УДК 004.8
DOI: 10.26102/2310-6018/2022.38.3.025

Алгоритм формирования базы уязвимостей и выбор архитектуры нейронной сети для их обработки

Е.Ю. Соболевская✉, И.Д. Шевченко, С.Е. Алексеев

Владивостокский государственный университет,
Владивосток, Российская Федерация
study_z@list.ru✉

Резюме. В статье рассматривается необходимость алгоритма для формирования базы уязвимости информационной системы и выбор архитектуры нейронной сети. Приведено описание существующих систем и критериев оценки уязвимостей, группы метрик. Проанализированы базы уязвимостей и выявлены расхождения в оценке уязвимостей, преимущества и недостатки. В работе выделены и исследованы следующие архитектуры: feed forward neural network, generative adversarial network, autoencoder, recurrent neural network without long short-term memory, recurrent neural network with long short-term memory, многослойный перцептрон Румельхарта, liquid state machine, Boltzmann machine. Представлен предварительный анализ архитектур нейронных сетей с учетом значимых параметров для дальнейшего использования в сфере информационной безопасности и классификации уязвимостей. Исходя из полученных результатов, в ходе исследования параметров нейронных сетей выделены feed forward neural network, recurrent neural network with long short-term memory и generative adversarial network. Предложен альтернативный способ формирования базы уязвимостей с использованием нейронных сетей. В результате предложен алгоритм формирования базы уязвимостей и предложен способ его автоматизации при помощи нейронной сети. Решение позволит нейронной сети постоянно получать актуальные данные для обучения, вследствие чего база уязвимостей будет пополняться с максимально возможной скоростью, что сделает ее наиболее полной, достоверной и актуальной из всех существующих баз уязвимостей.

Ключевые слова: уязвимости, нейронные сети, архитектура нейронной сети, алгоритм, угроза.

Для цитирования: Соболевская Е.Ю., Шевченко И.Д., Алексеев С.Е. Алгоритм формирования базы уязвимостей и выбор архитектуры нейронной сети для их обработки. *Моделирование, оптимизация и информационные технологии*. 2022;10(3). Доступно по: https://moitvivt.ru/ru/journal/pdf?id=1226 DOI: 10.26102/2310-6018/2022.38.3.025

Vulnerability base formation algorithm and neural network architecture selection for its processing

E.Y. Sobolevskaya✉, I.D. Shevchenko, S.E. Alekseev

Vladivostok State University, Vladivostok, Russian Federation
study_z@list.ru✉

Abstract. The article discusses the need for an algorithm to form the information system vulnerability base and the selection of the neural network architecture. A description of existing systems and criteria for assessing vulnerabilities as well as a group of metrics are given. The vulnerability databases were analyzed and discrepancies in the assessment of vulnerabilities, advantages and disadvantages were identified. The following architectures were identified and studied: feed forward neural network, generative adversarial network, Autoencoder, recurrent neural network without long short-term memory, recurrent neural network with long short-term memory, Rumelhart multilayer perceptron, liquid state machine, Boltzmann machine. A preliminary analysis of neural network architectures is presented taking into account significant parameters for further use in the field of information security

and vulnerability classification. Based on the results obtained during the study of the parameters of neural networks, feed forward neural network, recurrent neural network with long short-term memory and generative adversarial network were identified. An alternative method of forming a vulnerability database by means of neural networks is proposed. As a result, an algorithm for forming a vulnerability base and a method for automating it using a neural network are suggested. The solution will allow the neural network to constantly receive up-to-date data for training and, owing to this, the vulnerability database will be updated as quickly as possible, which will make it the most complete, reliable and up-to-date of all existing vulnerability databases.

Keywords: vulnerabilities, neural networks, neural network architecture, algorithm, threat.

For citation: Sobolevskaya E.Y., Shevchenko I.D., Alekseev S.E. Vulnerability base formation algorithm and neural network architecture selection for its processing. *Modeling, Optimization and Information Technology*. 2022;10(3). Available from: https://moitvivt.ru/ru/journal/pdf?id=1226 DOI: 10.26102/2310-6018/2022.38.3.025 (In Russ.).

Введение

Информационные технологии являются передовым направлением, которое динамично развивается и останется актуальным долгие годы. Искусственный интеллект и информационная безопасность – одни из самых востребованных и развивающихся направлений, которые окружают цифровой мир и используются во многих компаниях.

Зачастую программное обеспечение является комплексным и обеспечение безопасности не является тривиальной задачей. Необходимо учитывать все потенциальные уязвимости для обеспечения безопасности программного обеспечения и информационной системы в целом.

Реалии таковы, что современные информационные системы подвержены угрозам, которые связаны с уязвимостями программного и/или аппаратного обеспечения [1]. Кроме того, безопасность программного обеспечения связана с людьми, которые разрабатывают и используют приложения [2], а оценка актуальных угроз и формирование базы уязвимостей с применением нейронных сетей позволит учитывать множество факторов для обеспечения безопасности информационной системы в целом.

Цель исследования

Целью данного исследования является формирование алгоритма базы уязвимости информационной системы и выбор архитектуры нейронной сети с учетом особенностей, измерения параметров, их классификации и дальнейшее использование в сфере информационной безопасности. Алгоритм базы уязвимости информационной системы – это основа для анализа существующих баз уязвимостей, с целью дальнейшего формирования актуальной и достоверной базы данных и ее оперативное обновление и развитие критериев оценки уязвимостей. В дальнейшем исследование послужит для создания программного обеспечения, которое сможет находить, классифицировать и предпринимать попытку обезвредить уязвимость. Особенностью программного обеспечения является использование нейронной сети для самостоятельного обучения и как следствие повышения безопасности информационной системы.

Материалы и методы

Об актуальности проблемы и необходимости создания подобной базы говорит в первую очередь то, что ежедневно появляются уязвимости, на оценку которых уходит слишком много времени. Из-за этого процесс их устранения может сильно затянуться, так как этим занято недостаточное количество специалистов. Примером такой уязвимости является уязвимость одной из библиотек языка Java — Log4Shell, из-за нее

большое количество компаний понесли убытки, а данные огромного числа пользователей украдены злоумышленниками. Если бы существовала нейронная сеть, способная выносить оценку опасности уязвимостей за секунду, то этого ущерба можно избежать либо минимизировать.

В информационной среде множество ресурсов, содержащих примеры баз уязвимостей, например: Common Vulnerabilities and Exposures (CVE), Common Weakness Enumeration (CWE), National Vulnerability Database (NVD), Common Vulnerability Scoring System (CVSS), Common Attack Pattern Enumeration and Classification (CAPEC), MITRE ATT&CK Matrix, Common Platform Enumeration (CPE), банк данных угроз безопасности информации (БДУ) федеральной службы по техническому и экспортному контролю (ФСТЭК) России [3].

Согласно терминологии CWE и CAPEC, понятия слабость и уязвимость различны [4]. Программное обеспечение или информационная система, имеющие ошибки, например, в кодировании, архитектуре являются слабостью, а уязвимость – это слабое место, которое может быть использовано для дистанционных действий. В соответствии с ГОСТ Р 56545, уязвимость – это недостаток программного средства или информационной системы в целом, которая может быть использована, для реализации угроз безопасности информации [5] именно данное понятие уязвимости используется в настоящей работе.

Для понимания и обоснования необходимости формирования собственной базы уязвимости рассмотрены БДУ ФСТЭК России [6] и NVD [7]. Любая уязвимость информационной системы может быть представлена в виде набора, который включает в себя определенные характеристики, которые формируются по определенным правилам [8].

Для оценки критичности уязвимостей применяется стандарт CVSS. Базовая группа метрик CVSS включает в себя шесть метрик. Первые три – способ получения доступа, сложность получения доступа, показатель аутентификации [9]. На данный момент существуют два способа оценки. Первый способ – оценка по системе CVSS 2.0, а второй по системе CVSS 3.X. В CVSS 2.0 осталась проблема недостаточной информативности оценки, в стандарте также появлялись уязвимости с одинаковыми векторами, но с разной величиной опасности. В CVSS 3.0 предлагается учитывать воздействие в каждом показателе и дополнительно повышать оценку опасности, если область действия атаки изменена. Для этого добавлен показатель смены границы эксплуатации [9]. В системе оценки CVSS 2.0 существует проблема недостаточной информативности оценки, в данном стандарте также появлялись уязвимости с одинаковыми векторами, но при этом опасность их оценивалась по-разному. В CVSS 3.0 предлагается учитывать максимальное воздействие в каждой из метрик, а также дополнительно повышать оценку опасности, если область действия атаки изменена. Для этого добавлен новый критерий смены границы эксплуатации [9].

Таким образом, предложили для дальнейшей реализации базы данных уязвимостей использовать общепринятую CVSS 3.X, так как характеристика должна исходить из наносимого ущерба, а не простоты использования.

Результаты

Проанализировав БДУ ФСТЭК, выявлены следующие недостатки, во-первых, оценка опасности уязвимостей с 2015 года проходит как по системе CVSS 2.0, так и по системе CVSS 3.0. Самым большим недостатком базы является то, что ее обновление происходит недостаточно быстро. NVD, о которой речь пойдет ниже, обновляется несколько раз в день, а БДУ ФСТЭК обновляется раз в неделю, а иногда и реже.

Во-вторых, не может являться международной базой в связи с тем, что как сайт, так и описания уязвимостей выполнены на русском языке.

В-третьих, банк данных угроз является «пустым» по сравнению с другими базами. Это доказывает то, что с мая по июнь 2022 года в NVD появилось более 8000 новых уязвимостей, что является пятой частью всех уязвимостей БДУ, которые собраны их командой с 2008 года.

NVD – база уязвимостей правительства США типа CVE имеет ряд преимуществ. Во-первых, NVD является англоязычным ресурсом, информация по старым и новым уязвимостям появляется в кратчайшие сроки.

Во-вторых, метрики оценки уязвимостей соответствуют новейшим международным стандартам. О достоверности последнего суждения говорит результат анализа большого числа иностранных ресурсов, составляющих базы уязвимостей, большая часть из которых ссылается на NVD. Другие иностранные базы представляют собой более старые версии NVD, либо имеют устаревшие критерии оценивания.

Составлена сводная таблица баз уязвимостей NVD и БДУ ФСТЭК, фрагмент представлен в Таблице1.

Таблица 1 – Фрагмент сравнения баз уязвимостей NVD и ФСТЭК
Table 1 – A comparison fragment of NVD and FSTEC vulnerability databases

CVE	NVD		БДУ ФСТЭК	
	CVSS2.0	CVSS3.0	CVSS2.0	CVSS3.0
CVE-2018-8271	2,1	5,5	3,8	2,5
CVE-2016-7218	1,9	4,7	1,9	2,9
CVE-2016-7217	9,3	8,8	9,3	8,1
CVE-2016-7214	2,1	3,3	4,9	5,5
CVE-2016-7215	7,2	7,8	7,2	7,8
CVE-2017-11771	10	9,8	10	9,8
CVE-2018-8475	6,8	8,8	10	8,8
CVE-2018-8553	9,3	7,8	10	9,6
CVE-2018-8453	7,2	7,8	6,2	7
CVE-2019-0865	5	7,5	7,8	7,5

Проанализировав таблицу, очевидны расхождения в оценках, одних и тех же уязвимостей в разных базах. Разницу видно как при оценке по стандарту CVSS 2.0, так и при оценке по CVSS 3.X. Причиной этого, скорее всего, является то, что в разных базах различные метрики имеют разный уровень опасности. В связи с этим могут возникнуть следующие проблемы: невозможность объективно оценивать критичность уязвимостей, утечки данных, в связи с несвоевременным устранением уязвимостей и проникновением злоумышленников в информационную систему, в связи с тем, что уязвимость неверно оценена, в связи с устаревшими метриками в одной из баз уязвимостей.

Для оценки уязвимостей по стандарту CVSS можно использовать специальные программы. Одной из таких является калькулятор CVSS от NVD, с его помощью можно быстро оценить найденную уязвимость. Главной проблемой данного калькулятора является то, что с ним работают люди, что повышает шанс ошибки при оценке. Вследствие подобных ошибок и возникает разница в оценке уязвимостей в различных базах уязвимостей.

Во избежание данных проблем и минимизации ошибок целесообразно использовать самообучающиеся системы и составление универсальной базы

уязвимостей, в которой оперативно добавляются новые метрики и переоцениваются старые в соответствии с современными знаниями о различных уязвимостях.

Для того, чтобы составить такую базу, нужен алгоритм, который сможет непрерывно обучаться и иметь минимальный процент ошибки. Учитывая объем работы и ее сложность, крайне рационально использовать для этого технологии машинного обучения и нейронные сети, в частности. Алгоритм формирования базы уязвимости представлен на Рисунке 1.

Рисунок 1 – Алгоритм формирования базы уязвимости
Figure 1 – Algorithm for creating a database of system vulnerabilities

В соответствии с предложенным алгоритмом, первый шаг – создание критериев оценки, то есть формирование критериев оценки с применением анализа современных стандартов и критериев оценки уязвимостей. В результате данного шага сформированы актуальные и достоверные критерии оценки уязвимостей.

Второй шаг – сбор уязвимостей. Под сбором уязвимостей подразумевается сбор данных из уже сформированных баз уязвимостей, в том числе зарубежных. Цель данного шага – это актуализация баз уязвимостей и приведение к единому виду.

Третий шаг – формирование базы уязвимостей в соответствии с проанализированными критериями.

Заключительный шаг алгоритма – анализ полученных данных. Анализ данных и улучшение существующих критериев оценки, а также анализ уязвимостей, которые не могут быть правильно классифицированы в соответствии с текущими критериями оценки.

Исходя из алгоритма очевидна необходимость использования нейронной сети, которая сможет самостоятельно улучшать критерии оценки и классифицировать новые уязвимости с минимальным вмешательством человека. Предложенный алгоритм – часть системы сбора, анализа и обработки уязвимостей.

Для дальнейшей реализации представленного алгоритма формирования баз уязвимостей выделены и исследованы следующие архитектуры: feed forward neural network (FFNN) [10], generative adversarial network (GAN) [11], autoencoder, recurrent neural network (RNN without LSTM), recurrent neural network with LSTM (RNN with LSTM) [12], многослойный перцептрон Румельхарта (MLP) [13], liquid state machine (LSM) [14], Boltzmann machine (Машина Больцмана).

В данной работе использовался персональный компьютер со следующими характеристиками: процессор Intel Core i7-7800k, 6 ядер с частотой 3200 МГц (4600 МГц

Turbo), видеокарта NVIDIA GeForce RTX 2070, 1410-1620 Мгц, 10800млн транзисторов, объём памяти 8192мб, шина памяти 256bit GDDR6, частота работы памяти 14000МГц, техпроцесс 12нм, модули памяти Kingston Fury Beast Black, DDR4 – 2x8ГБ (итого 16ГБ) 3200МГц, двухканальный режим работы.

Каждая модель имеет следующие параметры: метрика Accuracy, функция потерь (mean squared error (MSE) и mean absolute error (MAE)), оптимизатор (adam и SGD), функция активации (linear и sigmoid). Параметры каждой архитектуры приведены в Таблице 2.

Таблица 2 – Параметры архитектур нейронных сетей
Table 2 – Neural network architecture parameters

Критерий	FFNN	GAN	Autoencoder	RNN with LSTM	RNN without LSTM	MLP	LSM
Метрика	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy
Функция потерь	MSE	MAE	MAE	MSE	MSE	MSE	MSE
Оптимизатор	adam	adam	adam	SGD	SGD	adam	adam
Функция активации	linear	sigmoid	linear	sigmoid	sigmoid	linear	linear

В качестве обучающей выборки служит массив из 200 чисел. Задачей нейросети является определение принадлежности числа к одному из двух классов. Числа являются девятизначным бинарным кодом, сгенерированным при помощи скрипта. Фрагмент выборки: […101011101, 101011110, 101011111…], как можно видеть из примера, числа генерировались в порядке возрастания, что нежелательно в данном случае, поэтому перед подачей в нейронную сеть, выборка перемешивалась. Тестовая выборка состоит из 100 чисел. Скрипт генерирует тестовую и обучающую выборки одновременно, после чего они разделяются на две части.

Проведены замеры у каждой архитектуры нейронной сети следующих значимых критериев:

- Скорость работы – то насколько быстро нейронная сеть заканчивает работу над выборкой данных, данный критерий измеряется в секундах. Выбирается минимальное значение, для наиболее быстрой обработки и решения поставленной задачи.
- Скорость обучения – количество эпох, которое потребовалось для обучения нейронной сети, данный критерий измеряется в количестве эпох. Аналогично скорости работы, наилучшая скорость обучения ускоряет процесс тестирования нейронной сети.
- Средние потери – оценка того, насколько подходит данная нейронная сеть для данной задачи, данный критерий измеряется в количественной оценке. Слишком большое значение данного критерия говорит о том, что нейронная сеть не предназначена для данной задачи, или по какой-то причине не может правильно обучиться.
- Точность – аналогично средним потерям, наибольшая точность позволяет понять, что данная модель хорошо подходит для решаемой задачи.

Результаты исследования представлены в Таблице 3.

Таблица 3 – Результаты сравнения нейронных сетей по выделенным критериям
Table 3 – The results of comparing neural networks according to the selected criteria

Критерий	FFNN	GAN	Autoencoder	RNN with LSTM	RNN without LSTM	MLP	LSM
Скорость работы	30,8	35,8	45,8	53,7	39,3	29	24,4
Скорость обучения	27	15	77	46	37	33	21
Средние потери	1-0	35-10	150,7-100,3	10-5	71,5-69,5	50,4-35,8	75,3-60,4
Точность	95,6	73,4	40,2	90,8	53,5	62,5	52,5

Исследования проведены на одинаковом оборудовании, характеристики которого приведены выше, на одинаковом количестве эпох, равном 100. Благодаря простоте использования, выбран язык программирования Python, фреймворк TensorFlow с использованием Keras.

Генеративно-состязательная нейросеть, в данном случае модель, показала хорошую скорость обучения, но слишком высокие потери и скорость работы. Нейронная сеть показала не лучшие результаты, так как не подходит под задачу. Следует отметить GAN, как достаточную модель нейронной сети по генерации данных и их классификации, но в данном случае она не подходит. Нейросеть с LSTM имеет малые потери, но низкую скорость работы и обучения. Без LSTM нейросеть показала не лучшие результаты. Такие нейронные сети как LSM, MLP и autoencoder имеют слишком высокие показатели потерь. Это означает, что данные архитектуры не подходят для настоящей задачи. FFNN архитектура показала результаты выше среднего, хорошо подходит для задачи классификации.

Исходя из полученных в результате анализа архитектур данных, можно сделать вывод, что наиболее подходящей архитектурой является FFNN, которая расширена и использована для классификации уязвимостей различных систем. Также хорошо себя показала RNN with LSTM, следует провести ее детальный анализ и найти способы оптимизации, особенно по отношению к скорости работы и быстроте обучения.

Обсуждение

В статье «Методика оценки актуальных угроз и уязвимостей на основе технологий когнитивного моделирования и Text Mining» [3] рассматривается нейросетевая модель, которая будет выдавать оценку уязвимости, основываясь на текстовых данных. В ходе анализа данного способа оценки выявлены следующие проблемы: возможность работы только с русским и английским языком, скорость получения данных из интернета. Для решения первой проблемы можно создать и обучить модели для поиска информации на всех возможных языках. К сожалению, данный способ не может решить вторую проблему, а скорее усугубляет ее.

Модель нейронной сети, которая работает, основываясь на текстовых данных, вынуждает тратить слишком много времени на поиск необходимой информации. Также для ее работы необходим компьютер с бесконечно растущей мощностью, так как постоянно появляются новые источники информации, для анализа которых и нужна такая мощность.

Решить проблему, связанную со временем работы модели нейросети, которая будет искать данные об уязвимостях, можно, используя способ, предложенный в статье «Semantic model of attacks and vulnerabilities based on CAPEC and CWE dictionaries» [4]. Автор описывает процесс создания базы уязвимостей, основываясь на базах CWE и CAPEC. Это также решает проблему с поиском информации для нейросети, так как ищем ее в уже существующих базах слабостей и шаблонов атак, что не требует большой мощности от используемого компьютера.

Также для предварительного обучения модели можно использовать еще и данные из баз уязвимостей. Это позволит модели использовать более точные критерии оценки уязвимостей. Стоит отметить и то, что для формирования базы может пригодиться сканер агрессивного типа, который будет пытаться использовать различные слабости системы и программного обеспечения, установленного в ней. Комбинируя слабости и используя разнообразные шаблоны для атак, сканер сможет находить новые уязвимости, которые еще не занесены в другие базы. Нейронная сеть, в свою очередь, оценивает их и корректирует свои метрики, если это потребуется для более совершенной оценки уязвимости.

Заключение

В результате проведенного исследования составлен алгоритм формирования базы уязвимостей, а также представлен способ его автоматизации при помощи нейронной сети. Исходя из полученных результатов в ходе исследования параметров нейронных сетей, выделены следующие нейронные сети: FFNN, RNN with LSTM и GAN. При масштабировании нейронной сети параметры также изменятся, в дальнейшем требуется тестирование всех трех архитектур, несмотря на то что лучше всего показала себя FFNN.

Дальнейшее тестирование будет проходить на различных персональных компьютерах, вследствие чего можно получить более точные и актуальные данные. Так же планируется реализация сканера уязвимостей агрессивного типа, который самостоятельно находит уязвимости и передает полученные данные далее для анализа их нейронной сетью. Такое решение позволит нейронной сети постоянно получать актуальные данные для обучения, вследствие чего база пополняется с максимально возможной скоростью, что сделает ее наиболее полной и актуальной из всех существующих баз уязвимостей. Предполагается, что сканер и предложенный алгоритм формирования базы уязвимостей – это модули системы поддержки принятия решений.

СПИСОК ИСТОЧНИКОВ

1. Баскаков А.В., Федорко Е.Д., Остапенко А.Г. Исследование глобальных баз данных уязвимостей информационно-телекоммуникационных систем. *Информация и безопасность*.2006;9(2):152–154.
2. Wen Sh.F. Software Security in Open Source Development: A Systematic Literature Review. *Conference of Open Innovations Associatio*n. 2017;21:364–373.
3. Васильев В.И., Вульфин А.М., Кириллова А.Д., Кучкарова Н.В. Методика оценки актуальных угроз и уязвимостей на основе технологий когнитивного моделирования и Text Mining. *Системы управления, связи и безопасности*. 2021;(3):110–134. DOI: 10.24412/2410-9916-2021-3-110-134.
4. Brazhuk A. Semantic model of attacks and vulnerabilities based on CAPEC and CWE dictionaries. *International Journal of Open Information Technologies*. 2019;7(3):38–41.
5. ГОСТ Р 56545-2015 Защита информации. Уязвимости информационных систем. Правила описания уязвимостей. Стандартинформ. Москва; 2015.

6. Реестр уязвимостей БДУ ФСТЭК России. Доступно по: http://www.bdu.fstec.ru/ (дата обращения: 10.06.2022).
7. Национальная база уязвимостей. Доступно по: https://nvd.nist.gov/ (дата обращения:10.06.2022).
8. Амиргамзаев Г.Г., Алимагомедов М.Г. Уязвимости информационных систем. *Вопросы устойчивого развития общества*. 2021;(4):440–442. DOI: 10.34755/ИРОК.2021.99.93.091.
9. Общий обзор систем оценки уязвимостей (CVSS 2.0/3.0). Доступно по: https://safe-surf.ru/specialists/article/5211/596644/ (дата обращения: 06.01.2022).
10. Круглов В.В., В.В. Борисов. *Искусственные нейронные сети. Теория и практика.* 2-е изд. М.: Горячая линия-Телеком; 2002. 382 с.
11. Goodfellow Y., Benjio I., Courville A. *Deep learning.* Cambridge: The MIT Press; 2016. 802 p.
12. Лаптев В.В., Данилов В.В., Гергет О.М. Исследование генеративно-состязательных сетей для синтеза новых медицинских данных. *Автоматизация и моделирование в проектировании и управлении*. 2020;2(8):17–23. DOI: 10.30987/2658-6436-2020-2-17-23.
13. Акинина Н.В., Акинин М.В., Соколова А.В. [и др.]. Автоэнкодер: подход к уменьшению размерности векторного пространства с контролируемой потерей информации. *Известия Тульского государственного университета. Технические науки*. 2016;(9):3–12.
14. Грушко Ю.В., Грушко В.В. Перцептивные интегральные паттерны – метод быстрого отслеживания зрачков на основе отклика персептрона ИНС-МЛП и модифицированный численный метод дифференциальной эволюции. *Международный исследовательский журнал.* 2022;6-1(120):36–55. DOI: 10.23670/IRJ.2022.120.6.001.

REFERENCES

1. Baskakov A.V., Fedorko E.D., Ostapenko A.G. Issledovanie global'nyh baz dannyh ujazvimostej informacionno-telekommunikacionnyh sistem. *Informacija i bezopasnost'*. 2006;9(2):152–154. (In Russ.).
2. Wen Sh.F. Software Security in Open Source Development: A Systematic Literature Review. *Conference of Open Innovations Association*. 2017;21:364–373.
3. Vasil'ev V.I., Vul'fin A.M., Kirillova A.D., Kuchkarova N.V. Metodika ocenki aktual'nyh ugroz i ujazvimostej na osnove tehnologij kognitivnogo modelirovanija i Text Mining. *Sistemy upravlenija, svjazi i bezopasnosti*. 2021;(3):110–134. DOI: 10.24412/2410-9916-2021-3-110-134. (In Russ.).
4. Brazhuk A. Semantic model of attacks and vulnerabilities based on CAPEC and CWE dictionaries. *International Journal of Open Information Technologies*. 2019;7(3):38–41.
5. GOST R 56545-2015 Zashhita informacii. Ujazvimosti informacionnyh sistem. Pravila opisanija ujazvimostej. Standartinform. Moskva; 2015. (In Russ.).
6. Reestr ujazvimostej BDU FSTJeK Rossii. Available at: http://www.bdu.fstec.ru/ (accessed on: 10.06.2022). (In Russ.).
7. Nacional'naja baza ujazvimostej. Available at: https://nvd.nist.gov/ (accessed on:10.06.2022). (In Russ.).
8. Amirgamzaev G.G., Alimagomedov M.G. Ujazvimosti informacionnyh sistem. *Voprosy ustojchivogo razvitija obshhestva.* 2021;(4):440–442. DOI: 10.34755/IROK.2021.99.93.091. (In Russ.).

9. Obshhij obzor sistem ocenki ujazvimostej (CVSS 2.0/3.0). Available at: https://safe-surf.ru/specialists/article/5211/596644/ (accessed on: 06.01.2022). (In Russ.).
10. Kruglov V.V., V.V. Borisov. *Iskusstvennye nejronnye seti. Teorija i praktika*. 2-e izd. M.: Gorjachaja linija-Telekom; 2002. 382 p. (In Russ.).
11. Goodfellow Y., Benjio I., Courville A. *Deep learning.* Cambridge: The MIT Press; 2016. 802 p.
12. Laptev V.V., Danilov V.V., Gerget O.M. Issledovanie generativno-sostjazatel'nyh setej dlja sinteza novyh medicinskih dannyh. *Avtomatizacija i modelirovanie v proektirovanii i upravlenii.* 2020;2(8):17–23. DOI: 10.30987/2658-6436-2020-2-17-23. (In Russ.).
13. Akinina N.V., Akinin M.V., Sokolova A.V. [i dr.]. Avtojenkoder: podhod k umen'sheniju razmernosti vektornogo prostranstva s kontroliruemoj poterej informacii. *Izvestija Tul'skogo gosudarstvennogo universiteta. Tehnicheskie nauki.* 2016;(9):3–12. (In Russ.).
14. Grushko Ju.V., Grushko V.V. Perceptivnye integral'nye patterny – metod bystrogo otslezhivanija zrachkov na osnove otklika perseptrona INS-MLP i modificirovannyj chislennyj metod differencial'noj jevoljucii. *Mezhdunarodnyj issledovatel'skij zhurnal*. 2022;6-1(120):36–55. DOI: 10.23670/IRJ.2022.120.6.001. (In Russ.).

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Соболевская Евгения Юрьевна, кандидат технических наук, доцент кафедры информационных технологий и систем, Владивостокский государственный университет, Владивосток, Российская Федерация.
e-mail: study_z@list.ru
ORCID: 0000-0001-9695-990X

Evgeniya Yurievna. Sobolevskaya, Candidate of Technical Sciences, Associate Professor at the Department of Information Technologies and Systems, Vladivostok State University, Vladivostok, Russian Federation.

Шевченко Иван Денисович, лаборант кафедры информационных технологий и систем Владивостокского государственного университета, Владивосток, Российская Федерация.
e-mail: iwshevchenko@yandex.ru

Ivan Denisovich Shevchenko, Laboratory Assistant at the Department of Information Technologies and Systems, Vladivostok State University, Vladivostok, Russian Federation.

Алексеев Сергей Евгеньевич, студент, Владивостокский государственный университет, Владивосток, Российская Федерация.
e-mail: aflexeevsrjjjj@gmail.com

Sergey Evgenievich Alekseev, Undergraduate Student, Vladivostok State University, Vladivostok, Russian Federation.

Статья поступила в редакцию 19.09.2022; одобрена после рецензирования 23.09.2022; принята к публикации 28.09.2022.

The article was submitted 19.09.2022; approved after reviewing 23.09.2022; accepted for publication 28.09.2022.