# ПРОГНОЗИРОВАНИЕ КАТЕГОРИЙ УЯЗВИМОСТЕЙ В КОНФИГУРАЦИЯХ УСТРОЙСТВ С ПОМОЩЬЮ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

*Левшун Д. С.[1], Веснин Д. В.[2], Котенко И. В.[3]*

**DOI:** 10.21681/2311-3456-2024-3-33-39

***Цель исследования:*** *исследование эффективности модификаций системы двунаправленного обучения трансформеров BERT при решении задачи прогнозирования категорий уязвимостей (CVE) для отдельных элементов (устройств) информационных систем на основе их конфигураций (CPE URIs).*

***Методы исследования:*** *методы обработки естественного языка, кросс-валидация моделей искусственного интеллекта, оптимизация гиперпараметров моделей искусственного интеллекта.*

***Полученные результаты:*** *на основе содержимого открытых баз уязвимостей собран набор данных, устанавливающий взаимосвязи между предобработанными CPE URI и выделенными 24 категориями CVE; исследована эффективность BERT, RoBERTa, XLM-RoBERTa и DeBERTaV3 при решении задачи прогнозирования категорий CVE на основе CPE URI на собранном наборе данных; получена модель BERT, оптимизированная для решения поставленной задачи; произведено сравнение полученного решения с аналогами.*

***Научная новизна:*** *данная работа является одной из первых в прогнозировании уязвимостей устройств на основе их конфигурации, что подчеркивает ее научную значимость и новизну. Более того, это также одна из первых работ, посвященная исследованию BERT для задачи прогнозирования уязвимостей.*

***Вклад:*** *Левшун Д. С., Котенко И. В. – выбор и постановка задачи исследования; Левшун Д. С., Веснин Д. В. – выбор решений, программная реализация и проведение экспериментов; Левшун Д. С., Котенко И. В. – обсуждение результатов экспериментов, анализ полученных результатов.*

***Ключевые слова:*** *информационная безопасность, анализ уязвимостей, BERT, CVE, CPE, CVSS, NVD.*

# PREDICTION OF VULNERABILITY CATEGORIES IN CONFIGURATIONS OF DEVICES USING ARTIFICIAL INTELLIGENCE METHODS

*Dmitry Levshun[4], Dmitry Vesnin[5], Igor Kotenko[6]*

***The purpose of the study:*** *investigation of the effectiveness of BERT modifications in solving the problem of predicting categories of vulnerabilities (CVE) for information system devices based on their configurations (CPE URIs).*

***Research methods:*** *natural language processing methods, cross-validation of artificial intelligence models, optimization of hyperparameters of artificial intelligence models.*

1 Левшун Дмитрий Сергеевич, кандидат технических наук, доктор философии компьютерных наук, старший научный сотрудник Лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: levshun@comsec.spb.ru. ORCID: 0000-0003-1898-6624.

2 Веснин Дмитрий Владимирович, магистр, младший научный сотрудник Лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: vesnin@comsec.spb.ru. ORCID: 0009-0004-8620-2996.

3 Котенко Игорь Витальевич, заслуженный деятель науки РФ, доктор технических наук, профессор, главный научный сотрудник и руководитель лаборатории проблем компьютерной безопасности, ФГБУН «Санкт-Петербургский Федеральный исследовательский центр Российской академии наук» (СПб ФИЦ РАН), г. Санкт-Петербург, Россия. E-mail: ivkote@comsec.spb.ru. ORCID: 0000-0001-6859-7120.

4 Dmitry S. Levshun, Ph.D. (in Tech.), Philosophy Doctor in Computer Science, Senior Researcher of Laboratory of Computer Security Problems at St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), St. Petersburg, Russia. E-mail: levshun@comsec.spb.ru. ORCID: 0000-0003-1898-6624.

5 Dmitry V. Vesnin, Master Student, Junior Researcher of Laboratory of Computer Security Problems at St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), St. Petersburg, Russia. E-mail: vesnin@comsec.spb.ru. ORCID: 0009-0004-8620-2996.

6 Igor v. Kotenko, Dr.Sc., Professor, Honored Worker of Science of the Russian Federation, Chief Scientist and Head of Laboratory of Computer Security Problems at St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), St. Petersburg, Russia. E-mail: ivkote@comsec.spb.ru. ORCID: 0000-0001-6859-7120.

***Results obtained:*** *based on the content of open vulnerability databases, we collected a data set that establishes relationships between preprocessed CPE URIs and the identified 24 CVE categories; we investigated the effectiveness of BERT, RoBERTa, XLM-RoBERTa and DeBERTaV3 in solving the problem of predicting CVE categories based on CPE URIs; we trained optimized BERT model to solve the problem of vulnerabilities prediction; we compared the resulting solution with available state-of-the-art.*

***Scientific novelty:*** *this work is one of the first in predicting device vulnerabilities based on their configuration, which emphasizes its scientific significance and novelty. Moreover, it is also one of the first works to explore BERT for the task of vulnerability prediction.*

***Contribution:*** *Levshun D. S., Kotenko I. V. – selection and formulation of the research problem; Levshun D. S., Vesnin D. V. – selection of solutions, software implementation and experiments; Levshun D. S., Kotenko I. V. – discussion of the experimental results, analysis of the results obtained.*

***Keywords:*** *information security, vulnerability analysis, BERT, CVE, CPE, CVSS, NVD.*

## Введение

Специалисты по информационной безопасности, ученые и энтузиасты по всему миру усердно работают над обеспечением защиты сетевых систем от вредоносной активности [1]. Данная задача усложняется широким разнообразием угроз и требований безопасности, особенно при защите систем Интернета вещей [2].

Один из популярных подходов к обеспечению безопасности сетевых систем – построение и анализ графов атак [3]. Такие графы позволяют отобразить все доступные пути для злоумышленников через систему, позволяя анализировать как предпосылки, так и последствия атак [4]. В этих графах каждое устройство представляется как узел, а связи между узлами определяются как сетевой политикой, так и потенциалом злоумышленника в компрометации этих устройств. В свою очередь, возможность компрометации устройств определяется наличием уязвимостей в их конфигурации [5].

Самый известный формат описания уязвимостей – CVE (Common Vulnerabilities and Exposures)[7]. CVE хранятся в различных открытых базах данных, наиболее популярной из которых является NVD (National Vulnerability Database)[8]. В NVD содержится почти 200 тысяч CVE, при этом каждая CVE имеет свой уникальный идентификатор, описание, ссылки, уязвимые конфигурации, и т. д.

Уязвимые конфигурации определяются с помощью логических выражений, которые объединяют несколько CPE URIs (Common Platform Enumeration Uniform Resource Identifiers)[9] с помощью логических операторов И и ИЛИ. CPE URI – это структурированная схема именования для всех видов приложений, операционных систем, прошивок и аппаратного обеспечения.

Проблема заключается в том, что конфигурации многих устройств не описаны в открытых базах данных. Это означает, что информация об их уязвимостях не может быть использована при построении графов атак. Таким образом, любое исследование, направленное на прогнозирование уязвимостей в неизвестных конфигурациях, является актуальным. И поскольку каждая уязвимость уникальна, большинство подходов направлены на прогнозирование их метрик.

Метрики уязвимостей описываются в соответствии с стандартом CVSS (Common Vulnerability Scoring System)[10]. В настоящее время наиболее часто используются 2-я (CVSS v2) и 3-я версии (CVSS v3), в то время как 4-я версия была только что представлена и пока не используется в открытых базах данных (CVSS v4). Эти стандарты содержат несколько метрик уязвимостей, 12 представлен в CVSS v2, 9 – CVSS v3.

Для построения графов атак наиболее важны следующие метрики: вектор доступа (access vector, представлен в CVSS v2 и CVSS v3); необходимые привилегии (privileges required, представлены только в CVSS v3,); и получаемые привилегии (obtain privileges, представлены только в CVSS v2 в рамках NVD).

Эти три метрики определяют условия необходимые для успешной эксплуатации уязвимости и ее последствия, а именно, как подключиться к уязвимому устройству (access vector), какие привилегии требуются для эксплуатации уязвимости (privileges required) и какие привилегии получает злоумышленник после эксплуатации (obtained privileges). В предыдущей работе авторов данные метрики были использованы, чтобы разделить все CVE на 24 категории [6].

Ключевым техническим нововведением в области искусственного интеллекта при создании системы BERT (Bidirectional Encoder Representations from Transformers — «двунаправленные представления

7 Официальный веб сайт проекта CVE: https://cve.mitre.org/
8 Официальный веб-сайт базы уязвимостей NVD: https://nvd.nist.gov/
9 Официальный веб-сайт описания конфигураций CPE: https://nvd.nist.gov/products/cpe
10 Официальный веб-сайт описания метрик уязвимостей CVSS: https://www.first.org/cvss/

кодировщика для трансформеров») [7] является применение двунаправленного обучения трансформеров (широко используемой в настоящее время модели с механизмом «внимания») к языковому моделированию.

Целью данной работы является исследование эффективности модификаций BERT для прогнозирования этих категорий в устройствах информационных систем на основе их конфигураций. Данное исследование основано на следующем предположении: CPE URIs, связанные с одинаковыми категориями CVE, более похожи друг на друга, чем CPE URIs, связанные с другими категориями. Таким образом, становится возможным прогнозировать категории CVE для устройств на основе их CPE URIs.

Анализ научной литературы показал, что данная работа является одной из первых в области прогнозирования уязвимостей в устройствах на основе их конфигураций, что подчеркивает ее научную значимость и новизну. Более того, это также одна из первых работ, посвященная исследованию BERT для этой задачи.

### Анализ работ

В работе [8] представлен обзор прогнозирования уязвимостей в исходном коде с использованием графовых нейронных сетей (GNN). Авторы сравнили 11 современных методов по архитектуре GNN, по методам представлений графов, наборам данных, точности и F-мере. Важно отметить, что почти все работы использовали различные наборы данных, поэтому сравнить результаты сложно. Например, в представленном сравнении точность варьируется от 58.90 % до 97.40 %, а F-мера – от 36.00 % до 96.11 %. Основной вывод заключается в следующем: существует нехватка реальных наборов данных, поэтому очень важно создать большую базу данных с образцами реального уязвимого исходного кода.

Систематический обзор литературы по обнаружению уязвимостей в программном обеспечении представлен в [9]. Авторы проанализировали 55 исследований, опубликованных с 2015 по 2021 год. Авторы сгруппировали эти исследования в 7 категорий, а именно, нейронные сети, машинное обучение, статический и динамический анализ, клонирование кода, классификация, модели и фреймворки, а также другие для исследований, которые не могут быть включены в эти категории. Было показано, что стратегии машинного обучения широко используются для обнаружения уязвимостей в программном обеспечении, поскольку они позволяют легко анализировать большие объемы данных. Несмотря на разработку многочисленных систем для обнаружения уязвимостей в программном обеспечении, ни одна из них не смогла точно определить конкретный тип обнаруженной уязвимости.

Авторы [10] разработали модель автоматической классификации уязвимостей. Данная модель объединяет TF-IDF (TF – Term Frequency, IDF – Inverse Document Frequency), IG (Information Gain) и глубокие нейронные сети (DNN). TF-IDF используется для определения частоты и важности каждого слова в описании уязвимости, в то время как IG используется для выбора признаков. Затем DNN используется для создания классификатора уязвимостей. Эффективность предложенной модели была проверена на NVD. По сравнению с методами SVM (Support Vector Machine), NB (Naive Bayes) и kNN (k-Nearest Neighbour). Модель авторов показала следующие результаты: аккуратность (accuracy) 87 %, точность (precision) 85 %, полнота (recall) 82 % и F-мера 81 %.

В работе [11] представлен обзор работ по автоматическому обнаружению и прогнозированию уязвимостей программного обеспечения. В этой работе технологии глубокого обучения были разделены на подходы для автоматического обнаружения уязвимостей, исправления программ и прогнозирования дефектов. В качестве основных будущих задач авторы выделили генерацию признаков и параметров, выбор и оценку моделей, а также формирование новых наборов данных.

Анализ работ по прогнозированию уязвимостей программного обеспечения представлен в [12]. Авторы проанализировали 180 исследований, их выводы следующие:

- в уязвимостях программного обеспечения существуют две основные области исследования: прогнозирование уязвимых компонентов программного обеспечения и прогнозирование новых уязвимостей программного обеспечения;
- большинство исследований в области уязвимостей создают собственные наборы данных, собирая информацию из баз данных уязвимостей, содержащих данные о реальном программном обеспечении;
- наблюдается увеличение интереса к моделям глубокого обучения и сдвиг к текстовому представлению исходного кода.

В [13] представлен обзор литературы по подготовке данных для прогнозирования уязвимостей программного обеспечения. Авторы рассмотрели 61 исследование и разработали таксономию подготовки данных для этой задачи. Подготовка данных была разделена на формирование требований (язык программирования, типы уязвимостей, детализация и контекст), сбор данных (реальный мир, синтетический или смешанный код), разметку (предоставленную, сгенерированную или основанную

на шаблонах) и очистку (несущественный код, шум, дублирование).

Анализ современных работ показывает, что прогнозирование уязвимостей с использованием различных типов входных данных находится в настоящее время на стадии активного развития. При этом важно отметить, что прогнозирование категорий уязвимостей на основе конфигураций устройств только начинает исследоваться, что подчеркивает научную значимость и новизну данного направления.

### Подход к исследованию

Подход, который был использован для исследования эффективности модификаций BERT, состоит из 4 шагов, начиная от подготовки данных и заканчивая оценкой результатов (рис. 1). Рассмотрим каждый шаг более подробно.

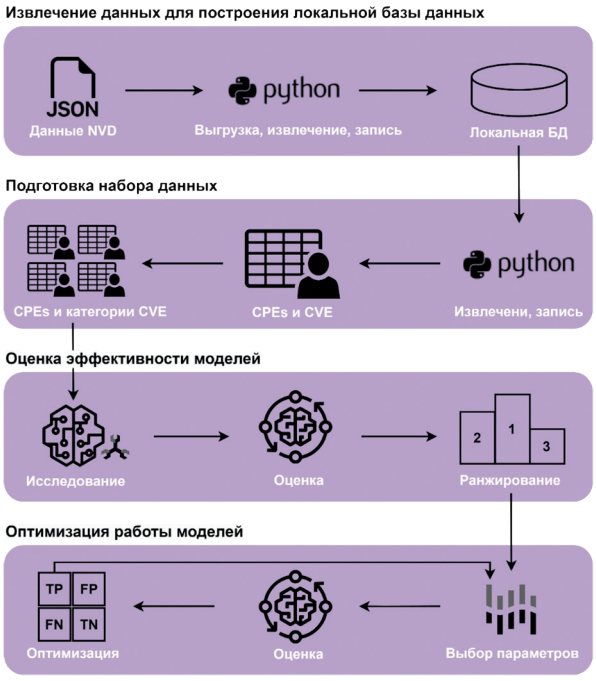


*Рис. 1. Подход для исследования эффективности BERT*

*Шаг 1. Извлечение данных для построения локальной базы данных.* На этом шаге данные о CVE выгружаются из файлов в формате JSON, которые доступны в открытой базе уязвимостей NVD. Данные файлы содержат информацию о каждой уязвимости и разделены на три основных типа:

1) файлы с CVE за определенный год;
2) файл с CVE, добавленными за последние 8 дней;
3) файл с CVE, измененными за последние 8 дней.

После загрузки и извлечения всех файлов, необходимо подготовить локальную базу данных для хранения CVE. Для этого в начале разрабатывается структура базы данных, которая затем наполняется данными, извлеченными из выгруженных файлов. В дальнейшем каждые 8 дней данные актуализируются и обновляются с помощью специального скрипта.

*Шаг 2. Подготовка набора данных.* Внутри базы данных, созданной на предыдущем шаге, данные об уязвимостях организованы в различные таблицы. Следовательно, для извлечения уязвимых конфигураций и их связей с CVE необходимо использовать различные SQL-запросы. Дополнительно, CPE URIs предобрабатываются и связываются с категориями уязвимостей. В осуществленных экспериментах предварительная обработка основана на замене символа «:» пробелом и удалении частей «cpe:2.3:» и «*»:

**cpe:2.3:a:gnu:glibc:2.38:*:*:*:*:*:*:* → a gnu glibc 2.38**

Итогом данного шага является набор данных для решения задачи с несколькими метками (*multilabel*). Это связано с тем, что различные CPE URI могут быть связаны с несколькими CVE, а CVE могут иметь различные значения метрик CVSS, а значит относиться к разным категориям. Кроме того, удаляются дубликаты из наборов данных.

*Шаг 3. Оценка эффективности моделей.* На данном шаге мы тестируем модели искусственного интеллекта на наборе данных, полученном на предыдущем шаге. С учетом особенностей BERT, перед подачей CPE URIs, представляющих собой строки текста, к ним применяется токенизация и паддинг.

Важно отметить, что на данном шаге обучение каждой модели осуществляется несколько раз с использованием различных частей набора данных (кросс-валидация). Затем, для каждой метрики эффективности рассчитывается среднее значение, а также среднеквадратичное отклонение. Задачей данного шага является отбор более эффективных моделей искусственного интеллекта для дальнейшей оптимизации их параметров.

*Шаг 4. Оптимизация работы моделей.* Цель данного шага – подобрать оптимальные гиперпараметры моделей для решаемой задачи, избегая переобучения. В рамках экспериментов оптимизация гиперпараметров осуществлялась с использованием фреймворка Optuna [14]. Итогом работы данного шага является модель, оптимизированная для прогнозирования категорий CVE на основе CPE URI.

### Полученный набор данных

В рамках данной работы был создан набор данных для прогнозирования категорий уязвимостей устройств на основе их конфигураций (табл. 1).

*Таблица 1*

*Набор данных для прогнозирования уязвимостей*

| $c_1$ | | $c_2$ | | $c_3$ | | $c_4$ | |
|---|---|---|---|---|---|---|---|
| True | False | True | False | True | False | True | False |
| 2261 | 92779 | 5723 | 89317 | 7526 | 87514 | 66684 | 28356 |
| $c_5$ | | $c_6$ | | $c_7$ | | $c_8$ | |
| True | False | True | False | True | False | True | False |
| 0 | 95040 | 0 | 95040 | 0 | 95040 | 6 | 95034 |
| $c_9$ | | $c_{10}$ | | $c_{11}$ | | $c_{12}$ | |
| True | False | True | False | True | False | True | False |
| 288 | 94752 | 15676 | 79364 | 523 | 94517 | 28266 | 66774 |
| $c_{13}$ | | $c_{14}$ | | $c_{15}$ | | $c_{16}$ | |
| True | False | True | False | True | False | True | False |
| 23 | 95017 | 8 | 95032 | 0 | 95040 | 102 | 94938 |
| $c_{17}$ | | $c_{18}$ | | $c_{19}$ | | $c_{20}$ | |
| True | False | True | False | True | False | True | False |
| 0 | 95040 | 138 | 94902 | 32 | 95008 | 157 | 94883 |
| $c_{21}$ | | $c_{22}$ | | $c_{23}$ | | $c_{24}$ | |
| True | False | True | False | True | False | True | False |
| 102 | 94938 | 7353 | 87687 | 141 | 94899 | 7563 | 87477 |

В данном наборе CPE URI связаны с 24 категориями CVE следующим образом:
**cpe,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16,c17,c18,c19,c20,c21,c22,c23,c24**
**a markdown it_project markdown it,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0**
**a oracle jd_edwards_enterpriseone_tools,0,0,0,1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0**
**a replit crosis,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0**

где cpe представляет собой предобработанные CPE URIs, а $c_1$ – $c_{24}$ являются бинарными отображениями наличия связи между CPE URI и соответствующей категорией CVE.

Всего в наборе данных 95 040 строк, 26 848 из которых относят одну CPE URI к нескольким категориям уязвимостей (задача с несколькими метками, multilabel). При этом важно отметить, что для категорий $c_5$, $c_6$, $c_7$, $c_{15}$ и $c_{17}$ на данный момент примеры не представлены, что позволяет снизить размерность решаемой задачи с 24 меток до 19. Также отметим, что часть категорий имеет очень малое количество примеров, что затрудняет эффективное решение поставленной задачи ввиду сильной несбалансированности данных.

### Полученные результаты

В рамках эксперимента, были протестированы базовые версии следующий моделей – BERT [7], RoBERTa [15], XLM-RoBERTa [16], DeBERTaV3 [17] (табл. 2).

*Таблица 2*

*Параметры протестированных моделей*

| Модель | Кол-во слоев | Кол-во скрытых слоев | Кол-во параметров, млн. |
|---|---|---|---|
| BERT | 12 | 768 | 110 |
| RoBERTa | 12 | 768 | 125 |
| XLM-RoBERTa | 12 | 768 | 125 |
| DeBERTaV3 | 12 | 768 | 184 |

На третьем шаге подхода лучшие результаты были продемонстрированы BERT, поэтому только данная модель была передана на оптимизацию гиперпараметров (табл. 3).

Полученное решение способно прогнозировать категории CVE на основе CPE URI с аккуратностью (*accuracy*) равной 0.7226. Насколько нам известно, сравнение данных результатов возможно только с предыдущими работами авторов (табл. 4) [18, 19].

*Таблица 3*

*Результаты оптимизации гиперпараметров*

| Параметр | Исследованные значения | Оптимальное значение |
|---|---|---|
| learning_rate | от 9e-5 до 1e-5 с шагом 1e-5, 9e-4, 8e-4 | 7e-05 |
| warm_up_epochs | от 0.00 до 1.50 с шагом 0.10 | 0.40 |
| weight_decay | от 0.00 до 0.05 с шагом 0.01 | 0.00 |

*Таблица 4*

*Сравнение полученного решения с аналогами*

| Подход | Метод | Задача | Модели | Аккуратность |
|---|---|---|---|---|
| [18] | Прогнозирование категорий CVE | Классификация по одной метке | Random Forest | 0.6450 |
| [19] | Прогнозирование метрик CVSS и объединение прогнозов | Классификация по множеству меток | Модификации BERT | 0.7382 |
| Предлагаемый | Прогнозирование категорий CVE | Классификация по множеству меток | Модификации BERT | 0.7226 |

Отметим, что хотя не удалось превзойти результаты [19], предлагаемый в данной статье подход также является перспективным. В дальнейшем, при улучшении и доработке использованного набора данных, результаты данного подхода могут превзойти полученные ранее результаты. Это означает, что на данный момент не представляется возможным предположить, какой из подходов покажет большую эффективность.

## Заключение

В работе была исследована эффективность таких моделей искусственного интеллекта, как BERT, RoBERTa, XLM-RoBERTa и DeBERTa-v3, для решения задачи прогнозировании категорий уязвимостей в устройствах информационных систем на основе их конфигурации. По итогам экспериментов наилучшие результаты были достигнуты с применением BERT, где аккуратность прогнозов составила 0.7226.

Отметим, что в процессе исследования возник ряд трудностей, связанных с несбалансированностью созданного набора данных. Поэтому в рамках дальнейших исследований, планируется проведение новых экспериментов на расширенном наборе данных, что, как ожидается, позволит улучшить полученные результаты.

Более того, планируется исследовать другие модификации BERT для решения задачи прогнозирования уязвимостей.

*Исследование выполнено за счет гранта Российского научного фонда № 22–71–00107, https://rscf.ru/project/22-71-00107/.*

***Рецензент:*** *Лаута Олег Сергеевич, доктор технических наук, профессор кафедры комплексного обеспечения информационной безопасности Государственного университета морского и речного флота имени адмирала С. О. Макарова, Санкт-Петербург, Россия. E-mail: laos-82@yandex.ru*

**Литература**

1. *Li Y., Huang G., Wang C., Li Y. Analysis framework of network security situational awareness and comparison of implementation methods // EURASIP Journal on Wireless Communications and Networking. 2019. Vol. 2019. P. 1–32. DOI: 10.1186/s13638-019-1506-1.*
2. *Израилов К. Е., Левшун Д. С., Чечулин А. А. Модель классификации уязвимостей интерфейсов транспортной инфраструктуры «умного города» // Системы управления, связи и безопасности. 2021. №. 5. С. 199–223. DOI: 10.24412/2410-9916-2021-5-199-223.*
3. *Lallie H. S., Debattista K., Bal J. A review of attack graph and attack tree visual syntax in cyber security // Computer Science Review. 2020. Vol. 35. P: 100219. DOI: 10.1016/j.cosrev.2019.100219.*
4. *Федорченко Е. В., Котенко И. В., Федорченко А. В., Новикова Е. С., Саенко И. Б. Оценивание защищенности информационных систем на основе графовой модели эксплойтов // Вопросы кибербезопасности. 2023. № 3 (55). С.23-36. DOI:10.21681/2311-3456-2023-3-23-36.*
5. *Kotenko I., Izrailov K., Buinevich M., Saenko I., Shorey R. Modeling the Development of Energy Network Software, Taking into Account the Detection and Elimination of Vulnerabilities // Energies. 2023. Volume 16, Issue 13, 5111. P.1-40. https://doi.org/10.3390/en16135111.*
6. *Levshun D., Chechulin A. Vulnerability Categorization for Fast Multistep Attack Modelling // Proceedings of the 33rd Conference of the Open Innovations Association FRUCT. May 24-26, Zilina, Slovakia. 2023. P. 169-175. DOI: 10.23919/FRUCT58615.2023.10143048.*

7. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol.1. 2019. P.4171–4186. DOI:10.18653/v1/N19-1423.
8. Katsadouros E., Patrikakis C. A Survey on Vulnerability Prediction using GNNs // Proceedings of the 26th Pan-Hellenic Conference on Informatics. 2022. P. 38-43. DOI: 10.1145/3575879.3575964.
9. Eberendu A. C., Udegbe V. I., Ezennorom E. O., Ibegbulam A. C., Chinebu T. I. A systematic literature review of software vulnerability detection // European Journal of Computer Science and Information Technology. 2022. Vol. 10. No. 1. P. 23–37. DOI: 10.37745/ejcsit.2013.
10. Huang G., Li Y., Wang Q., Ren J., Cheng Y., Zhao, X. Automatic classification method for software vulnerability based on deep neural network // IEEE Access. 2019. Vol. 7. P. 28291-28298. DOI: 10.1109/ACCESS.2019.2900462.
11. Shen Z., Chen S. A survey of automatic software vulnerability detection, program repair, and defect prediction techniques // Security and Communication Networks. 2020. Vol. 2020. P. 1–16. DOI: 10.1155/2020/8858010.
12. Kalouptsoglou I., Kalouptsoglou I., Siavvas M., Ampatzoglou A., Kehagias D., Chatzigeorgiou A. Software vulnerability prediction: A systematic mapping study // Information and Software Technology. 2023. P. 107303. DOI: 10.1016/j.infsof.2023.107303.
13. Croft R., Xie Y., Babar M. A. Data preparation for software vulnerability prediction: A systematic literature review // IEEE Transactions on Software Engineering. 2022. Vol. 49. No. 3. P. 1044-1063. DOI: 10.1109/TSE.2022.3171202.
14. Akiba T., Sano S., Toshihiko Y., Ohta T., Koyama M. Optuna: A next generation hyperparameter optimization framework // Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. 2019. P. 2623–2631. DOI: 10.1145/3292500.3330701.
15. Liu Z., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V. A robustly optimized BERT pre-training approach with post-training // Proceedings of the China National Conference on Chinese Computational Linguistics. Cham: Springer International Publishing, 2021. P. 471–484. DOI: 10.48550/arXiv.1907.11692.
16. Conneau A., Chaudhary V., Wenzek G., Guzman F., Grave E., Ott M., Zettlemoyer L., Stoyanov V. Unsupervised Cross-lingual Representation Learning at Scale // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics. 2020. DOI: 10.48550/arXiv.1911.02116.
17. He P., Gao J., Chen W. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing // Proceedings of the Eleventh International Conference on Learning Representations. 2022. DOI: 10.48550/arXiv.2111.09543.
18. Levshun D. Comparative analysis of machine learning methods in vulnerability categories prediction based on configuration similarity // Proceedings of the 16th International Symposium on Intelligent Distributed Computing (IDC-2023). September 13–15, Hamburg, Germany. 2023. P. 231–242.
19. Levshun D., Vesnin D. Exploring BERT for Predicting Vulnerability Categories in Device Configurations // Proceedings of the 10th International Conference on Information Systems Security and Privacy (ICISSP 2024). February 26–28, Rome, Italy. 2024. P. 452–461. DOI: 10.5220/0012471800003648.