход к выявлению вредоносных URL-адресов на основе лексической обработки URL-адресов и усредненного ансамбля моделей имеет преимущество перед современными алгоритмами машинного обучения при выявлении информационных признаков и позволяет повысить качество выявления вредоносных URL-адресов.

Но несмотря на многообещающую способность подходов машинного обучения одним из основных недостатков разработанного подхода выявления вредоносных URL-адресов может быть его ресурсоемкий характер (особенно при извлечении признаков), что снижает их практическую ценность по сравнению с методами внесения в черный список.

Таким образом, дальнейшим развитием исследований может быть:

- исследование вопросов сбора дополнительных данных об URL-адресах;
- исследование вопросов влияния дисбаланса данных для обучения моделей;
- исследование вопросов повышения производительности выявления вредоносных URL-адресов;
- разработка технических решений безопасности почтовых сервисов разных типов.

Список литературы

1. **Sahoo D., Liu C., Steven C.** Malicious URL Detection using Machine Learning // A Survey. 2017. URL: https://arxiv.org/pdf/1701.07179.pdf

2. **Hong J.** The state of phishing attacks // Communications of the ACM. 2012. Vol. 55, N. 1. P. 74—81.

3. **Berners-Lee T.** Uniform Resource Locators (URL): A Syntax for the Expression of Access Information of Objects on the Network // World Wide Web Consortium. 2015.

4. **Liang B., Huang J., Liu F., Wang D., Dong D., Liang Z.** Malicious web pages detection based on abnormal visibility recognition // E-Business and Information System Security: International Conference on. IEEE. 2009. P. 1—5.

5. **Данные** вредоносных сайтов компании Phishtank. URL: https://www.phishtank.com/ (дата обращения 26.01.2019).

6. **Sinha S., Bailey M., Jahanian F.** Shades of grey: On the effectiveness of reputation-based "blacklists" in Malicious and Unwanted Software // MALWARE 2008. 3rd International Conference on. IEEE. 2008. P. 57—64.

7. **Garera S., Provos N., Chew M., Rubin A.** A framework for detection and measurement of phishing attacks // ACM workshop on Recurring malcode. ACM, 2007. P. 1—8.

8. **Alshboul Y., Nepali R., Wang Y.** Detecting malicious short urls on twitter // Twenty-first Americas Conference on Information Systems, Puerto Rico, 2015.

9. **Patil D. R., Patil J.** Survey on malicious web pages detection techniques // Science and Technology. 2015. Vol. 8, N. 5. P. 195—206.

10. **McGrath D. K., Gupta M.** Behind phishing: An examination of phisher modi operandi // LEET. 2008. Vol. 8. P. 4.

11. **Ma J., Saul L. K., Savage S., Voelker G. M.** Beyond blacklists: learning to detect malicious web sites from suspicious urls // The 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009. P. 1245—1254.

12. **Learning** to detect malicious urls // ACM Transactions on Intelligent Systems and Technology (TIST). 2011. Vol. 2, N. 3. P. 30.

13. **Purkait S.** Phishing counter measures and their effectiveness—literature review // Information Management & Computer Security. 2012. Vol. 20, N. 5. P. 382—420.

14. **Khonji M., Iraqi Y., Jones A.** Phishing detection: a literature survey // IEEE Communications Surveys & Tutorials. 2013. Vol. 15, N. 4. P. 2091—2121.

15. **Nepali R. K., Wang Y.** You look suspicious: Leveraging visible attributes to classify malicious short urls on twitter // The 49th Hawaii International Conference on System Sciences (HICSS). IEEE, 2016. P. 2648—2655.

16. **Kuyama M., Kakizaki Y., Sasaki R.** Method for detecting a malicious domain by using whois and dns features // The Third International Conference on Digital Security and Forensics (DigitalSec2016). 2016. P. 74.

17. **Kolari P., Finin T., Joshi A.** Svms for the blogosphere: Blog identification and splog detection, // AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs. 2006. P. 92—99.

18. **Ma J., Saul L. K., Savage S., Voelker G. M.** Identifying suspicious urls: an application of large-scale online learning // The 26th Annual International Conference on Machine Learning. ACM, 2009. P. 681—688.

19. **Yadav S., Reddy A. K. K., Reddy A., Ranjan S.** Detecting algorithmically generated malicious domain names // The 10th ACM SIGCOMM conference on Internet measurement. ACM, 2010. P. 48—61.

20. **Le A., Markopoulou A., Faloutsos M.** Phishdef: Url names say it all // INFOCOM, 2011. IEEE, 2011. P. 191—195.

21. **Ramanathan V., Wechsler H.** Phishing website detection using latent dirichlet allocation and adaboost // Intelligence and Security Informatics (ISI), 2012. IEEE International Conference on. IEEE, 2012. P. 102—107.

22. **Astorino A., Chiarello A., Gaudioso M.** Piccolo Malicious url detection via spherical classification // Neural Computing and Applications. 2016. P. 1—7.

23. **Hu Z., Chiong R., Pranata I., Susilo W., Bao Y.** Identifying malicious web domains using machine learning techniques with online credibility and performance data // Evolutionary Computation (CEC). IEEE, 2016. P. 5186—5194.

24. **Dekel O., Shalev-Shwartz S., Singer Y.** The forgetron: A kernelbased perceptron on a budget // SIAM Journal on Computing. 2008. Vol. 37, N. 5. P. 1342—1372.

25. **Lu J., Hoi S. C., Wang J., Zhao P., Liu Z.-Y.** Large scale online kernel learning // JMLR. 2016.

26. **Hoi S. C., Jin R., Zhao P., Yang T.** Online multiple kernel classification // Machine Learning. 2013. Vol. 90, N. 2. P. 289—316.

27. **Документация** разработчика модуля SciKit Python. URL: https://scikit-learn.org/stable/modules/ensemble.html (дата обращения: 5.02.2019).

28. **Haykin S.** Neural networks: a comprehensive foundation. N. J.: Prentice Hall, 1999.

29. **Hashem S.** Optimal linear combinations of neural networks // Neural Networks. 1997. Vol. 10, N. 4. P. 599—614.

30. **Описание** TF-IDF. URL: https://ru.wikipedia.org/wiki/TF-IDF (дата обращения: 5.02.2019).

31. **Свалин А.** CatBoost против Light GBM против XGBoost. URL: https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db (дата обращения: 6.02.2019).

32. **Документация** разработчика LGBM. URL: https://lightgbm.readthedocs.io (дата обращения: 7.02.2019).

33. **Документация** разработчика Gradient Boosting. URL: (дата обращения: 7.02.2019).

34. **Документация** разработчика XGBoost. URL: https://xgboost.readthedocs.io (дата обращения: 7.02.2019).

35. **Документация** разработчика CatBoost. URL: https://tech.yandex.ru/catboost/ (дата обращения: 7.02.2019).

36. **Набор** данных URL-адресов. URL: http://www.squidguard.org/blacklists.html (дата обращения: 12.02.2019).

37. **Набор** данных URL-адресов. URL: https://www.kaggle.com/teseract/datasets (дата обращения: 12.02.2019).

A. B. Menisov, Ph. D., Scientist of Military Researching Department, e-mail: men.arty@yandex.ru,
I. A. Shastun, Ph. D., Lecturer, e-mail: shastunivan1982@gmail.com,
Space military academy named by A. F. Mozhaysky, Saint-Petersburg,
S. U. Kapitsin, Ph. D., e-mail: wolf76@inbox.ru, Military Academy of the General Staff, Moscow

The Approach of Detecting Malicious Internet Sites Based on the Processing of Lexical Attributes of Addresses (URLs) and Averaged Ensemble of Models

Currently, the detection and blocking of access to malicious Internet sites is performed mainly by including URLs in blacklists. However, blacklists cannot be exhaustive, and they cannot be used to identify newly created malicious sites. The purpose of the article is to develop a new approach to detect malicious sites based on the average ensemble of models, allowing to improve the accuracy of detection. The developed approach, compared to modern technical solutions in the field of information security, takes into account the lexical features of the address bar (URL) of malicious sites that attackers try to modify to the address of a well-known or secure site. Also, the authors compare the results of identifying malicious URLs obtained by modern and developed approaches. The article will be useful not only to researchers in the field of machine learning, but also to experts in the cybersecurity industry.

Keywords: *information security, malicious websites, machine learning*

DOI: 10.17587/it.25.691-697

References

1. **Sahoo D., Liu C., Steven C.** Malicious URL Detection using Machine Learning, *A Survey*, 2017, available at: https://arxiv.org/pdf/1701.07179.pdf

2. **Hong J.** *Communications of the ACM*, 2012, vol. 55, no. 1, pp. 74—81.

3. **Berners-Lee T.** Uniform Resource Locators (URL): A Syntax for the Expression of Access Information of Objects on the Network, *World Wide Web Consortium*, 2015.

4. **Liang B., Huang J., Liu F., Wang D., Dong D., Liang Z.** *E-Business and Information System Security: International Conference on. IEEE*, 2009, pp. 1—5 (in Rssian).

5. **Malicious** URL dataset of Phishtank, available at: https://www.phishtank.com/ (date: 26.01.2019).

6. **Sinha S., Bailey M., Jahanian F.** *MALWARE 2008. 3rd International Conference on. IEEE*, 2008, pp. 57—64.

7. **Garera S., Provos N., Chew M., Rubin A.** *ACM workshop on Recurring malcode. ACM*, 2007, pp. 1—8.

8. **Alshboul Y., Nepali R., Wang Y.** Detecting malicious short urls on twitter, *Twenty-first Americas Conference on Information Systems, Puerto Rico,* 2015.

9. **Patil D. R., Patil J.** *Science and Technology*, 2015, vol. 8, no. 5, pp. 195—206.

10. **McGrath D. K., Gupta M.** *LEET*, 2008, vol. 8, pp. 4.

11. **Ma J., Saul L. K., Savage S., Voelker G. M.** *The 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM*, 2009, pp. 1245—1254.

12. **Learning** to detect malicious urls, *ACM Transactions on Intelligent Systems and Technology (TIST),* 2011, vol. 2, no. 3, pp. 30.

13. **Purkait S.** *Information Management & Computer Security*, 2012, vol. 20, no. 5, pp. 382—420.

14. **Khonji M., Iraqi Y., Jones A.** *IEEE Communications Surveys & Tutorials*, 2013, vol. 15, no. 4, pp. 2091—2121.

15. **Nepali R. K., Wang Y.** The 49th Hawaii International Conference on System Sciences (HICSS), IEEE, 2016, pp. 2648—2655.

16. **Kuyama M., Kakizaki Y., Sasaki R.** The Third International Conference on Digital Security and Forensics (DigitalSec 2016), 2016, pp. 74.

17. **Kolari P., Finin T., Joshi A.** AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs, 2006, pp. 92—99.

18. **Ma J., Saul L. K., Savage S., Voelker G. M.** *The 26th Annual International Conference on Machine Learning. ACM,* 2009, pp. 681—688.

19. **Yadav S., Reddy A. K. K., Reddy A., Ranjan S.** *The 10th ACM SIGCOMM conference on Internet measurement. ACM*, 2010, pp. 48—61.

20. **Le A., Markopoulou A., Faloutsos M.** *INFOCOM*, 2011, *IEEE*, 2011, pp. 191—195.

21. **Ramanathan V., Wechsler H.** *Intelligence and Security Informatics (ISI), 2012. IEEE International Conference on. IEEE*, 2012, pp. 102—107.

22. **Astorino A., Chiarello A., Gaudioso M.** *Neural Computing and Applications*, 2016, pp. 1—7.

23. **Hu Z., Chiong R., Pranata I., Susilo W., Bao Y.** *Evolutionary Computation (CEC), IEEE*, 2016, pp. 5186—5194.

24. **Dekel O., Shalev-Shwartz S., Singer Y.** *SIAM Journal on Computing*, 2008, vol. 37, no. 5, pp. 1342—1372.

25. **Lu J., Hoi S. C., Wang J., Zhao P., Liu Z.-Y.** Large scale online kernel learning, *JMLR*, 2016.

26. **Hoi S. C., Jin R., Zhao P., Yang T.** *Machine Learning*, 2013, vol. 90, no. 2, pp. 289—316.

27. **SciKit** Development documentation, available at: https://scikit-learn.org/stable/modules/ensemble.html (date: 5.02.2019).

28. **Haykin S.** Neural networks: a comprehensive foundation, N. J., Prentice Hall, 1999.

29. **Hashem S.** *Neural Networks*, 1997, vol. 10, no. 4, pp. 599—614.

30. **Definition** of TF-IDF, available at: https://wikipedia.org/wiki/TF-IDF (date: 5.02.2019) (in Rssian).

31. **Svalin A.** CatBoost vs Light GBM vs XGBoost, available at: https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db (date: 6.02.2019) (in Russian).

32. **LGBM** Development documentation, available at: https://lightgbm.readthedocs.io (date: 7.02.2019) (in Rssian).

33. **Gradient** Boosting Development documentation, available at: (date: 7.02.2019) (in Rssian).

34. **XGBoost** Development documentation, available at: https://xgboost.readthedocs.io (date: 7.02.2019) (in Rssian).

35. **CatBoost** Development documentation, available at: https://tech.yandex.ru/catboost/ (date: 7.02.2019) (in Rssian).

36. **URL dataset**, available at: http://www.squidguard.org/blacklists.html (date: 12.02.2019) (in Rssian).

37. **URL dataset**, available at: https://www.kaggle.com/teseract/datasets (date: 12.02.2019) (in Rssian).