

УДК 621.396

Обзорная статья

<https://doi.org/10.32603/2071-8985-2024-17-8-48-64>**Методы и средства расчета и применения метрик кода программных продуктов: систематическое картографирование литературы****А. О. Корзников[✉], Н. Н. Дацун**

Пермский государственный национальный исследовательский университет, Пермь, Россия

[✉] artemkorz@mail.ru

Аннотация. Систематическое картографирование литературы (SMS) – это методология, которая предоставляет средства для систематического анализа темы исследования с целью наблюдения общего ландшафта исследования и определения количества и типа доступных исследований и их результатов. Метрика программного продукта выражается числовым значением и определяет некоторое свойство программы. Обычно метрики применяют для анализа участков исходного кода приложений. Однако на значение метрики могут оказать влияние как шаблоны проектирования, стиль написания программы разработчиком, так и модификация кода с целью искусственного завышения результатов вычисления метрик. Все это делает метрики нетривиальным инструментом для оценки программных продуктов, и возникает задача их корректной интерпретации и применения для практических задач. В данной статье было проведено систематическое картографирование литературы с целью построения ландшафта исследований в данной области, выявления методов и средств практического использования метрик кода для анализа приложений, в том числе обзор существующих программных решений. Сформирован корпус релевантных документов и получены ответы на поставленные исследовательские вопросы. Построены распределения публикаций по типу документа, географическому принципу и числу авторов. Предложена классификация задач исследований с помощью метрик и их наборов. Проанализированы зависимости между метриками и их наборами, языками программирования. Исследованы распределения метрик и их наборов по решаемым задачам в соответствии с предложенной классификацией.

Ключевые слова: систематическое картографирование литературы, анализ приложений, исходный код, метрика кода, применение метрик кода

Для цитирования: Корзников А. О., Дацун Н. Н. Методы и средства расчета и применения метрик кода программных продуктов: систематическое картографирование литературы // Изв. СПбГЭТУ «ЛЭТИ». 2024. Т. 17, № 8. С. 48–64. doi: 10.32603/2071-8985-2024-17-8-48-64.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Review article

**Methods for Calculation and Application of Software Code Metrics:
A Systematic Mapping Study****A. O. Korznikov[✉], N. N. Datsun**

Perm State University, Perm, Russia

[✉] artemkorz@mail.ru

Abstract. Systematic mapping study (SMS) is a methodology that provides a means to systematically analyze a research topic, in order to observe the overall research landscape and determine available studies, their amount, types, and results. A metric of a software product is expressed as a numeric value, defining a certain

property of the program. Metrics are usually used to analyze sections of the source code of software applications. However, the value of a metric can be influenced by both design patterns, the style of writing the program by the developer, and code modification in order to artificially inflate the results of calculating metrics. All this makes metrics a non-trivial tool for evaluating software products, requiring their correct interpretation and application for practical tasks. In this paper, a systematic mapping of the literature was carried out in order to outline the current research landscape in this area, identify methods and means for practical use of code metrics for application analysis, including an overview of existing solutions. A corpus of relevant documents was formed and answers to the research questions were received. The distributions of publications by document type, geographical principle, and number of authors were constructed. The classification of research tasks using metrics and their sets was proposed. The dependencies between metrics, their sets, and programming languages were analyzed. The distributions of metrics and their sets for the problems to be solved were studied in accordance with the proposed classification.

Keywords: systematic mapping study, software analysis, source code, code metric, application of code metrics

For citation: Korznikov A. O., Datsun N. N. Methods for Calculation and Application of Software Code Metrics: A Systematic Mapping Study // LETI Transactions on Electrical Engineering & Computer Science. 2024. Vol. 17, no. 8. P. 48–64. doi: 10.32603/2071-8985-2024-17-8-48-64.

Conflict of interest. The authors declare no conflicts of interest.

SMS [1]–[5] – это воспроизводимое исследование литературы, включающее следующие этапы:

- 1) определение исследовательских вопросов;
- 2) поиск публикаций;
- 3) скрининг полученных документов;
- 4) выделение ключевых слов;
- 5) извлечение данных и картографирование процессов.

Структура статьи соответствует данным этапам и протоколу SMS.

Определение исследовательских вопросов.

В SMS исследовательские вопросы (ИВ) необходимы для определения направления, области и цели работы. Для проведения исследования были поставлены следующие вопросы:

– ИВ1: Как выглядит ландшафт исследований за 2017–2023 гг. метрик исходного кода?

– ИВ2: Какие практические задачи могут быть решены с помощью способов и подходов применения метрик?

– ИВ3: Какие существуют программные средства, осуществляющие расчет метрик кода и использующие их для решения практических задач?

Поиск проводился по источникам данных, представляющим собой научные электронные библиотеки. Также проведен поиск вручную по спискам источников наиболее релевантных публикаций, найденных в результате автоматического поиска.

В процессе предварительного исследования публикаций были найдены SMS [1]–[5] по тематике метрик за 2013–2022 гг. Публикация 2017 г., исследовавшая статьи за 2010–2015 гг., – наиболее общая и цитировалась 141 раз, в то время как

иные исследования ставили перед собой задачу изучить более узкий круг научных статей. Поэтому в стратегии выбран временной интервал: 2017–2023 гг.

В качестве основных источников данных использовались:

- научные электронные библиотеки eLIBRARY.RU, IEEE Xplore Digital Library (IEEE Xplore DL) и ACM Digital Library (ACM DL);
- академическая база данных ScienceDirect;
- онлайн-сервис по предоставлению информации о научных, технических и медицинских книгах и журналах SpringerLink.

Дополнительными источниками служат: научно-информационная социальная сеть ResearchGate и электронный архив с открытым доступом – arXiv.

Для поиска публикаций по тематике исследовательских вопросов были выделены ключевые слова: метрики, исходный код, программный код, метрики кода, метрики программ, анализ метрик, метрики объектно-ориентированного программирования. Они были переведены на английский язык для возможности поиска на обозначенных зарубежных ресурсах: metrics, source code, program code, code metrics, software metrics, metrics analysis, object-oriented metrics.

Поиск публикаций. Корпус первичных документов сформирован из публикаций, полученных по запросам к источникам данных на основе выделенных ключевых слов безотносительно к их содержанию.

Табл. 1. Запросы для получения метаданных
Tab. 1. Requests for metadata

Источник	Запрос в нотации источника	Комментарий
eLIBRARY.RU	Что искать: метрики AND («исходного кода» OR «программного продукта» OR «оценка качества»); годы публикации: 2017–2023	Поиск по заголовкам публикаций. Исключены публикации, в которых не используются метрики исходного кода
IEEE Xplore DL	Keywords: («Document Title»:metrics) AND («Document Title»:«source code»); Publication year: 2017–2023	Поиск по заголовкам публикаций
IEEE Xplore DL	Keywords: («Document Title»:metrics) AND («Document Title»:software); Publication year: 2017–2023	Поиск по заголовкам публикаций
ACM DL	[Title: metrics] AND [[Title: «source code»] OR [Title:«software metrics»]] AND [E-Publication Date: (01/01/2017 TO 09/30/2023)]	Поиск по заголовкам публикаций
ACM DL	Title: metrics] AND [[Abstract: «source code»] OR [Abstract: «software metrics»]] AND [E-Publication Date: (01/01/2017 TO 09/30/2023)]	Поиск по аннотациям, были исключены публикации, не относящиеся к типу «Research article»
ScienceDirect	Year: 2017–2023; Title, abstract, keywords: metrics; Title: «source code»	Year: 2017–2023; Title, abstract, keywords: metrics; [Title: «software metrics»]
SpringerLink	with all the words: 'software AND code AND metrics'; without the words: Engineering; show documents published: between 2017 and 2023; filter: Computer Science, Computational Intelligence	Поиск выполнен с помощью старой версии интерфейса. Большой объем публикаций, в том числе множество нерелевантных

Табл. 2. Распределение публикаций корпуса первичных по годам
Tab. 2. Distribution of primary set publications by year

Источник	Число публикаций по годам							Всего
	2017	2018	2019	2020	2021	2022	2023	
eLIBRARY.RU	4	6	6	2	1	4	4	27
IEEE Xplore DL	22	17	6	7	6	4	4	66
ScienceDirect	4	4	2	3	2	3	8	26
ACM DL	11	6	4	5	5	3	7	41
SpringerLink	10	19	18	22	15	26	25	135

Табл. 3. Распределение релевантных публикаций по годам
Tab. 3. Distribution of relevant publications by year

Источник	Число публикаций по годам							Всего	
	2017	2018	2019	2020	2021	2022	2023	публ.	%
eLIBRARY.RU	–	–	–	–	–	–	1	1	2.0
IEEE Xplore DL	9	8	1	4	4	2	1	29	59.2
ACM DL	1	1	2	2	–	–	–	6	12.2
ScienceDirect	2	1	1	–	–	–	–	4	8.2
SpringerLink	2	2	3	–	1	–	1	9	18.4
Итого	14	12	7	6	5	2	3	49	100.0

На первом этапе формирования корпуса документов метаданные экспортируются для дальнейшего получения полных текстов и анализа публикаций. Запросы для получения выборок публикаций и формирования файлов с метаданными представлены в табл. 1. Диапазон лет: 2017–2023 гг. Из полученных файлов сформирован корпус первичных публикаций.

Для запросов использован расширенный поиск. Не рассматривались только публикации с предварительным просмотром.

В табл. 2 представлено распределение числа публикаций из корпуса первичных по годам и источникам.

Скрининг документов. На основе корпуса первичных публикаций сформирован корпус уникальных публикаций посредством удаления дубликатов. Далее из полученного корпуса уникальных отобраны публикации, удовлетворяющие целям исследования для ответа на поставленные исследовательские вопросы. Для этого были определены критерии включения и исключения.

Критерии включения:

- в публикациях рассматриваются обозначенные исследовательские вопросы, используются выделенные ключевые слова;
- публикации представляют материалы научных журналов, статьи научных мероприятий;
- тематика публикаций связана с расчетом метрик кода или их применением для практических задач.

Критерии исключения:

- объем публикации менее 5 страниц;
- систематические обзоры литературы.

В результате применения этих критериев был получен корпус релевантных публикаций [6]–[54], распределение количества которых по годам представлено в табл. 3. В результате отбора релевантных публикаций из библиотеки eLIBRARY.RU осталась единственная публикация. Это произошло вследствие удаления из корпуса повторяющихся публикаций из различных источников, применения критериев включения-исключения, а также в связи с тем, что публикации индексируются в нескольких источниках одновременно.

Выделение ключевых слов с помощью аннотаций. Для детального изучения ландшафта исследований предметной области, обнаружения

задач и определения практических вопросов, разрешаемых при помощи метрик, была произведена классификация публикаций.

Для корпуса релевантных документов по аннотациям был построен частотный словарь, представленный в табл. 4. Из него были удалены стоп-слова и понятия из запросов для поиска.

Данный частотный словарь совпадает с таким для ИВ1 и ИВ2, поскольку для анализа ландшафта исследований требуется рассмотреть все релевантные публикации, а также каждая из них содержит описание используемых методов, подходов и технологий для расчета метрик, исследования их применимости для определенных задач или интерпретации вычисленных значений.

Для ответа на ИВ3 были отобраны публикации, включающие непосредственную разработку библиотеки, приложения или их усовершенствование. Для отобранных публикаций был построен частотный словарь, представленный в табл. 5, из которого были исключены стоп-слова, а также понятия, связанные с разработкой приложения: «tool», «implement», «implementation», «application», «apply».

Был проведен контекстный анализ ключевых терминов для определения взаимосвязи между ними, соотнесения методов и задач с целью их использования. На рис. 1 представлены результаты для ответов на ИВ1 и ИВ2, на рис. 2 – ИВ3.

Табл. 4. Частотный словарь для ИВ1 и ИВ2

Tab. 4. Frequency dictionary for research questions 1 and 2

Понятие	Перевод	Количество
model	модель	68
prediction	прогноз	63
vulnerability	уязвимость	58
quality	качество	53
design	проектирование	45
pattern	шаблон	44
tool	инструмент	41
security	безопасность	41
technique	технология	41
approach	подход	41
class	класс	40
fault	дефект	38
feature	особенность	33
development	разработка	32
method	метод	31
learn	обучать	31
compare	сравнивать	29
level	уровень/степень	28
machine	машинный	28
analysis	анализ	28
test	тест	27
smell	запах	27
detection	обнаружение	24
problem	проблема	24

Табл. 5. Частотный словарь для ИВ3

Tab. 5. Frequency dictionary for research questions 3

Понятие	Перевод	Количество
design	проектирование	33
pattern	шаблон	30
quality	качество	29
vulnerability	уязвимость	29
class	класс	26
development	разработка	25
approach	подход	21
security	безопасность	20
method	метод	20
level	уровень/степень	19
prediction	прогноз	19
compare	сравнивать	19
analysis	анализ	19
model	модель	18
technique	технология	16
type	тип	15
problem	проблема	14
learn	обучать	13
decision	решение	13
support	поддержка	13
smell	запах	12
measurement	оценка	12

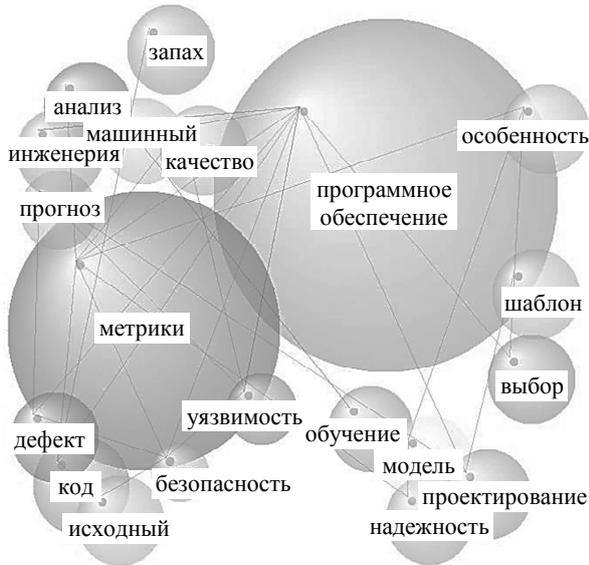


Рис. 1. Контекстный анализ корпусов по ключевым словам для ответа на ИВ1 и ИВ2
Fig. 1. Contextual analysis for research question 1 and 2 by keywords

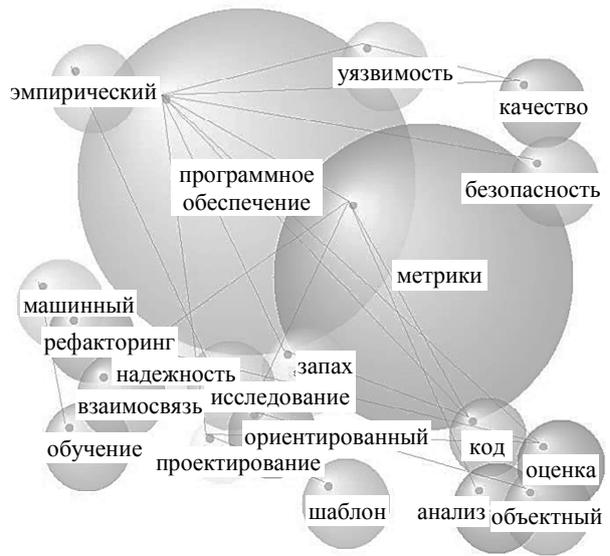


Рис. 2. Контекстный анализ корпусов по ключевым словам для ответа на ИВ3
Fig. 2. Contextual analysis for research question 3 by keywords

Извлечение данных и картографирование процессов. Картографирование выполняют с помощью «систематических карт» – таблиц или графиков/диаграмм. Количественные данные представляют в виде графиков/диаграмм, а качественные – в табличном виде [55]. По типу документа найденные публикации делятся на материалы научных мероприятий (семинаров, симпозиумов, конференций) и статьи журналов. Распределение представлено на рис. 3.

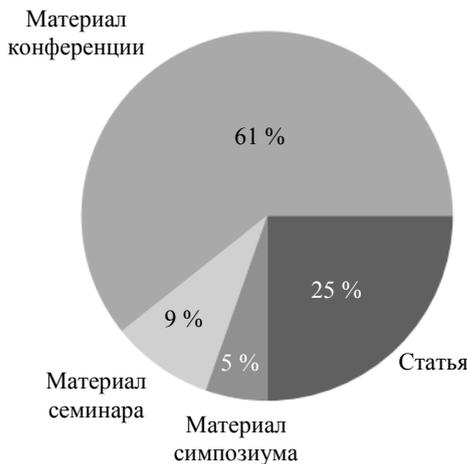


Рис. 3. Распределение публикаций по типу документа
Fig. 3. Distribution of publications by document type

Для распределения релевантных публикаций по странам организаций авторов, была построена диаграмма (рис. 4).

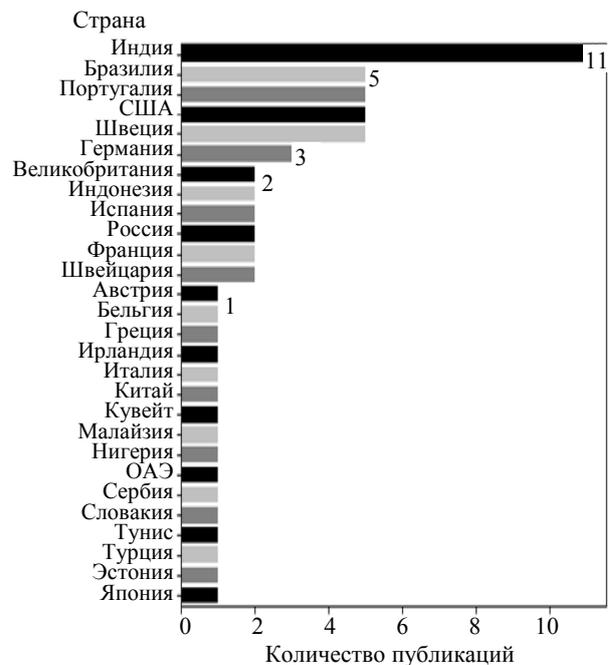


Рис. 4. Распределение публикаций по географическому принципу
Fig. 4. Distribution of publications by geographical principle

В табл. 6 отображена информация об организациях, в которых работают авторы наибольшего числа публикаций из корпуса релевантных.

Релевантные публикации были разделены по количеству авторов, распределение продемонстрировано на рис. 5.

Исследование метрик. Для получения более детальной информации об использовании метрик было необходимо рассмотреть содержание отобранных релевантных статей на предмет исполь-

Табл. 6. Топ-10 организаций, в которых работают авторы
Tab. 6. Top-10 author affiliations

Организация	Вуз	Количество публикаций
National Institute of Technology, India	+	5
University of Coimbra, Portugal	+	5
Polytechnic Institute of Coimbra, Portugal	+	4
Banaras Hindu University, India	+	3
Montclair State University, USA	+	3
ABB Corporate Research, India	–	3
Jawaharlal Nehru University, India	+	2
ООО Лаборатория «Вычислительная механика», Россия	–	1
Санкт-Петербургский политехнический университет Петра Великого, Россия	+	1
ИжГТУ им. М. Т. Калашникова, Россия	+	1

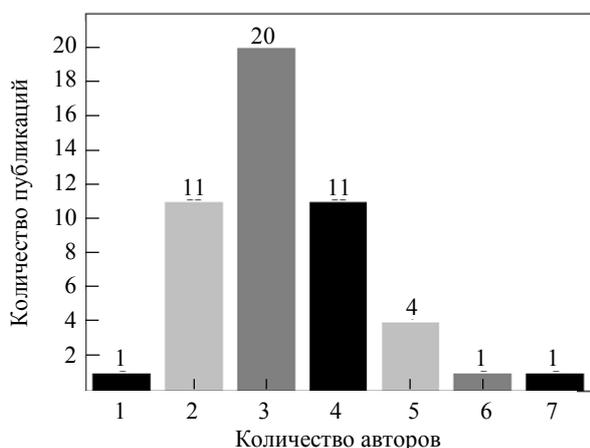


Рис. 5. Распределение публикаций по числу авторов
Fig. 5. Distribution of publications by the number of authors

зуемых метрик и их наборов, а также целей использования.

Поэтому был создан набор данных со следующими признаками: <название метрики, название набора метрик, решаемая задача, язык программирования, сочетание с технологиями машинного обучения>. Набор данных был дополнен признаком наличия совмещения с технологиями машинного обучения, так как обнаружилось частое совместное использование этого подхода с расчетом метрик на практике.

Общепринятой таксономии метрик нет, кроме определения их места в системе показателей качества программного продукта или системы (ГОСТ Р ИСО/МЭК 25023–2021). Отечественные авторы указывают, что существует множество различных классификаций метрик программного обеспечения [56], [57], и используют термин «группы метрик» [56], [58]. При исследовании

релевантных публикаций нашего SMS было выявлено, что терминология предметной области еще не устоялась: некоторые названия метрик и аббревиатуры не строго определены. Поэтому в данной статье также исследуется зависимость между метриками и их наборами.

На рис. 6 представлена тепловая карта этой зависимости. Она отражает, какие метрики могут быть отнесены к тому или иному набору и какие из них чаще всего применяются совместно. Более темным оттенком показан больший процент публикаций, в которых используется и определенная метрика, и набор метрик. В табл. 7 представлены используемые на диаграмме сокращения.

Задачи исследований, для которых авторы применяли метрики, предлагается разделить на следующие категории:

- «запахи кода» (Code Smell): выявление «запахов кода» (в том числе архитектурных проблем);

- дефекты (Fault): поиск/прогнозирование дефектов и ошибок в коде;

- метрики (Metrics): изучение метрик (сравнение с иными подходами и рассмотрение способов интеграции, определение границ оптимальных значений, способов вычисления и т. д.);

- шаблоны (Pattern): идентификация шаблонов (паттернов) проектирования и выявление клонов;

- качество (Quality): оценка/измерение качества кода;

- рефакторинг (Refactoring): непосредственно задача рефакторинга;

- надежность (Trustworthiness): оценка надежности кода;

- уязвимости (Vulnerability): поиск/прогнозирование уязвимостей в коде;

- другие: менее популярные (одиночные) задачи (отслеживание метрик между версиями, обнаружение заимствованного кода, оценка вклада разработчиков в проект, потребления энергии приложением и др.).

В табл. 7–8 представлены условные обозначения для наборов метрик, используемые далее при их именовании на рис. 6–10.

На рис. 7–8 представлены решаемые исследователями задачи с использованием тех или иных наборов метрик и конкретных метрик кода соответственно.

Табл. 7. Используемые обозначения названий наборов метрик

Tab. 7. Designations used for metrics sets

Обозначение	Название набора
H1	MOOD (метрики для объектно-ориентированного проектирования)
H2	QMOOD (метрики качества для объектно-ориентированного проектирования)
H3	Robert C. Martin
H4	Chidamber & Kemerer
H5	Cohesion metrics
H6	Lorenz and Kidd
H7	Li and Henry
H8	Lanza and Marinescu
H9	Halstead
H10	Structural metrics
H11	Al-Dallal and Briand
H12	Bieman and Kang
H13	Object-oriented
H14	Другие наборы

Табл. 8. Используемые обозначения названий метрик

Tab. 8. Designations used for metrics

Обозначение	Аббревиатура/название
M1	LOC
M2	LCOM (LCOM1)
M3	NOC
M4	CBO
M5	WMC
M6	DTI
M7	RFC
M8	NOM
M9	Ce
M10	Ca
M11	NOA
M12	DAM
M13	CBM
M14	IC
M15	CAM
M16	NPM
M17	MOA
M18	MFA
M19	AMC
M20	CC
M21	NOP
M22	NOPA
M23	LCOM3
M24	Fan out
M25	Fan in
M26	Cyclomatic complexity
M27	Statement declaration
M28	Sum cyclomatic modified
M29	Average essential
M30	Count line comment
M31	Number of methods calls
M32	Count path
M33	Max nesting

Также было проверено наличие закономерности между языком исходного кода и используемыми для его исследования метриками. На рис. 9 демонстрируется распределение метрик по языкам исходного кода. На диаграмме представлены все языки программирования, код на которых исследовался в рамках релевантных публикаций: Java, C++, C и другие.

На рис. 10 изображена диаграмма соотношения общего числа публикаций по теме и публикаций, использующих алгоритмы или технологии машинного обучения. В число отобранных статей изначально входило множество исследований, использовавших машинное обучение безотносительно цели. Это произошло ввиду тесной связи данной области с метриками, которые не относятся к измерению исходного кода. В число релевантных попали лишь те публикации, которые применяют машинное обучение совместно с подходом оценки исходного кода посредством метрик.

Обсуждение. В результате изучения ИВ1 выяснилось, что основной пик научных исследований в области расчета и применения метрик кода пришелся на 2017 г. (28.57 % от общего количества публикаций), и далее интерес к данной тематике постепенно снижался. Наибольшее число результатов было представлено в рамках научных мероприятий (75 % от общего числа), большинство из которых (50 %) было организовано институтом инженеров электротехники и электроники (IEEE). В разные годы при проведении мероприятия International Conference on Program Comprehension (ICPC) были представлены две публикации по данной тематике – это самый высокий показатель. Наиболее активно исследования проводились в Индии (22.45 %), 85.71 % авторских коллективов составляет 2–4 человека, абсолютное большинство из которых (86.36 % от общего числа) – сотрудники высших учебных заведений.

Ответ на ИВ2 показал, что в настоящее время исследователи занимаются определением оптимальных значений метрик и зависимостей между метриками [12], [16], [18], [29]–[31], [40], [54], их сравнением между собой для выявления наиболее показательных и эффективных с помощью методов машинного обучения [6], [9], [25], [26], [35]–[37], [42], [43], [49], [50], а также с другими методами изучения характеристик исходного кода программных продуктов [10], [24], [27], [29], [34], [48]. Так, нанопаттерны [48] рассматриваются в контексте их совместного применения с расчетом

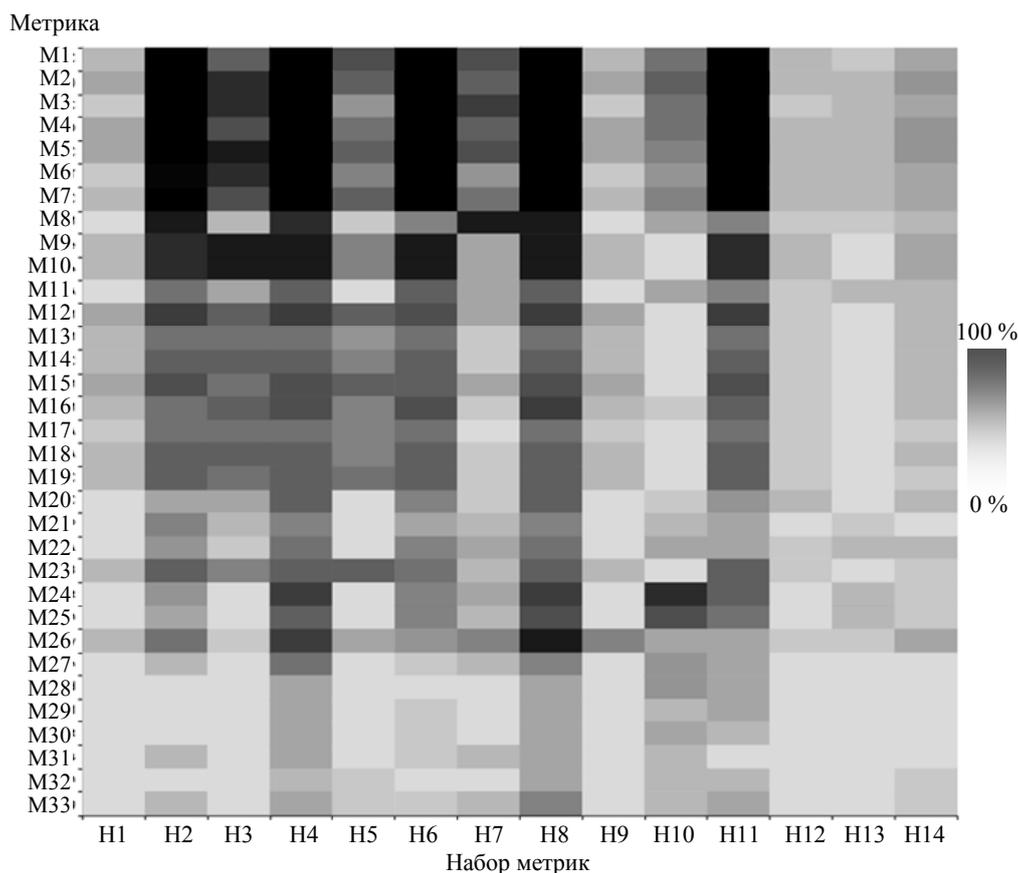


Рис. 6. Тепловая карта зависимости между метриками и наборами

Fig. 6. Heat map of the relationship between metrics and sets

метриков кода. Разрабатываются и апробируются подходы, методы и технологии для выявления ошибок [9]–[11], [20], [21], [23], [26], [47], [51] и уязвимостей в коде и архитектуре [19], [22], [33], [35], [43], [46]–[48], [54] приложений, выявления «запахов кода» [11], [28], [39], [41], [42], оценки качества проектирования [8], [32], [45], [49], [50], [52], [53] и надежности [34], [36] исходного кода, рефакторинга [6], [14], [17], [20], [40], [44]. Кроме того, исследуется возможность распознавания шаблонов проектирования и программирования [7], [16], [37], [52], их влияние на метрики, а также изменений между версиями [29].

По полученным данным можно сказать, что большинство метрик (72.73 % от общего количества) применялись для прогнозирования дефектов. Некоторые метрики универсальны, например количество строк кода (LOC) применяется для всех выделенных задач, число подклассов (NOC) – для 88.89 % задач, количество методов класса (WMC) и высота дерева наследования (DIT) – для 77.78 %. Относительно наборов метрик также можно выделить универсальные – метрики Chidamber and Kemerer, Lanza and Marinescu.

Кроме того, четко видны наиболее часто используемые метрики: LOC, LCOM1, NOC, WMC, DIT, CBO, RFC, McCabe’s cyclomatic complexity, NOM, Ce, Ca и fan out. Частота их использования составляет две трети от общего числа применений. Среди наборов метрик наиболее популярны: Lanza and Marinescu [6], [10]–[17], [19]–[26], [28]–[31], [34]–[37], [39], [41], [42], [45]–[52], [54]; Chidamber and Kemerer [6]–[15], [19]–[26], [28]–[31], [33]–[35], [37], [42]–[47], [49]–[52]; Lorenz and Kidd [6], [10]–[12], [15], [19]–[26], [28], [37], [46], [47], [49]–[51]; Al-Dallal and Briand [6], [7], [10], [11], [14], [15], [19]–[26], [33]–[35], [46], [47], [49]–[51] и QMOOD [6], [7], [10]–[12], [15], [19]–[21], [23]–[26], [30], [31], [45], [46], [50]–[52], [54].

Также существует закономерность между языком исходного кода и используемыми для его исследования метриками. Все рассмотренные языки, за исключением С, объектно-ориентированные, однако набор используемых метрик значительно различается. Например, число входящих (fan in) и исходящих (fan out) зависимостей часто применяется для кода на С-подобных языках С [9], [33]–[35], [40], [41], [43], [52] и С++ [9], [11], [18], [21], [33]–

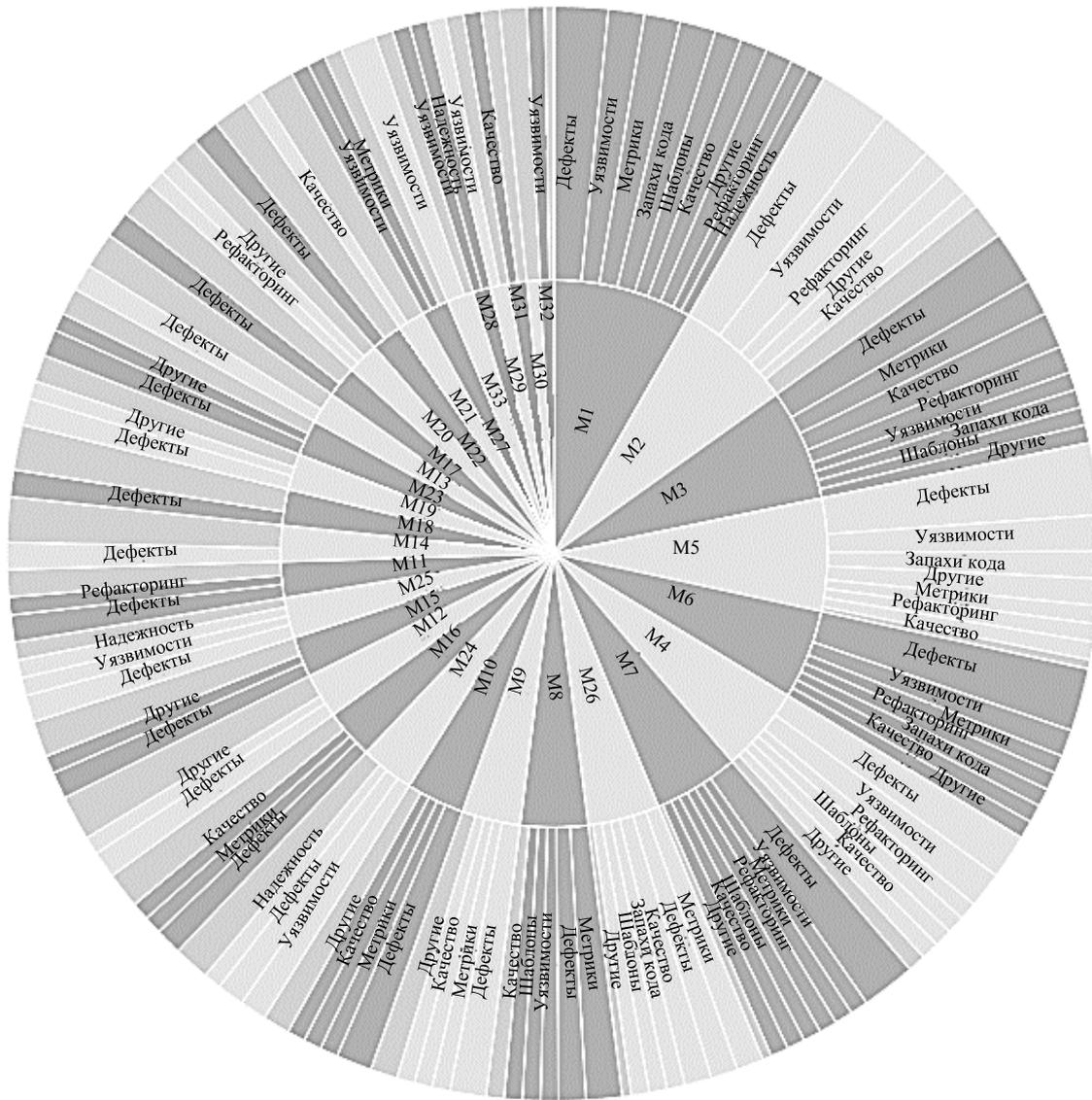


Рис. 7. Распределение метрик по решаемым задачам
 Fig. 7. Distribution of metrics by the tasks to be solved

[36], [41], [43], [53]. Для исследования кода на языке Java [6]–[7], [9]–[17], [19], [20], [22]–[26], [28], [30], [31], [39], [41], [42], [44], [46]–[52], [54] наиболее часто используются метрики из набора Chidamber and Kemerer (NOC, CBO, WMC, DIT). Анализируя код на других языках программирования, исследователи применяют большинство метрик, но не используют NOA, NOPA, CC.

Часто использование подхода расчета метрик сочетается с применением технологий машинного обучения для решения поставленных задач [6], [9], [10], [15], [17], [19], [24]–[26], [29], [33], [35]–[37], [42], [43], [49], [50] (36.73 % от общего количества). Наиболее часто они применяются для нахождения уязвимостей кода и дефектов (14.28 % от общего числа, 38.89 % исследований данных вопросов), а также менее распространенных задач (обнаружение заимствованного кода

[52], оценка потребления энергии [27] и т. д.). Зачастую технологии машинного обучения используются для упрощения расчета метрик на готовых наборах данных исходного кода (например, код mozilla [9], [34], [41]).

Для ответа на ИВЗ были проанализированы существующие программные средства для расчета метрик и их анализа. Исследование показало, что существуют приложения, комбинирующие расчет метрик исходного кода с подходами машинного обучения [6], [10], [25], [36], [42] или микро-паттернами [48]. При этом сочетания расчета метрик с машинным обучением наблюдается в 30–50 % публикаций ежегодно. Все приложения выполняют расчет и анализ метрик кода на определенном языке программирования [6], [7], [10], [12], [15]–[22], [24]–[26], [28], [30], [31], [40], [44]–[51], [53], [54] или группе близких (до трех

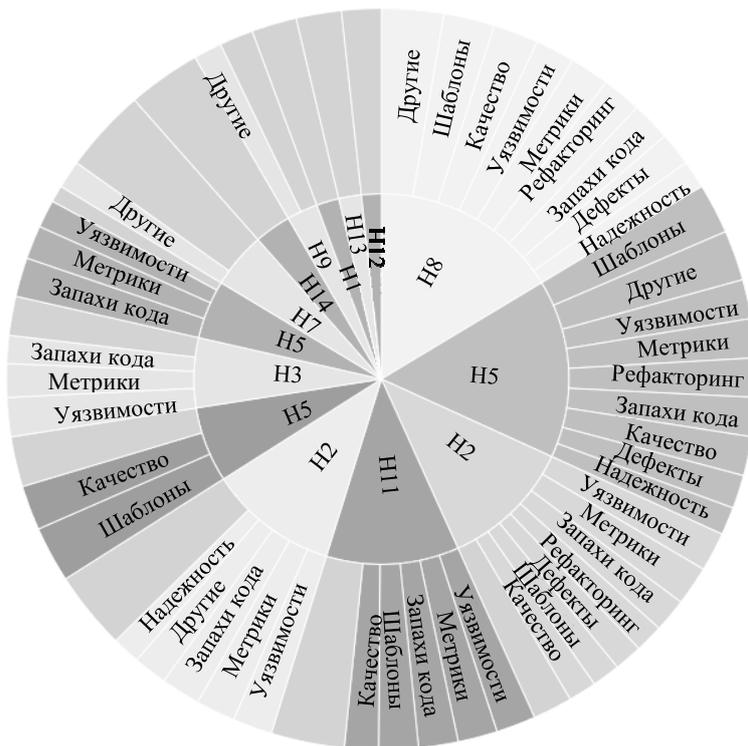


Рис. 8. Распределение наборов метрик по решаемым задачам
 Fig. 8. Distribution of metrics sets by the tasks to be solved

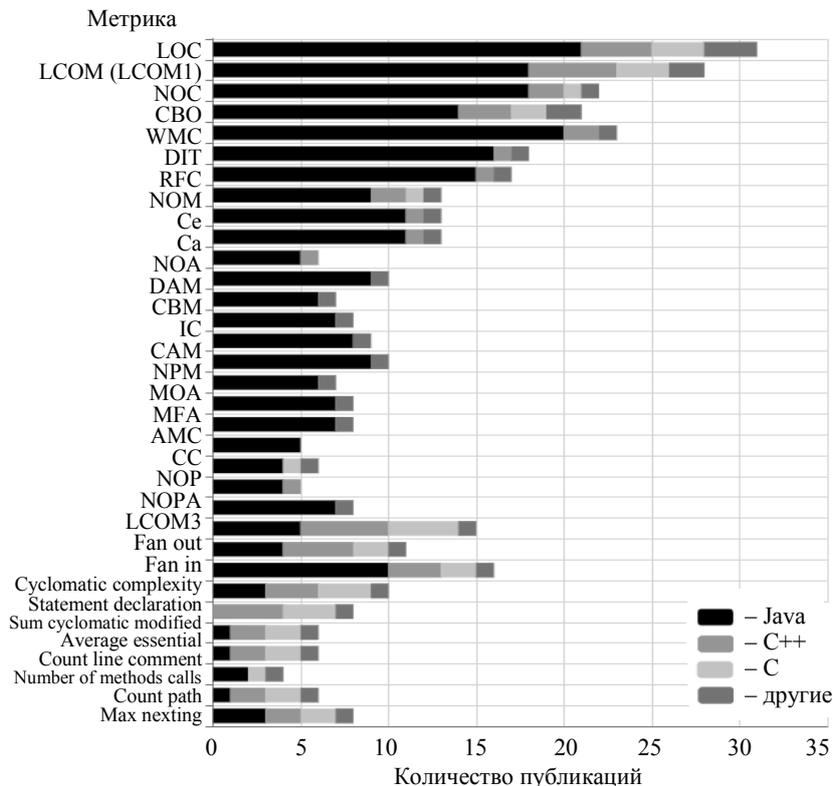


Рис. 9. Распределение метрик по языкам программирования
 Fig. 9. Distribution of metrics by programming languages

языков программирования) [9], [34], [43]. Существующие программные средства предназначены для решения некоторых задач из предложенной

нами классификации: оценка качества приложений [8], [45], [52], [53], прогноз количества уязвимостей в коде [46]–[48], [54], обнаружение недо-

статков проектирования [28], [39], [42], определение наличия ошибок в коде и предсказания их появления [10], [47] или устранения после внесения изменений в исходный код.

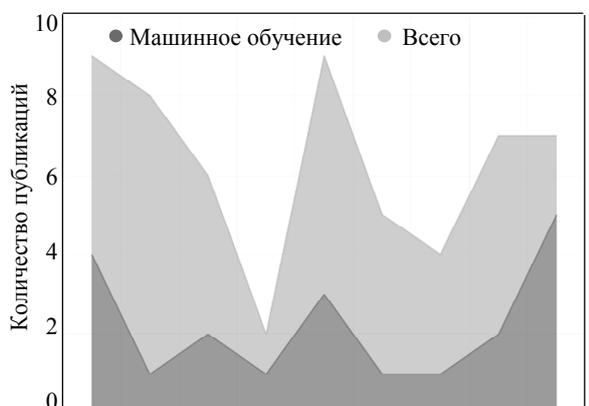


Рис. 10. Соотношение общего числа публикаций и публикаций, включающих машинное обучение

Fig. 10. Ratio of the total number of publications and those involving machine learning

Ограничения и рекомендации. К ограничениям полученного корпуса релевантных документов возможно отнести:

- небольшое количество отечественных публикаций;
- исключение из рассмотрения метрик, не исследующих исходный код приложения;
- фокусирование на метриках кода программ, предполагающих исключительно статический анализ приложений.

В последующих систематических обзорах литературы имеет смысл расширить рассматриваемую область с учетом выделенных ограничений. Также необходимо сделать акцент на развивающихся направлениях в данной области: комбинировании новых методов с расчетом метрик кода, применении машинного обучения и паттернов программирования для отдельных функций; предсказании появления и устранения ошибок по результатам расчета метрик, нахождении недостатков исходного кода с помощью «запахов кода».

Выводы. Целью исследования стал анализ методов, подходов и средств применения метрик для анализа программ. Выполнение систематического картографирования литературы доказало актуальность данной области для научного сообщества в настоящее время и показало основные направления деятельности на данный момент.

В рамках методологии SMS была разработана стратегия поиска научных публикаций и определены их источники, а также ключевые слова, ис-

пользуемые в исследуемой области. Были сформулированы запросы к выделенным источникам данных для обеспечения возможности повтора исследования. Из числа найденных публикаций были отобраны релевантные по описанным критериям включения-исключения, что позволило сократить число публикаций в выборке с 295 до 49, соответствующих выбранной тематике.

Полученные релевантные публикации были классифицированы для наблюдения ландшафта исследований. Были построены частотные словари и диаграммы контекстного анализа и распределений публикаций по различным признакам: типу документа, географическому принципу, количеству авторов. Также были выделены организации и университеты, занимающиеся темой метрик.

Также проведено исследование терминологии предметной области, метрик и наборов метрик относительно решаемых задач, анализируемых языков программирования и современных вариантов сочетания других подходов с расчетом метрик кода, самое востребованное из которых – машинное обучение. Для данных аспектов работы были построены соответствующие диаграммы. Предложена классификационная схема из 9 категорий для задач исследований с помощью метрик и их наборов.

Были даны ответы на вопросы о ландшафте исследований по тематике, существующих методах и подходах, а также о разработанных программных решениях. Для полученного корпуса сформулированы ограничения и рекомендации.

В результате нашего SMS выявлена тенденция снижения за период 2017–2023 гг. количества релевантных работ, исследующих методы и средства расчета и применения метрик кода программных продуктов. Наибольшую публикационную активность в указанном домене проявили сотрудники университетов NIT (Индия) и University of Coimbra (Португалия).

На основе предложенной в этой статье классификации определены публикации с соответствующими задачами исследований с помощью метрик.

Для каждой из категорий предложенной здесь классификации задач исследования с помощью метрик определены наиболее часто используемые метрики и их наборы, а также частота их применения.

Обнаружены закономерности между используемыми метриками и языком программирования анализируемого кода.

Сочетание подхода расчета метрик кода с применением технологий машинного обучения чаще всего находит применение при решении задачи обнаружения уязвимостей кода.

С помощью нашего SMS определены Топ-3 языков программирования, для которых оцениваются характеристики кода: Java, C++ и С.

Для измерения и анализа характеристик, показателей кода на одном языке программирования или нескольких близких (до трех) разработаны программные средства. С их помощью решаются как традиционные задачи оценки качества программных продуктов, так и другие – прогноз их безопасности и надежности.

Список литературы

1. Tahir A., MacDonell S. G. A systematic mapping study on dynamic metrics and software quality // 28th IEEE Int. Conf. on Software Maintenance (ICSM2012). Trento, Italy: IEEE, 2012. P. 326–335. doi: 10.1109/ICSM.2012.6405289.
2. Bandi A., Williams B. J., Allen E. B. Empirical evidence of code decay: A systematic mapping study // 20th Working Conf. on Reverse Engin. (WCRE). Koblenz, Germany: IEEE, 2013. P. 341–350. doi: 10.1109/WCRE.2013.6671309.
3. Source code metrics: A systematic mapping study / A. S. Nunez-Varela, H. G. Pérez-Gonzalez, F. E. Martínez-Perez, C. Soubervielle-Montalvo // J. of Systems and Software. 2017. Vol. 128. P. 164–197. doi: 10.1016/j.jss.2017.03.044.
4. Sayago-Heredia J., Perez-Castillo R., Piattini M. A systematic mapping study on analysis of code repositories // Informatica. Vol. 32, iss. 3. 2021. P. 619–660. doi: 10.15388/21-INFOR454.
5. NurColakoglu F., Yazici A., Mishra A. Software product quality metrics: a systematic mapping study // IEEE Access. 2021. Vol. 9. P. 44647–44670. doi: 10.1109/ACCESS.2021.3054730.
6. Machine learning-based exploration of the impact of move method refactoring on object-oriented software quality attributes / J. Al Dalla, H. Abdulsalam, M. AlMarzouq, A. Selamat // Arabian J. for Sci. and Engin. 2023. Vol. 49. P. 3867–3885. doi: 10.1007/s13369-023-08174-0.
7. Al-Obeidallah M. G. Towards a framework to assess the impact of design patterns on software metrics // 2023 Int. Conf. on Multimedia Comp., Networking and Appl. (MCNA). Valencia, Spain: IEEE, 2023. P. 67–72. doi: 10.1109/MCNA59361.2023.10185865.
8. Amara D., Ben Arfa Rabai L. Towards a new framework of software reliability measurement based on software metrics // Procedia Comp. Sci. 2017. Vol. 109. P. 725–730. doi: 10.1016/j.procs.2017.05.428.
9. Bhandari G. P., Gupta R. Machine learning based software fault prediction utilizing source code metrics // 2018 IEEE 3rd Int. Conf. on Comp., Communication and Security (ICCCS). Kathmandu, Nepal: IEEE, 2018. P. 40–45. doi: 10.1109/CCCS.2018.8586805.
10. Bhandari G. P., Gupta R. Measuring the fault predictability of software using deep learning techniques with software metrics // 2018 5th IEEE Uttar Pradesh Section Int. Conf. on Electr., Electronics and Comp. Engin. (UPCON). Gorakhpur, India: IEEE, 2018. P. 1–6. doi: 10.1109/UPCON.2018.8597154.
11. The usefulness of software metric thresholds for detection of bad smells and fault prediction / M. A. S. Bigonha, K. Ferreira, P. Souza, B. L. Sousa, M. Januário, D. Lima // Information and Software Technol. 2019. Vol. 115(9). P. 79–92. doi: 10.1016/j.infsof.2019.08.005.
12. Burakov V. V., Borovkov A. I. Advanced metric analysis tool for Java source code // Information and Control Systems. 2023. № 1 (122). P. 17–28. doi: 10.31799/1684-8853-2023-1-17-28.
13. Chong T.-Y., Anu V., Sultana K. Z. Using software metrics for predicting vulnerable code-components: A study on Java and Python open source projects // 2019 IEEE Int. Conf. on Computational Sci. and Engin. (CSE) and IEEE Int. Conf. on Embedded and Ubiquitous Comp. (EUC). New York, NY, USA: IEEE, 2019. P. 98–103. doi: 10.1109/CSE/EUC.2019.00028.
14. An experimental search-based approach to cohesion metric evaluation / M. Ó. Cinnéide, I. Hemati Moghadam, M. Harman, S. Counsell, L. Tratt // Empirical Software Engin. 2017. Vol. 22(1). P. 292–329. doi: 10.1007/s10664-016-9427-7.
15. Measuring developers' contribution in source code using quality metrics / P. R. de Bassi, G. M. P. Wanderley, P. H. Banali, E. C. Paraiso // 2018 IEEE 22nd Int. Conf. on Comp. Supported Cooperative Work in Design (CSCWD). Nanjing, China: IEEE, 2018. P. 39–44. doi: 10.1109/CSCWD.2018.8465320.
16. Dósea M., Sant'Anna C., da Silva B. C. How do design decisions affect the distribution of software metrics? // 2018 IEEE/ACM 26th Int. Conf. on Program Comprehension (ICPC). Gothenburg, Sweden: IEEE, 2018. P. 74–85. doi: 10.1145/3196321.3196337.
17. Using source code metrics for predicting metamorphic relations at method level / A. Duque-Torres, D. Pfahl, C. Klammer, S. Fischer // 2022 IEEE Int. Conf. on Software Analysis, Evolution and Reengin. (SANER). Honolulu, HI, USA: IEEE, 2022. P. 1147–1154. doi: 10.1109/SANER53432.2022.00132.
18. El-Sharkawy S., Krafczyk A., Schmid K. Fast static analyses of software product lines: An example with more than 42.000 metrics // 14th Int. Working Conf. on Variability Modelling of Software-Intensive Systems (VaMoS). Magdeburg, Germany: ACM, 2020. Art. 8. P. 1–9. doi: 10.1145/3377024.3377031.
19. Ganesh S., Ohlsson T., Palma F. Predicting security vulnerabilities using source code metrics // 2021 9th Swedish Workshop on Data Sci. (SweDS). Växjö, Sweden: IEEE, 2021. P. 1–7. doi: 10.1109/SweDS53855.2021.9638301.

20. Joon A., Kumar Tyagi R., Kumar K. Noise filtering and imbalance class distribution removal for optimizing software fault prediction using best software metrics suite // 2020 5th Int. Conf. on Communication and Electronics Systems (ICCES). Coimbatore, India: IEEE, 2020. P. 1381–1389. doi: 10.1109/ICCES48766.2020.9137899.
21. Komalasari A., Candra M. Z. C. Improving defect prediction using combination of software metrics // 2022 Int. Conf. on Data and Software Engin. (ICoDSE). Denpasar, Indonesia: IEEE, 2022. P. 89–94. doi: 10.1109/ICoDSE56892.2022.9971813.
22. Kuk K., Milić P., Denić S. Object-oriented software metrics in software code vulnerability analysis // 2020 Int. Conf. on INnovations in Intelligent SysTems and Appl. (INISTA). Novi Sad, Serbia: IEEE, 2020. P. 1–6. doi: 10.1109/INISTA49547.2020.9194645.
23. Kumar L., Misra S., Rath S. Ku. An empirical analysis of the effectiveness of software metrics and fault prediction model for identifying faulty classes // *Comp. Standards & Interfaces*. 2017. Vol. 53. P. 1–32. doi: 10.1016/j.csi.2017.02.003.
24. Kumar L., Rath S. K., Sureka A. Using source code metrics and multivariate adaptive regression splines to predict maintainability of service oriented software // 2017 IEEE 18th Int. Symp. on High Assurance Systems Engin. (HASE). Singapore: IEEE, 2017. P. 88–95. doi: 10.1109/HASE.2017.11.
25. Kumar L., Rath S. K., Sureka A. Using source code metrics to predict change-prone web services: A case-study on ebay services // 2017 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE). Klagenfurt, Austria: IEEE, 2017. P. 1–7. doi: 10.1109/MALTESQUE.2017.7882009.
26. Kumar L., Sureka A. Analyzing fault prediction usefulness from cost perspective using source code metrics // 2017 Tenth Int. Conf. on Contemporary Computing (IC3). Noida, India: IEEE, 2017. P. 1–7. doi: 10.1109/IC3.2017.8284297.
27. Kunkel J., Dolz M. F. Understanding hardware and software metrics with respect to power consumption // *Sustainable Comp.: Informatics and Systems*. 2018. Vol. 17. P. 43–54. doi: 10.1016/j.suscom.2017.10.016.
28. Lenhard J., Blom M., Herold S. Exploring the suitability of source code metrics for indicating architectural inconsistencies // *Software Quality J.* 2019. Vol. 27. P. 241–274. doi: 10.1007/s11219-018-9404-z.
29. Liu Y., Li Y., Guo J., Zhou Y., Xu B. Connecting software metrics across versions to predict defects // 2018 IEEE 25th Int. Conf. on Software Analysis, Evolution and Reengin. (SANER). Campobasso, Italy: IEEE, 2018. P. 232–243. doi: 10.1109/SANER.2018.8330212.
30. Mamun M. A. A., Berger C., Hansson J. A. Correlations of software code metrics: An empirical study // 27th Int. Workshop on Software Measurement and 12th Int. Conf. on Software Process and Product Measurement. Gothenburg, Sweden: ACM, 2017. P. 255–266. doi: 10.1145/3143434.3143445.
31. Mamun M. A. A., Berger C., Hansson J. Effects of measurements on correlations of software code metrics // *Empirical Software Engin.* 2019. Vol. 24, no. 4. P. 2764–2818. doi: 10.1007/s10664-019-09714-9.
32. Martinez-Ortiz A. -L., Lizcano D., Ortega M. Software metrics artifacts making Web Quality Measurable // 2019 IEEE/ACM 14th Int. Workshop on Automation of Software Test (AST), Montreal, QC, Canada: IEEE, 2019. P. 1–6. doi: 10.1109/AST.2019.000-2.
33. Medeiros N., Ivaki N., Costa P., Vieira M. Software metrics as indicators of security vulnerabilities // 2017 IEEE 28th Int. Symp. on Software Reliability Engin. (ISSRE). Toulouse, France: IEEE, 2017. P. 216–227. doi: 10.1109/ISSRE.2017.11.
34. An approach for trustworthiness benchmarking using software metrics / N. Medeiros, N. Ivaki, P. Costa, M. Vieira // 2018 IEEE 23rd Pacific Rim Int. Symp. on Dependable Comp. (PRDC). Taipei, Taiwan: IEEE, 2018. P. 84–93. doi: 10.1109/PRDC.2018.00019.
35. Vulnerable code detection using software metrics and machine learning / N. Medeiros, N. Ivaki, P. Costa, M. Vieira // *IEEE Access*. 2020. Vol. 8. P. 219174–219198. doi: 10.1109/ACCESS.2020.3041181.
36. An empirical study on software metrics and machine learning to identify untrustworthy code / N. Medeiros, N. Ivaki, P. Costa, M. Vieira // 2021 17th European Dependable Comp. Conf. (EDCC). Munich, Germany: IEEE, 2021. P. 87–94. doi: 10.1109/EDCC53658.2021.00020.
37. Mhawish M. Y., Gupta M. Software metrics and tree-based machine learning algorithms for distinguishing and detecting similar structure design patterns // *SN Appl. Sci.* 2019. Vol. 2. Art. 11. P. 1–10. doi: 10.1007/s42452-019-1815-3.
38. Mshelia Y. U., Apeh S. T., Edoghogho O. A comparative assessment of software metrics tools // 2017 Int. Conf. on Comp. Networking and Informatics (ICCNi). Lagos, Nigeria: IEEE, 2017. P. 1–9. doi: 10.1109/ICCNi.2017.8123809.
39. On the identification of design problems in stinky code: experiences and tool support / W. Oizumi, L. Sousa, A. Oliveira, A. Garcia, A. B. Agbachi, R. Oliveira, C. Lucena // *J. of the Brazilian Comp. Soc.* 2018. Vol. 24. Art. 13. P. 1–30. doi: 10.1186/s13173-018-0078-y.
40. Ozawa M., Dohi T., Okamura H. How do software metrics affect test case prioritization? // 2018 IEEE 42nd Ann. Comp. Software and Appl. Conf. (COMPSAC). Tokyo, Japan: IEEE, 2018. P. 245–250. doi: 10.1109/COMPSAC.2018.00038.
41. On the evaluation of code smells and detection tools / T. Paiva, A. Damasceno, E. Figueiredo, C. Sant'Anna // *J. of Software Engin. Research and Development*. 2017. Vol. 5. Art. 7. P. 1–28. doi: 10.1186/s40411-017-0041-1.
42. Comparing heuristic and machine learning approaches for metric-based code smell detection / F. Pecorelli, F. Palomba, D. Di Nucci, A. De Lucia // 2019 IEEE/ACM 27th Int. Conf. on Program Comprehension (ICPC), Montreal, QC, Canada: IEEE, 2019. P. 93–104. doi: 10.1109/ICPC.2019.00023.

43. Pereira J. D'A., Campos J. R., Vieira M. Machine learning to combine static analysis alerts with software metrics to detect security vulnerabilities: An empirical study // 2021 17th European Dependable Comp. Conf. (EDCC). Munich, Germany: IEEE, 2021. P. 1–8. doi: 10.1109/EDCC53658.2021.00008.
44. Controlling the attack surface of object-oriented refactorings / S. Ruland, G. Kulcsár, E. Leblebici, S. Peldszus, M. Lochau // Int. Conf. on Fundamental Approaches to Software Engin. FASE 2018. Thessaloniki, Greece: Springer, Cham., 2018. Vol. 10802. P. 38–55. doi: 10.1007/978-3-319-89363-1_3.
45. Warnars H. L. H. S., Gaol F. L., Randriatoamana R. Object oriented metrics to measure the quality of software upon PHP source code with PHP_depend study case request online system application // 2017 Int. Conf. on Appl. Comp. and Communication Technol. (ComCom). Jakarta, Indonesia: IEEE, 2017. P. 1–5. doi: 10.1109/COMCOM.2017.8167097.
46. Siavvas M., Kehagias D., Tzovaras D. A Preliminary study on the relationship among software metrics and specific vulnerability types // 2017 Int. Conf. on Computational Sci. and Computational Intelligence (CSCI). Las Vegas, NV, USA: IEEE, 2017. P. 916–921. doi: 10.1109/CSCI.2017.159.
47. Sultana K. Z., Williams B. J. Evaluating micro patterns and software metrics in vulnerability prediction // 2017 6th Int. Workshop on Software Mining (Software-Mining). Urbana, IL, USA: IEEE, 2017. P. 40–47. doi: 10.1109/SOFTWAREMINING.2017.8100852.
48. Sultana K. Z., Williams B. J., Bosu A. A Comparison of nano-patterns vs. Software metrics in vulnerability prediction // 2018 25th Asia-Pacific Software Engin. Conf. (APSEC). Nara, Japan: IEEE, 2018. P. 355–364. doi: 10.1109/APSEC.2018.00050.
49. Prediction of quality of service parameters using aggregate software metrics and machine learning techniques / M. K. Tripathi, D. Chaubisa, L. Kumar, L. B. Murthy Neti // 2018 15th IEEE India Council Int. Conf. (INDICON). Coimbatore, India: IEEE, 2018. P. 1–6. doi: 10.1109/INDICON45594.2018.8986987.
50. Tummalapalli S., Kumar L., Bhanu Murthy N. L. Prediction of Web service anti-patterns using aggregate software metrics and machine learning techniques // 13th Innovations in Software Engin. Conf. (ISEC 2020). Jabalpur, India, 2020. Art. 8. P. 1–11. doi: 10.1145/3385032.3385042.
51. Weighted software metrics aggregation and its application to defect prediction / M. Ulan, W. Löwe, M. Ericsson, A. Wingkvist // Empirical Software Engin. 2021. Vol. 26(5). Art. 86. P. 86–120. doi: 10.1007/s10664-021-09984-2.
52. Velec I., Pietriková E. Metrics for Student Source Code Analysis // 2020 18th Int. Conf. on Emerging eLearning Technol. and Appl. (ICETA). Košice, Slovenia: IEEE, 2020. P. 739–744. doi: 10.1109/ICETA51985.2020.9379264.
53. Vytovtov P., Markov E. Source code quality classification based on software metrics // 2017 20th Conf. of Open Innovations Association (FRUCT). St. Petersburg, Russia: IEEE, 2017. P. 505–511. doi: 10.23919/FRUCT.2017.8071355.
54. Zhou A., Sultana K. Z., Samanthula B. K. Investigating the changes in software metrics after vulnerability is fixed // 2021 IEEE Int. Conf. on Big Data (Big Data). Orlando, FL, USA: IEEE, 2021. P. 5658–5663. doi: 10.1109/BigData52589.2021.9671334.
55. Kitchenham B., Charters S. Guidelines for performing systematic literature reviews in software engineering. Ver. 2.3. Technical Report EBSE 2007-001. Keele University, Durham University, UK, 2007. URL: http://legacyfileshare.elsevier.com/promis_misc/525444systematicreviewsguide.pdf (дата обращения: 02.03.2024).
56. Звездин С. В. Метрики программного кода (обзор) // Информационные системы и технологии. 2009. № 6(56). С. 5–10.
57. Бурдыко Т. Г., Бушмелева К. И. Метрики как средство управления качеством программного обеспечения // Инновационные, информационные и коммуникационные технологии. 2018. № 1. С. 38–41.
58. Дорофеева О. С., Казаков Б. В., Казакова И. А. Метрики качества программного обеспечения // XII Межд. науч.-техн. конф. «Новые информационные технологии и системы» (НИТИС-2015). Пенза: изд-во ПГУ, 2015. С. 109–111.

Информация об авторе

Корзников Артем Олегович – студент магистратуры ПГНИУ. Пермский государственный национальный исследовательский университет, ул. Букирева, д. 15, Пермь, 614068, Россия.

E-mail: artemkorz@mail.ru

<http://orcid.org/0009-0006-3941-9214>

Дацин Наталья Николаевна – канд. физ.-мат. наук, доцент кафедры математического обеспечения вычислительных систем ПГНИУ. Пермский государственный национальный исследовательский университет, ул. Букирева, д. 15, Пермь, 614068, Россия.

E-mail: nndatsun@inbox.ru

<http://orcid.org/0000-0001-8560-7036>

References

1. Tahir A., MacDonell S. G. A systematic mapping study on dynamic metrics and software quality // 28th IEEE Int. Conf. on Software Maintenance (ICSM2012). Trento, Italy: IEEE, 2012. P. 326–335. doi: 10.1109/ICSM.2012.6405289.
2. Bandi A., Williams B. J., Allen E. B. Empirical evidence of code decay: A systematic mapping study // 20th Working Conf. on Reverse Engin. (WCRE). Koblenz, Germany: IEEE, 2013. P. 341–350. doi: 10.1109/WCRE.2013.6671309.
3. Source code metrics: A systematic mapping study / A. S. Nunez-Varela, H. G. Pérez-Gonzalez, F. E. Martínez-Perez, C. Soubervielle-Montalvo // J. of Systems and Software. 2017. Vol. 128. P. 164–197. doi: 10.1016/j.jss.2017.03.044.
4. Sayago-Heredia J., Perez-Castillo R., Piattini M. A systematic mapping study on analysis of code repositories // Informatica. Vol. 32, iss. 3. 2021. P. 619–660. doi: 10.15388/21-INFOR454.
5. NurColakoglu F., Yazici A., Mishra A. Software product quality metrics: a systematic mapping study // IEEE Access. 2021. Vol. 9. P. 44647–44670. doi: 10.1109/ACCESS.2021.3054730.
6. Machine learning-based exploration of the impact of move method refactoring on object-oriented software quality attributes / J. Al Dalla, H. Abdulsalam, M. AlMarzouq, A. Selamat // Arabian J. for Sci. and Engin. 2023. Vol. 49. P. 3867–3885. doi: 10.1007/s13369-023-08174-0.
7. Al-Obeidallah M. G. Towards a framework to assess the impact of design patterns on software metrics // 2023 Int. Conf. on Multimedia Comp., Networking and Appl. (MCNA). Valencia, Spain: IEEE, 2023. P. 67–72. doi: 10.1109/MCNA59361.2023.10185865.
8. Amara D., Ben Arfa Rabai L. Towards a new framework of software reliability measurement based on software metrics // Procedia Comp. Sci. 2017. Vol. 109. P. 725–730. doi: 10.1016/j.procs.2017.05.428.
9. Bhandari G. P., Gupta R. Machine learning based software fault prediction utilizing source code metrics // 2018 IEEE 3rd Int. Conf. on Comp., Communication and Security (ICCCS). Kathmandu, Nepal: IEEE, 2018. P. 40–45. doi: 10.1109/ICCCS.2018.8586805.
10. Bhandari G. P., Gupta R. Measuring the fault predictability of software using deep learning techniques with software metrics // 2018 5th IEEE Uttar Pradesh Section Int. Conf. on Electr., Electronics and Comp. Engin. (UPCON). Gorakhpur, India: IEEE, 2018. P. 1–6. doi: 10.1109/UPCON.2018.8597154.
11. The usefulness of software metric thresholds for detection of bad smells and fault prediction / M. A. S. Bigonha, K. Ferreira, P. Souza, B. L. Sousa, M. Januário, D. Lima // Information and Software Technol. 2019. Vol. 115(9). P. 79–92. doi: 10.1016/j.infsof.2019.08.005.
12. Burakov V. V., Borovkov A. I. Advanced metric analysis tool for Java source code // Information and Control Systems. 2023. № 1 (122). P. 17–28. doi: 10.31799/1684-8853-2023-1-17-28.
13. Chong T.-Y., Anu V., Sultana K. Z. Using software metrics for predicting vulnerable code-components: A study on Java and Python open source projects // 2019 IEEE Int. Conf. on Computational Sci. and Engin. (CSE) and IEEE Int. Conf. on Embedded and Ubiquitous Comp. (EUC). New York, NY, USA: IEEE, 2019. P. 98–103. doi: 10.1109/CSE/EUC.2019.00028.
14. An experimental search-based approach to cohesion metric evaluation / M. Ó. Cinnéide, I. Hemati Moghadam, M. Harman, S. Counsell, L. Tratt // Empirical Software Engin. 2017. Vol. 22(1). P. 292–329. doi: 10.1007/s10664-016-9427-7.
15. Measuring developers' contribution in source code using quality metrics / P. R. de Bassi, G. M. P. Wanderley, P. H. Banali, E. C. Paraiso // 2018 IEEE 22nd Int. Conf. on Comp. Supported Cooperative Work in Design (CSCWD). Nanjing, China: IEEE, 2018. P. 39–44. doi: 10.1109/CSCWD.2018.8465320.
16. Dósea M., Sant'Anna C., da Silva B. C. How do design decisions affect the distribution of software metrics? // 2018 IEEE/ACM 26th Int. Conf. on Program Comprehension (ICPC). Gothenburg, Sweden: IEEE, 2018. P. 74–85. doi: 10.1145/3196321.3196337.
17. Using source code metrics for predicting metamorphic relations at method level / A. Duque-Torres, D. Pfahl, C. Klammer, S. Fischer // 2022 IEEE Int. Conf. on Software Analysis, Evolution and Reengin. (SANER). Honolulu, HI, USA: IEEE, 2022. P. 1147–1154. doi: 10.1109/SANER53432.2022.00132.
18. El-Sharkawy S., Krafczyk A., Schmid K. Fast static analyses of software product lines: An example with more than 42.000 metrics // 14th Int. Working Conf. on Variability Modelling of Software-Intensive Systems (VaMoS). Magdeburg, Germany: ACM, 2020. Art. 8. P. 1–9. doi: 10.1145/3377024.3377031.
19. Ganesh S., Ohlsson T., Palma F. Predicting security vulnerabilities using source code metrics // 2021 9th Swedish Workshop on Data Sci. (SweDS). Växjö, Sweden: IEEE, 2021. P. 1–7. doi: 10.1109/SweDS53855.2021.9638301.
20. Joon A., Kumar Tyagi R., Kumar K. Noise filtering and imbalance class distribution removal for optimizing software fault prediction using best software metrics suite // 2020 5th Int. Conf. on Communication and Electronics Systems (ICES). Coimbatore, India: IEEE, 2020. P. 1381–1389. doi: 10.1109/ICES48766.2020.9137899.
21. Komalasari A., Candra M. Z. C. Improving defect prediction using combination of software metrics // 2022 Int. Conf. on Data and Software Engin. (ICoDSE). Denpasar, Indonesia: IEEE, 2022. P. 89–94. doi: 10.1109/ICoDSE56892.2022.9971813.
22. Kuk K., Milić P., Denić S. Object-oriented software metrics in software code vulnerability analysis // 2020 Int. Conf. on INnovations in Intelligent SysTems and Appl. (INISTA). Novi Sad, Serbia: IEEE, 2020. P. 1–6. doi: 10.1109/INISTA49547.2020.9194645.
23. Kumar L., Misra S., Rath S. Ku. An empirical analysis of the effectiveness of software metrics and fault

prediction model for identifying faulty classes // *Comp. Standards & Interfaces*. 2017. Vol. 53. P. 1–32. doi: 10.1016/j.csi.2017.02.003.

24. Kumar L., Rath S. K., Sureka A. Using source code metrics and multivariate adaptive regression splines to predict maintainability of service oriented software // 2017 IEEE 18th Int. Symp. on High Assurance Systems Engin. (HASE). Singapore: IEEE, 2017. P. 88–95. doi: 10.1109/HASE.2017.11.

25. Kumar L., Rath S. K., Sureka A. Using source code metrics to predict change-prone web services: A case-study on ebay services // 2017 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTesQuE). Klagenfurt, Austria: IEEE, 2017. P. 1–7. doi: 10.1109/MALTESQUE.2017.7882009.

26. Kumar L., Sureka A. Analyzing fault prediction usefulness from cost perspective using source code metrics // 2017 Tenth Int. Conf. on Contemporary Computing (IC3). Noida, India: IEEE, 2017. P. 1–7. doi: 10.1109/IC3.2017.8284297.

27. Kunkel J., Dolz M. F. Understanding hardware and software metrics with respect to power consumption // *Sustainable Comp.: Informatics and Systems*. 2018. Vol. 17. P. 43–54. doi: 10.1016/j.suscom.2017.10.016.

28. Lenhard J., Blom M., Herold S. Exploring the suitability of source code metrics for indicating architectural inconsistencies // *Software Quality J.* 2019. Vol. 27. P. 241–274. doi: 10.1007/s11219-018-9404-z.

29. Liu Y., Li Y., Guo J., Zhou Y., Xu B. Connecting software metrics across versions to predict defects // 2018 IEEE 25th Int. Conf. on Software Analysis, Evolution and Reengin. (SANER). Campobasso, Italy: IEEE, 2018. P. 232–243. doi: 10.1109/SANER.2018.8330212.

30. Mamun M. A. A., Berger C., Hansson J. A. Correlations of Software Code Metrics: An Empirical Study // 27th Int. Workshop on Software Measurement and 12th Int. Conf. on Software Process and Product Measurement. Gothenburg, Sweden: ACM, 2017. P. 255–266. doi: 10.1145/3143434.3143445.

31. Mamun M. A. A., Berger C., Hansson J. Effects of measurements on correlations of software code metrics // *Empirical Software Engin.* 2019. Vol. 24, no. 4. P. 2764–2818. doi: 10.1007/s10664-019-09714-9.

32. Martinez-Ortiz A. -L., Lizcano D., Ortega M. Software metrics artifacts making Web Quality Measurable // 2019 IEEE/ACM 14th Int. Workshop on Automation of Software Test (AST), Montreal, QC, Canada: IEEE, 2019. P. 1–6. doi: 10.1109/AST.2019.000-2.

33. Medeiros N., Ivaki N., Costa P., Vieira M. Software Metrics as Indicators of Security Vulnerabilities // 2017 IEEE 28th Int. Symp. on Software Reliability Engin. (ISSRE). Toulouse, France: IEEE, 2017. P. 216–227. doi: 10.1109/ISSRE.2017.11.

34. An approach for trustworthiness benchmarking using software metrics / N. Medeiros, N. Ivaki, P. Costa, M. Vieira // 2018 IEEE 23rd Pacific Rim Int. Symp. on Dependable Comp. (PRDC). Taipei, Taiwan: IEEE, 2018. P. 84–93. doi: 10.1109/PRDC.2018.00019.

35. Vulnerable code detection using software metrics and machine learning / N. Medeiros, N. Ivaki, P. Costa, M. Vieira // *IEEE Access*. 2020. Vol. 8. P. 219174–219198. doi: 10.1109/ACCESS.2020.3041181.

36. An empirical study on software metrics and machine learning to identify untrustworthy code / N. Medeiros, N. Ivaki, P. Costa, M. Vieira // 2021 17th European Dependable Comp. Conf. (EDCC). Munich, Germany: IEEE, 2021. P. 87–94. doi: 10.1109/EDCC53658.2021.00020.

37. Mhawish M. Y., Gupta M. Software Metrics and tree-based machine learning algorithms for distinguishing and detecting similar structure design patterns // *SN Appl. Sci.* 2019. Vol. 2. Art. 11. P. 1–10. doi: 10.1007/s42452-019-1815-3.

38. Mshelia Y. U., Apeh S. T., Edoghogho O. A comparative assessment of software metrics tools // 2017 Int. Conf. on Comp. Networking and Informatics (ICCNi). Lagos, Nigeria: IEEE, 2017. P. 1–9. doi: 10.1109/ICCNi.2017.8123809.

39. On the identification of design problems in stinky code: experiences and tool support / W. Oizumi, L. Sousa, A. Oliveira, A. Garcia, A. B. Agbachi, R. Oliveira, C. Lucena // *J. of the Brazilian Comp. Soc.* 2018. Vol. 24. Art. 13. P. 1–30. doi: 10.1186/s13173-018-0078-y.

40. Ozawa M., Dohi T., Okamura H. How do software metrics affect test case prioritization? // 2018 IEEE 42nd Ann. Comp. Software and Appl. Conf. (COMPSAC). Tokyo, Japan: IEEE, 2018. P. 245–250. doi: 10.1109/COMPSAC.2018.00038.

41. On the evaluation of code smells and detection tools / T. Paiva, A. Damasceno, E. Figueiredo, C. Sant'Anna // *J. of Software Engin. Research and Development*. 2017. Vol. 5. Art. 7. P. 1–28. doi: 10.1186/s40411-017-0041-1.

42. Comparing heuristic and machine learning approaches for metric-based code smell detection / F. Pecorelli, F. Palomba, D. Di Nucci, A. De Lucia // 2019 IEEE/ACM 27th Int. Conf. on Program Comprehension (ICPC), Montreal, QC, Canada: IEEE, 2019. P. 93–104. doi: 10.1109/ICPC.2019.00023.

43. Pereira J. D'A., Campos J. R., Vieira M. Machine Learning to combine static analysis alerts with software metrics to detect security vulnerabilities: An empirical study // 2021 17th European Dependable Comp. Conf. (EDCC). Munich, Germany: IEEE, 2021. P. 1–8. doi: 10.1109/EDCC53658.2021.00008.

44. Controlling the attack surface of object-oriented refactorings / S. Ruland, G. Kulcsár, E. Leblebici, S. Peldszus, M. Lochau // *Int. Conf. on Fundamental Approaches to Software Engin. FASE 2018*. Thessaloniki, Greece: Springer, Cham., 2018. Vol. 10802. P. 38–55. doi: 10.1007/978-3-319-89363-1_3.

45. Warnars H. L. H. S., Gaol F. L., Randriatoamana R. Object oriented metrics to measure the quality of software upon PHP source code with PHP_depend study case request online system application // 2017 Int. Conf. on Appl. Comp. and Communication Technol. (ComCom). Jakarta, Indonesia: IEEE, 2017. P. 1–5. doi: 10.1109/COMCOM.2017.8167097.

46. Siavvas M., Kehagias D., Tzovaras D. A Preliminary study on the relationship among software metrics and specific vulnerability types // 2017 Int. Conf. on Computational Sci. and Computational Intelligence (CSCI). Las Vegas, NV, USA: IEEE, 2017. P. 916–921. doi: 10.1109/CSCI.2017.159.
47. Sultana K. Z., Williams B. J. Evaluating micro patterns and software metrics in vulnerability prediction // 2017 6th Int. Workshop on Software Mining (Software-Mining). Urbana, IL, USA: IEEE, 2017. P. 40–47. doi: 10.1109/SOFTWAREMINING.2017.8100852.
48. Sultana K. Z., Williams B. J., Bosu A. A Comparison of nano-patterns vs. Software metrics in vulnerability prediction // 2018 25th Asia-Pacific Software Engin. Conf. (APSEC). Nara, Japan: IEEE, 2018. P. 355–364. doi: 10.1109/APSEC.2018.00050.
49. Prediction of quality of service parameters using aggregate software metrics and machine learning techniques / M. K. Tripathi, D. Chaubisa, L. Kumar, L. B. Murthy Neti // 2018 15th IEEE India Council Int. Conf. (INDICON). Coimbatore, India: IEEE, 2018. P. 1–6. doi: 10.1109/INDI CON45594.2018.8986987.
50. Tummalapalli S., Kumar L., Bhanu Murthy N. L. Prediction of Web service anti-patterns using aggregate software metrics and machine learning techniques // 13th Innovations in Software Engin. Conf. (ISEC 2020). Jabalpur, India, 2020. Art. 8. P. 1–11. doi: 10.1145/3385032.3385042.
51. Weighted software metrics aggregation and its application to defect prediction / M. Ulan, W. Löwe, M. Ericsson, A. Wingkvist // Empirical Software Engin. 2021. Vol. 26(5). Art. 86. P. 86–120. doi: 10.1007/s10664-021-09984-2.
52. Velec I., Pietriková E. Metrics for Student Source Code Analysis // 2020 18th Int. Conf. on Emerging eLearning Technol. and Appl. (ICETA). Košice, Slovenia: IEEE, 2020. P. 739–744. doi: 10.1109/ICETA51985.2020.9379264.
53. Vytovtov P., Markov E. Source code quality classification based on software metrics // 2017 20th Conf. of Open Innovations Association (FRUCT). St. Petersburg, Russia: IEEE, 2017. P. 505–511. doi: 10.23919/FRUCT.2017.8071355.
54. Zhou A., Sultana K. Z., Samanthula B. K. Investigating the changes in software metrics after vulnerability is fixed // 2021 IEEE Int. Conf. on Big Data (Big Data). Orlando, FL, USA: IEEE, 2021. P. 5658–5663. doi: 10.1109/BigData52589.2021.9671334.
55. Kitchenham B., Charters S. Guidelines for performing systematic literature reviews in software engineering. Ver. 2.3. Technical Report EBSE 2007-001. Keele University, Durham University, UK, 2007. URL: http://legacyfileshare.elsevier.com/promis_misc/525444systematicreviewsguide.pdf (data obrashhenija: 02.03.2024).
56. Zvezdin S. V. Metriki programmnogo koda (obzor) // Informacionnye sistemy i tehnologii. 2009. № 6(56). S. 5–10. (In Russ.).
57. Burdyko T. G., Bushmeleva K. I. Metriki kak sredstvo upravlenija kachestvom programmogo obespechenija // Innovacionnye, informacionnye i kommunikacionnye tehnologii. 2018. № 1. S. 38–41. (In Russ.).
58. Dorofeeva O. S., Kazakov B. V., Kazakova I. A. Metriki kachestva programmogo obespechenija // XII Mezhd. nauch.-tehn. konf. «Novye informacionnye tehnologii i sistemy» (NITIS-2015). Penza: izd-vo PGU, 2015. S. 109–111. (In Russ.).
-

Information about the author

Artem O. Korznikov – masters student of Perm State University, Bukireva St., 15, Perm, 614068, Russia.
E-mail: artemkorz@mail.ru
<http://orcid.org/0009-0006-3941-9214>

Natalia N. Datsun – Cand. Sci. (Phys. and Math.), Assistant Professor of Perm State University, Bukireva St., 15, Perm, 614068, Russia.
E-mail: nndatsun@inbox.ru
<http://orcid.org/0000-0001-8560-7036>

Статья поступила в редакцию 25.04.2024; принята к публикации после рецензирования 26.06.2024; опубликована онлайн 24.10.2024.

Submitted 25.04.2024; accepted 26.06.2024; published online 24.10.2024.
