МЕТОДЫ ПРОГНОЗИРОВАНИЯ ДЕФЕКТОВ В ПРОГРАММНЫХ ПРОДУКТАХ НА ОСНОВЕ РЕТРОСПЕКТИВНОЙ И ТЕКУЩЕЙ ИНФОРМАЦИИ

DOI: 10.36724/2072-8735-2026-20-1-52-59

Леохин Юрий Львович,
Московский технический университет связи и информатики, Москва, Россия, ***y.l.leokhin@mtuci.ru***

Дымкова Светлана Сергеевна,
Московский технический университет связи и информатики, Москва, Россия, ***s.s.dymkova@mtuci.ru***

Фатхулин Тимур Джалилевич,
Московский технический университет связи и информатики, Москва, Россия, ***t.d.fatkhulin@mtuci.ru***

Мяличева Альбина Андреевна,
Московский технический университет связи и информатики, Москва, Россия

Manuscript received 30 September 2025;
Accepted 11 December 2025

Ключевые слова: *метод, информация, прогнозирование, программное обеспечение, алгоритм, дефект*

В настоящей работе исследуются проблемы управления в организационных системах IT-компаний, связанные с прогнозированием дефектов в исходном коде программных продуктов. Цель данной работы заключается в определении наиболее эффективного метода, позволяющего с высокой точностью прогнозировать дефекты в исходном коде программных продуктов, реализуемых организационными системами IT-компаний. Актуальность работы обусловлена тем, что традиционные статистические методы и модели не дают достаточно точный прогноз о качестве продукта, а статические инструменты анализа кода имеют большое число ложноположительных и ложноотрицательных срабатываний. В связи с этим возрастает потребность в применении новых методов при прогнозировании ошибок в коде программных продуктов. При этом стоит учитывать, что не все новые алгоритмы и модели, реализующие методы, подходят для решения данной проблемы. Объектом исследования является растущий спрос на качественные программные продукты, реализуемые организационными системами IT-компаний. Предметом исследования являются метрики оценки качества методов и алгоритмов, предназначенных для прогнозирования дефектов в исходном коде программных продуктов. Метриками оценки эффективности являются точность, полнота и F1-мера. В результате проведенных экспериментов сделаны выводы об эффективности каждой модели, а также обусловлен выбор наиболее подходящего алгоритма для решения данной прикладной задачи. Обозначены перспективы дальнейших исследований, направленных на улучшение показателей методов прогнозирования дефектов в исходном коде программных продуктов, реализуемых организационными системами IT-компаний. Методологической основой работы служат методы анализа, сопоставления, сравнения, эксперимент и обобщение.

Информация об авторах:
Леохин Юрий Львович, *профессор, д.т.н., Московский технический университет связи и информатики, Москва, Россия, orcid.org/0000-0003-3321-4497*
Дымкова Светлана Сергеевна, *к.т.н., Московский технический университет связи и информатики, Москва, Россия, orcid.org/0000-0003-0945-9850*
Фатхулин Тимур Джалилевич, доцент, *к.т.н., Московский технический университет связи и информатики, Москва, Россия, orcid.org/0000-0003-0998-1055*
Мяличева Альбина Андреевна, *магистрант, Московский технический университет связи и информатики, Москва, Россия, orcid.org/0009-0004-3267-4146*

Для цитирования:
Леохин Ю.Л., Дымкова С.С., Фатхулин Т.Д., Мяличева А. А. Методы прогнозирования дефектов в программных продуктах на основе ретроспективной и текущей информации // T-Comm: Телекоммуникации и транспорт. 2026. Том 20. №1. С. 52-59.

For citation:
Yu.L. Leokhin, S.S. Dymkova, T.D. Fatkhulin, A.A. Myalicheva, "Methods of predicting defects in software products based on retrospective and current information," *T-Comm*, 2026, vol. 20, no. 1, pp. 52-59. (*in Russian*)

Введение

В настоящее время IT-компании представляют собой сложные организационные системы. Управление в них сводится не только к менеджменту бизнес-процессов, политик и процедур, регулирующих деятельность таких организаций, но также важной составляющей является фаза контроля качества разрабатываемых программных продуктов. В области разработки программного обеспечения главной задачей является предоставление качественного продукта, который отвечает заявленным требованиям, и действия его компонентов являются предсказуемыми. Для решения данной задачи необходимо поставлять программное обеспечение (ПО) без ошибок и уязвимостей в программном коде. Существуют различные способы, помогающие достичь этой цели: традиционные модели, появление которых датируется серединой прошлого века, и современные подходы, основанные на применении машинного обучения (МО) [1, 9, 14, 15, 17-19].

Более адаптивными к разнообразным методологиям разработки ПО и доменным областям проектов являются методы прогнозирования дефектов, основанные на машинном обучении. Данный подход зарекомендовал себя как эффективный и дающий наиболее точный прогноз по сравнению с другими методами [9, 13].

Для достижения поставленной ранее цели необходимо решить следующие задачи:

- проанализировать основные методы и алгоритмы, применяемые в прогнозировании дефектов ПО;
- осуществить анализ обучающих данных;
- провести эксперименты по обучению моделей, применяемых для решения указанной проблемы;
- провести сравнительный анализ обученных моделей;
- сделать вывод о качестве прогнозирования моделей.

1 Традиционные методы прогнозирования дефектов в исходном программном коде

В прогнозировании дефектов ПО применяются различные по своим теоретическим и практическим принципам методы. Первыми методами, заложившими основу для дальнейших исследований, стали эвристические, опирающиеся на опыт и интуицию разработчиков, но лишенные систематичности и объективности (экспертный метод, метод исторических аналогий). Данная проблема была решена применением статистических методов [16] в обнаружении дефектных участков кода [1, 9, 18]. Такие методы, как модели роста надежности (например, модель Холстеда, модель Мотли-Брукса), конструктивная модель качества (Constructive Quality Model, COQUALMO), позволили более точно оценивать качество разрабатываемого продукта на основе характеристик его кодовой базы.

С 70-х годов прошлого века активно применяли модели роста надежности ПО. Каждая модель была разработана под конкретное ПО (например, системное, функционирующее в режиме реального времени), что делает их устаревшими для современных программных продуктов [11, 12, 19]. Конструктивная модель качества (80-е годы XX века) по своей структуре намного сложнее моделей роста надежности, так как прогнозирование дефектов рассматривается в контексте сроков сдачи и бюджета проекта. Применение описанных моделей в современных реалиях является неэффективным из-за их громоздкости и неспособности адаптироваться к разнообразным методологиям разработки и доменным областям проектов [15].

С 90-х годов начали широко применять методы МО, так как данное направление оправдало свою эффективность во многих других областях [2-8, 11, 12]. Для прогнозирования дефектов используют как ML (от англ. Machine Learning, ML – машинное обучение) методы, так и DL (от англ. Deep Learning, DL – глубокое обучение) методы [10, 15]. Эти методы отличаются гибкостью и способностью адаптироваться к любым типам проектов и языкам программирования. Также алгоритмы машинного обучения имеют динамичный характер, и по мере обучения улучшается их прогностическая способность. Обучаясь на различных наборах данных и фиксируя сложные закономерности внутри кодовой базы, модели на основе ML и DL демонстрируют повышенную точность прогнозирования дефектов по сравнению с традиционными методами.

2 Современные методы для решения задачи классификации программного кода

В прогнозировании дефектного кода используются, как правило, методы машинного обучения, предназначенные для решения задачи классификации. В данном случае необходимо определить входной экземпляр как подверженный дефектам или без дефектов. Основываясь на практике применения различных моделей и их эффективности прогнозирования, можно выделить наиболее подходящие алгоритмы обучения.

Наивный байесовский классификатор (Naïve Bayes classifier, NB) – это классификатор, в принцип работы которого заложена теорема Байеса и предположение о том, что признаки являются независимыми. Несмотря на то, что в реальных задачах предположение о независимости признаков часто нарушается, что может привести к ухудшению качества прогноза, данный классификатор иногда успешно конкурирует с более сложными классификаторами, в том числе с нейронными сетями [9, 13].

Методы опорных векторов (Support Vector Machines, SVM) – алгоритмы машинного обучения, которые способны решать линейные и нелинейные задачи путем поиска оптимальной гиперплоскости, разделяющей объекты разных классов. Задача алгоритма сводится к максимизации расстояния, или зазора, между гиперплоскостью и объектами классов, что позволяет уменьшить среднюю ошибку [1, 9, 10, 15].

Метод случайного леса (Random Forest, RF) – алгоритм машинного обучения, заключающийся в использовании ансамбля деревьев решений (decision tree). Принцип работы алгоритма основан на определении класса каждым деревом решений, входящим в комитет. Тот класс, который получил наибольшее количество голосов, считается ответом модели. Количество используемых деревьев, критерий разбиения и число признаков для разбиения имеют решающее значение в обучении модели, поэтому необходимо проводить тщательную настройку гиперпараметров, чтобы предотвратить переобучение [1, 9, 15].

Другими ансамблевыми методами, которые также основаны на построении деревьев решений, являются градиентный бустинг (Gradient Boosting, GB) и его улучшенная реализация – экстремальный градиентный бустинг (Extreme Gradient Boosting, XGBoost).

Градиентный бустинг – это ансамблевый алгоритм, в основе которого лежит оценка функции потерь с использованием градиентного спуска, то есть на каждом шаге измеряется, насколько были ошибочны предыдущие предсказания за счет расчета отрицательного градиента ошибки. Таким образом, алгоритм последовательно обучает модели исправлять ошибки, допущенные предыдущими моделями в процессе обучения [9, 13, 18, 19].

XGBoost – это усовершенствованная версия алгоритма градиентного бустинга, которая позволяет быстрее обучать модели, давая более точный результат. Целью создания данного алгоритма было достижение эффективности использования вычислительных ресурсов и более точных прогнозов. В основу модифицированной версии заложены методы оптимизации и регуляризации (L-1, L-2), что способствует улучшению производительности и уменьшению рисков переобучения модели [9, 13, 18, 19].

Многослойный персептрон (MultiLayer Perceptron, MLP) представляет собой простую архитектуру нейронной сети, состоящей из входного и выходного слоев, между которыми располагаются скрытые полносвязные слои нейронов. Обучение модели происходит методом обратного распространения ошибки для минимизации функции потерь. Нейроны изменяют значения предыдущего слоя во взвешенную сумму, которая подается на вход функции активации для нелинейного преобразования данных, после чего к «активациям» добавляются смещения и делается прогноз [9, 13, 18, 19].

3 Особенности получения данных, используемых в задаче классификации

Прогнозирование дефектов основывается на анализе статических метрик исходного кода проектов. Измерение ПО на основе метрик необходимо для определения и улучшения качества продукта, поэтому данные характеристики кода выбраны для решения задачи классификации. К основным метрикам, на которых строится прогноз о качестве, относятся метрики размерности и сложности программного кода.

Одними из наиболее используемых датасетов (от англ. dataset – набор данных), применяемых для обучения моделей, являются датасеты репозитория PROMISE [9, 13, 18, 19]. Их недостатком является небольшой размер – до 17 тысяч единиц. Также статические данные, содержащиеся в датасетах, собраны с несовременных проектов, датируемых началом 2000-х годов. Тем не менее, данные датасеты широко применяются для обучения моделей [9, 13, 18, 19].

Многие исследователи в качестве признаков также используют JIT-метрики кода (от англ. Just-In-Time, JIT – точно в срок) [9, 13]. JIT-подход основан на анализе изменений в исходном коде, которые отражены в коммитах (от англ. commit - «снимки» состояния проекта в определённый момент времени в Git). С помощью JIT-метрик можно получить более точный прогноз [18, 19]. Системы контроля версий позволяют получить доступ к данным об изменениях в проекте. К их числу относятся история изменений, количество изменений, автор, сообщение коммита и другие.

В рамках данного исследования было подготовлено 3 датасета. Первый датасет был собран из датасетов проектов на языке C репозитория PROMISE. В общей сложности размер датасета составил 37735 экземпляров с размерностью в 40 признаков. Второй датасет также был собран из датасетов PROMISE, но за основу были взяты проекты на Java [1]. Размер второго датасета составляет 12772, размерность – 21. Третий датасет был собран из датасетов открытого ресурса [9, 13, 18, 19]. Сборный датасет содержит данные как о статических метриках кода, так и о JIT-метриках проектов на Java и состоит из 5371 экземпляра и 38 признаков.

В силу своей природы необработанные датасеты содержат несбалансированные данные, так как на практике объём ошибочного кода во много раз меньше объёма кода без ошибок. Для того чтобы модели обучились в равной степени классифицировать оба класса, важной задачей на этапе предобработки данных является балансировка классов.

4 Методология проведения обучения моделей

На этапе предобработки данных, как было отмечено выше, необходимо устранить разницу в количестве экземпляров двух классов. Методом балансировки классов выбран SMOTE (от англ. Synthetic Minority Oversampling Technique – метод искусственного увеличения примеров миноритарного класса), или метод увеличения числа экземпляров миноритарного класса за счет генерации искусственных экземпляров.

Целевые метки классов были преобразованы в числовые значения 0 (код без дефектов) и 1 (код с дефектами). Также проведена нормализация входных значений, то есть приведение значений к диапазону от -1 до 1, так как в датасетах наблюдается широкий диапазон значений. Деление датасетов на обучающую и тестовую выборку происходит с расчетом 80/20.

Оптимизация гиперпараметров – трудоемкий этап, требующий много вычислительных ресурсов и времени, особенно в обучении ансамблевых моделей и нейронных сетей, поэтому в рамках данных экспериментов были отобраны наиболее значимые гиперпараметры для каждой из моделей, представленные в таблице 1.

Для оценки эффективности обученных моделей выбраны ***точность*** (precision), ***полнота*** (recall) и ***F1-мера*** (F1-score) [9, 13, 18, 19].

Точность – это метрика, показывающая долю истинно положительных результатов в общем количестве положительных результатов, которые предсказала модель, то есть способность модели правильно относить экземпляры к классу «код с дефектами».

Полнота отражает долю истинно положительных результатов модели из всего числа объектов положительного класса, то есть показывает способность модели относить экземпляр положительного класса к данному классу.

F1-мера – это среднее гармоническое между точностью и полнотой. Поскольку увеличить значение точности без потери в полноте и наоборот невозможно на практике, значение F1-меры позволяет измерить баланс между двумя метриками.

Таблица 1

Настраиваемые гиперпараметры моделей

Модель	Гиперпараметр	Описание
SVM	**kernel** (значения: linear, poly, rbf, sigmoid, precomputed)	Ядерная функция модели для работы с данными в высокоразмерном пространстве
RF	n_estimators (от 100 до 500 с шагом 20)	Количество деревьев решений
	max_depth (от 2 до 32)	Максимальная глубина дерева
	min_samples_split (от 2 до 10)	Минимальное количество образцов ноды (от англ. node – элемент системы) для ее разделения на дочерние ноды
	min_samples_leaf (от 1 до 10)	Минимальное количество образцов, которые должны оставаться в ноде после разделения
GB	learning_rate (от 0.001 до 1)	Скорость обучения
	n_estimators (от 100 до 500 с шагом 20)	Количество деревьев решений
	max_depth (от 2 до 32)	Максимальная глубина дерева
XGBoost	learning_rate (от 0.001 до 1)	Скорость обучения
	n_estimators (от 100 до 1000 с шагом 100)	Количество деревьев решений
	max_depth (от 2 до 32)	Максимальная глубина дерева
	min_samples_split (от 2 до 10)	Минимальное количество образцов ноды для ее разделения на дочерние ноды
	min_child_weight (от 1 до 20)	Минимальная сумма весов образцов для предотвращения деления ноды
MLP	learning_rate (значения: constant, adaptive, invscaling)	Изменение скорости обучения
	solver (значения: lbfgs, sgd, adam)	Метод оптимизации для минимизации функции потерь
	activation (значения: identity, logistic, tanh, relu)	Функция активации
	batch_size (от 2 до 32 с шагом 4)	Размер батча для стохастических оптимизаторов

Таким образом, чтобы не улучшать значение одной метрики в ущерб другой, а достичь компромисса для значений обеих метрик, необходимо проводить оптимизацию гиперпараметров моделей для F1-меры.

5 Экспериментальные исследования

Для определения наиболее подходящей модели проведены три эксперимента с использованием описанных датасетов. В каждом эксперименте обучение моделей происходило как с заданными по умолчанию значениями гиперпараметров, так и с оптимизированными. Оптимизация гиперпараметров проведена с помощью библиотеки Optuna для всех моделей, за исключением наивного байесовского классификатора.

В ***эксперименте №1*** проведено обучение моделей на первом сборном сбалансированном датасете, численность экземпляров которого составила 73184. Результаты эксперимента представлены в таблице 2.

Таблица 2

Результаты обучения моделей в эксперименте №1

Модель	С гиперпараметрами по умолчанию			С оптимизированными гиперпараметрами		
	Precision	***Recall***	***F1-мера***	***Precision***	***Recall***	***F1-мера***
NB	0.3112	0.9108	0.6807	-	-	-
SVM	0.9218	0.9078	0.9134	0.9218	0.9078	0.9134
RF	**0.9901**	0.9777	0.9836	**0.9901**	0.9781	0.9838
GB	0.9687	0.9587	0.9632	0.9867	**0.9867**	0.9867
XGB	0.9823	**0.9888**	**0.9854**	0.9888	0.9857	**0.9871**
MLP	0.9747	0.9609	0.9672	0.979	0.9724	0.9754

NB показал самые низкие значения для точности и F1-меры, то есть эта модель чаще остальных относит код без дефектов к классу «код с дефектами».

Основная идея, которая лежит в основе классификации наивного Байеса, заключается в том, что все признаки объекта не зависят друг друга, из чего следует, что качество прогноза данной модели ухудшается при наличии мультиколлинеарности (от англ. multicollinearity – статистическое явление, при котором два или более признаков в модели сильно коррелируют между собой).

При анализе первого датасета на наличие корреляции между признаками было найдено 8 признаков, чей коэффициент корреляции превышал 0.85, чем можно объяснить невысокие значения точности и, соответственно, F1-меры. На основании этого во всех экспериментах во втором опыте NB не использовался, так как значения метрик значительно не изменились бы без соответствующих изменений в наборах данных.

Однако все остальные модели в обоих опытах показали высокие значения по всем метрикам, превышающие 0.9. Значения для модели SVM не изменились во втором опыте, потому что по результатам оптимизации гиперпараметра – ядерной функции – лучшее значение F1-меры показала радиальная базисная функция, которая была задана по умолчанию в первом опыте.

Для некоторых моделей, таких как RF, XGBoost, оптимизация гиперпараметров не дала значительный прирост значений используемых метрик, а для XGBoost даже уменьшилось значение полноты. Наибольший прирост в точности и полноте после оптимизации имеет GB. Несмотря на то, что для настройки данной модели потребовалось больше всего ресурсов, значения метрик увеличились на 0.02 (точность), 0.03 (полнота) и 0.02 (F1-мера). Настройка гиперпараметров также улучшила прогностическую способность MLP.

В совокупности анализируя результаты прогнозов в двух опытах, можно сделать вывод, что лучшие результаты показала модель XGBoost. Несмотря на то, что значения ее точности уступают RF, а во втором опыте значение полноты на 0.001 ниже, чем у GB, она имеет самые высокие значения F1-меры и значение полноты в первом опыте.

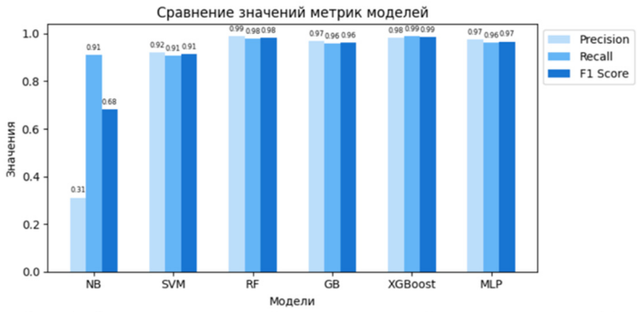

Рис. 1. Сравнение значений метрик моделей в эксперименте №1 (с параметрами по умолчанию)

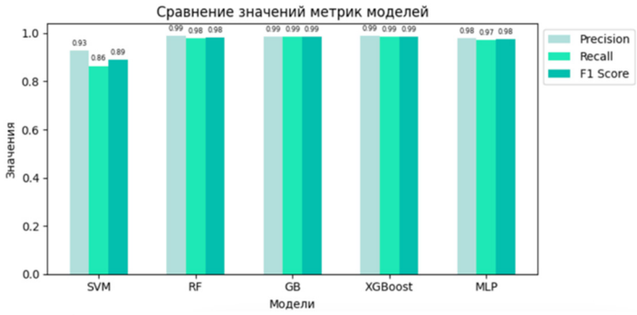

Рис. 2. Сравнение значений метрик моделей в эксперименте №1 (с оптимизированными параметрами)

В ***эксперименте №2*** был применен второй сборный датасет, размер которого после балансировки классов составил 16390 экземпляров. В данном датасете так же, как и в первом, присутствуют коррелирующие признаки. Результаты обучения моделей представлены в таблице 3.

Таблица 3

Результаты обучения моделей в эксперименте №2

Модель	С гиперпараметрами по умолчанию			С оптимизированными гиперпараметрами		
	Precision	*Recall*	*F1-мера*	*Precision*	*Recall*	*F1-мера*
NB	0.2897	0.7029	0.62	-	-	-
SVM	0.6509	0.7257	0.7011	0.6945	0.6929	0.6913
RF	**0.7206**	0.7353	0.7288	0.7218	**0.7576**	**0.7437**
GB	0.6703	0.7281	0.7084	**0.7358**	0.7485	0.7425
XGB	0.7158	**0.7537**	**0.7393**	0.7321	0.7498	0.7422
MLP	0.6606	0.7286	0.7059	0.6964	0.7213	0.7117

NB показал самые низкие результаты, что связано с наличием линейно зависимых признаков. Наибольшее значение точности при обучении с заданными по умолчанию гиперпараметрами имеет модель RF, почти на одну сотую превышая значение модели XGBoost. Точность остальных моделей не достигает 0.7. Как и в первом эксперименте, лучшее значение полноты имеет XGBoost, значение полноты других моделей находится в пределах от 0.7 до 0.73. Соответственно, наибольшей F1-мерой обладает XGBoost.

После оптимизации гиперпараметров значение F1-меры улучшилось у всех моделей, за исключением SVM. По результатам обучения лучший прогноз во втором опыте дает RF, имея самые высокие значения полноты и F1-меры.

Однако значение точности RF ниже, чем у других ансамблевых моделей – наибольшую точность имеет модель GB. MLP в данном опыте обладает большей прогностической способностью, чем SVM, но уступает ансамблевым моделям.

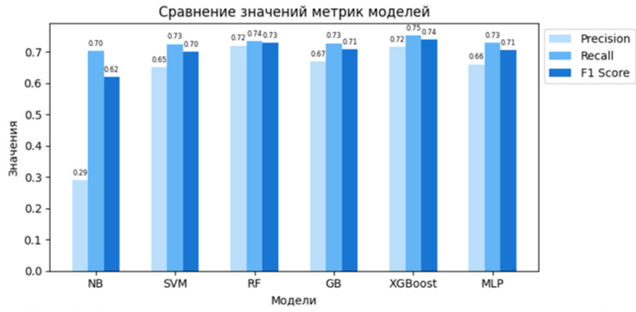

Рис. 3. Сравнение значений метрик моделей в эксперименте №2 (с параметрами по умолчанию)

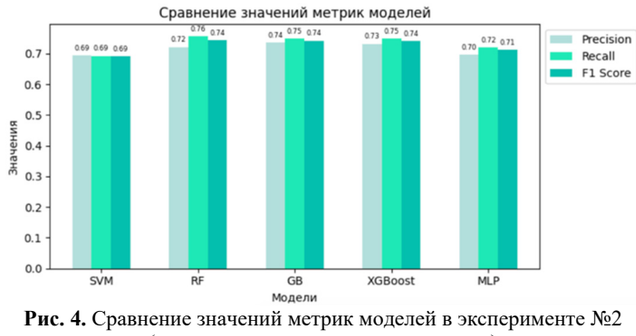

Рис. 4. Сравнение значений метрик моделей в эксперименте №2 (с оптимизированными параметрами)

Эксперимент №3 был проведен с использованием третьего сборного датасета, численность экземпляров которого после балансировки составила 9036. В датасете также имеются коллинеарные признаки. В таблице 4 отражены результаты проведения эксперимента.

Наивный байесовский классификатор так же, как и в предыдущих экспериментах, имеет низкое значение точности, несмотря на сравнимое с другими моделями значением полноты.

В первом опыте лучшую прогностическую способность показывают модели на основе деревьев решений, RF и XGBoost, первая из которых имеет самые высокие значения точности и F1-меры, вторая имеет наибольшее значение полноты среди остальных моделей. Модель GB имеет невысокую эффективность по сравнению с другими ансамблевыми решениями. SVM по сравнению с остальными сложными моделями имеет относительно низкие значения рассматриваемых метрик. Модель MLP показала лучшие результаты, чем GB.

Во втором опыте оптимизация гиперпараметров не смогла повысить результаты для моделей SVM и RF, а для нейронной сети понизила значение F1-меры, что в свою очередь оказало влияние на уменьшение ее точности и полноты. Наиболее качественный прогноз по результатам обучения дает модель GB – у нее получились самые высокие значения метрик, однако значения точности и полноты модели XGBoost только на тысячные доли меньше, чем у GB. Из трех ансамблевых моделей RF имеет наименьшие значения во втором опыте.

Таблица 4

Результаты обучения моделей в эксперименте №3

Модель	С гиперпараметрами по умолчанию			С оптимизированными гиперпараметрами		
	Precision	*Recall*	*F1-мера*	*Precision*	*Recall*	*F1-мера*
NB	0.2779	0.8301	0.646	-	-	-
SVM	0.691	0.832	0.7699	0.691	0.832	0.7688
RF	**0.9442**	0.9148	**0.9259**	0.9442	0.9148	0.9259
GB	0.8305	0.8459	0.8346	**0.9485**	**0.9394**	**0.9419**
XGB	0.9313	**0.9195**	0.9226	0.9474	0.9364	0.9397
MLP	0.8777	0.8739	0.8717	0.8766	0.871	0.8695

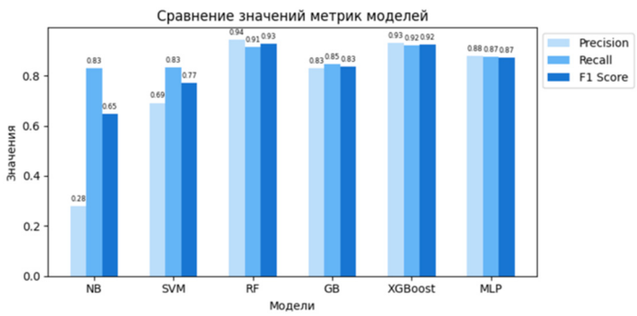

Рис. 5. Сравнение значений метрик моделей в эксперименте №3 (с параметрами по умолчанию)

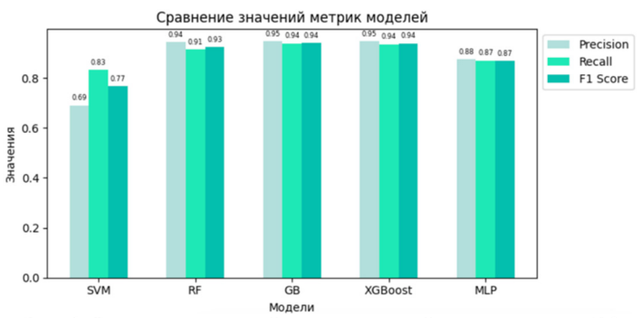

Рис. 6. Сравнение значений метрик моделей в эксперименте №3 (с оптимизированными параметрами)

Таким образом, опираясь на полученные экспериментальные данные, можно сделать вывод, что для задачи классификации исходного кода наиболее подходит алгоритм XGBoost. Однако, несмотря на выявленные преимущества, у данного алгоритма имеются недостатки, связанные с переобучением моделей и негативным влиянием на прогноз аномалий в данных. Также для того, чтобы добиться удовлетворяющих показателей по всем метрикам, необходимо более тщательно проводить настройку гиперпараметров под каждый датасет, так как от структуры ансамбля деревьев зависят предсказания модели.

6 Выводы

В настоящей работе были проведены исследования, направленные на определение наиболее подходящего по метрикам метода прогнозирования дефектов в исходном коде программных продуктов, реализуемых организационными системами IT-компаний. В ходе исследования методов, предназначенных для прогнозирования дефектов в исходном программном коде были сделаны следующие выводы. Считается, что результаты прогнозирования моделей машинного и глубокого обучения значительно превосходят результаты традиционных моделей и инструментов в нахождении ошибок в коде. Однако на данном этапе применения ML/DL в предметной области имеются сложности в общем процессе обучения моделей.

По результатам проведенных экспериментов и анализа предметной области, особенностей данных и моделей машинного обучения, можно сделать выводы, что модель, обученная с помощью алгоритма XGBoost, показала лучшие результаты по сравнению с другими моделями. Данная модель отличается большой обобщающей способностью и возможностью ее применения с различными по своему объему и размерности данными. Значение метрик оценки качества модели XGBoost оказались наилучшими в подавляющем количестве опытов как с настройкой гиперпараметров, так и без ее применения. Также проведенные эксперименты показали эффективность использования XGBoost и продемонстрировали минимальные значения времени обучения модели и высокую скорость предсказания.

Однако выбранный алгоритм обучения, несмотря на свои достоинства, не всегда дает максимально возможные значения точности или полноты, что может быть связано с обучением на зашумленных данных, данных с сильной корреляцией признаков, неподходящей структурой ансамбля или системой принятия решений. В продолжении исследования необходимо решить данные недостатки модели.

Литература

1. *Мяличева А.А., Фатхулин Т.Д.* Анализ методов машинного обучения для прогнозирования дефектов в исходном коде // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 16-19. EDN IVJCZF.

2. *Портнов Э.Л., Фатхулин Т.Д.* Технологии достижения высоких скоростей передачи в современных когерентных DWDM-системах связи // T-Comm: Телекоммуникации и транспорт. 2015. Т. 9. №8. С. 34-37.

3. *Деарт В.Ю., Фатхулин Т.Д.* Анализ современного состояния транспортных сетей с целью внедрения технологии программно-конфигурируемых сетей (SDN) // T-Comm: Телекоммуникации и транспорт. 2017. Том 11. №6. С. 4-9.

4. *Deart V., Fatkhulin T.* Analysis of the functioning of a multi-domain transport software-defined network with controlled optical layer // 2017 21st Conference of Open Innovations Association (FRUCT), Helsinki, Finland, 2017, pp. 79-87, DOI: 10.23919/FRUCT.2017.8250168.

5. *Деарт В.Ю., Фатхулин Т.Д.* Анализ транспортных программно-конфигурируемых сетей (T-SDN) с управляемым оптическим уровнем с целью получения модели, позволяющей оценить возможность предоставления сервиса Bandwidth on Demand // T-Comm: Телекоммуникации и транспорт. 2018. Т.12. №4. С.35-42.

6. *Деарт В.Ю., Фатхулин Т.Д.* Анализ процесса создания суперканала с необходимой пропускной способностью в сети, построенной по технологии транспортных программно-конфигурируемых сетей (T-SDN) // T-Comm: телекоммуникации и транспорт. 2018. Том 12. №10. С. 23-30.

7. *Leokhin Y. L., Fatkhulin T. D.* Approach to Estimating the Probability of Providing “Cloud” Services in the SDN // 2020 Systems of

Signals Generating and Processing in the Field of on Board Communications, Moscow, Russia, 2020, pp. 1-9, DOI: 10.1109/IEEECONF48371.2020.9078593.

8. *Leokhin Y. L., Fatkhulin T. D.* Evaluation of Service Availability in Software-Defined Optical Network // 2021 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russia, 2021, pp. 1-6, DOI: 10.1109/IEEECONF51389.2021.9416122.

9. *Вишневский В.М., Леохин Ю.Л., Фатхулин Т.Д., Занегин А.В.* Методы машинного обучения в решении задачи прогнозирования спроса на отдельные виды товаров // T-Comm: Телекоммуникации и транспорт. 2024. Том 18. №10. С. 34-43.

10. *Ринас Н.А., Золкин А.Л., Каберова А.Р., Скибин Ю.В.* Влияние автоматизации и искусственного интеллекта на социальное неравенство // Экономика и управление: проблемы, решения. 2025. Т. 7, № 1(154). С. 116-125. DOI 10.36871/ek.up.p.r.2025.01.07.015. EDN BGBKZZ.

11. *Беспалова В.В., Каберова А.Р., Белинская Д.Б.* и др. Методический подход к управлению устойчивостью развития региона // Экономика и управление: проблемы, решения. 2024. Т. 11, № 11(152). С. 88-93. DOI 10.36871/ek.up.p.r.2024.11.11.011. EDN GDEJYC.

12. *Драгуленко В.В., Золкин А.Л., Есина О.И., Каберова А.Р.* Влияние численности населения на экономический рост и развитие стран // Экономика и управление: проблемы, решения. 2024. Т. 11, № 9(150). С. 67-75. DOI 10.36871/ek.up.p.r.2024.09.11.009. EDN ANEUBG.

13. *Леохин Ю.Л., Фатхулин Т.Д., Занегин А.В.* Модификация метода градиентного усиления для прогнозирования спроса на отдельные виды товаров // Наукоемкие технологии в космических исследованиях Земли. 2025. Т. 17. № 2. С. 32-41. DOI: 10.36724/2409-5419-2025-17-2-32-41

14. *Леохин Ю.Л., Дымкова С.С., Фатхулин Т.Д.* Исследование и разработка инструментальных средств повышения качества изображений // T-Comm: Телекоммуникации и транспорт. 2025. Т. 19. №4. С. 45-56. (in English)

15. *Леохин Ю.Л., Дымкова С.С., Фатхулин Т.Д.* Методы машинного обучения в прикладных задачах прогнозирования динамично изменяющихся данных // T-Comm: Телекоммуникации и транспорт. 2025. Т. 19. №8. С. 49-63.

16. *Дымкова С.С., Кретова И.С., Варламов О.В.* Наукометрический анализ результатов рецензирования материалов конференции TIRVED2024 // T-Comm: Телекоммуникации и транспорт. 2024. Т. 18. №12. С. 19-26.

17. *Leokhin Y., Fatkhulin T. Boitsov K.* Computer Vision Methods in Applied Problems of Classifying Objects in Images // 2025 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), St. Petersburg, Russian Federation, 2025, pp. 1-10, DOI: 10.1109/WECONF65186.2025.11017109.

18. *Leokhin Y., Fatkhulin T., Zanegin A., Rakhmatova A.* Researching the Efficiency of Machine Learning Methods Used in Forecasting Demand for Certain Types of Goods // 2025 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russian Federation, 2025, pp. 1-8, DOI: 10.1109/IEEECONF64229.2025.10948113.

19. *Fatkhulin T., Leokhin Y., Zanegin A., Rakhmatova A.* Development and Research of a Modified Gradient Boosting Method Effectiveness to Solve Applied Problems of Time-Series Forecasting // 2025 Systems of Signals Generating and Processing in the Field of on Board Communications, Moscow, Russian Federation, 2025, pp. 1-10, DOI: 10.1109/IEEECONF64229.2025.10948023.

20. *Леохин Ю. Л., Дымкова С. С., Фатхулин Т. Д., Зозуля И. С.* Методы и алгоритмы интеллектуальной поддержки принятия управленческих решений в организационных системах торговых компаний // T-Comm: Телекоммуникации и транспорт. 2025. Т. 19. №12. С. 44-50.

METHODS OF PREDICTING DEFECTS IN SOFTWARE PRODUCTS BASED ON RETROSPECTIVE AND CURRENT INFORMATION

Yuri L. Leokhin, *Moscow Technical University of Communications and Informatics, Moscow, Russia,* ***y.l.leokhin@mtuci.ru***
Svetlana S. Dymkova, *Moscow Technical University of Communications and Informatics, Moscow, Russia,* ***s.s.dymkova@mtuci.ru***
Timur D. Fatkhulin, *Moscow Technical University of Communications and Informatics, Moscow, Russia,* ***t.d.fatkhulin@mtuci.ru***
Albina A. Myalicheva, *Moscow Technical University of Communications and Informatics, Moscow, Russia*

Abstract
This paper examines the management problems in organizational systems of IT companies related to predicting defects in the source code of software products. The purpose of this work is to determine the most effective method for accurately predicting defects in the source code of software products implemented by organizational systems of IT companies. The relevance of the work is due to the fact that traditional statistical methods and models do not provide a sufficiently accurate forecast of the quality of the product, and static code analysis tools have a large number of false positives and false negatives. In this regard, there is an increasing need to apply new methods in predicting errors in the code of software products. At the same time, it is necessary to take into account that not all new algorithms and models implementing the methods are suitable for solving this problem. The object of the study is the growing demand for high-quality software products implemented by organizational systems of IT companies. The subject of the study is the metrics for assessing the quality of methods and algorithms designed to predict defects in the source code of software products. The performance evaluation metrics are accuracy, completeness and F1-measure. As a result of the experiments, conclusions were made about the performance of each model, and the choice of the most suitable algorithm for solving this applied problem was determined. The prospects for further research aimed at improving the performance of defect prediction methods in the source code of software products implemented by the organizational systems of IT companies are outlined. The methodological basis of the work is the methods of analysis, comparison, experiment and generalization.

Keywords: *method, information, forecasting, software, algorithm, defect*

References

[1] A. A. Myalicheva and T. D. Fatkhulin, "Analysis of machine learning methods for predicting defects in source code," *Proceedings of the North Caucasian branch of the Moscow Technical University of Communications and Informatics*, 2024. No. 2. pp. 16-19. (in Russian).
[2] E. L. Portnov and T. D. Fatkhulin, "Technologies aimed at achieving high speed transmission in modern coherent DWDM communication systems," *T-Comm*. 2015. Vol 9. No.8, pp. 34-37. (in Russian).
[3] V. Yu. Deart and T. D. Fatkhulin, "Analysis of current state of transport networks with the purpose of introducing software defined networks (SDN) technology," *T-Comm*, 2017, vol. 11, no.6, pp. 4-9. (in Russian).
[4] V. Deart and T. Fatkhulin, "Analysis of the functioning of a multi-domain transport software-defined network with controlled optical layer," *2017 21st Conference of Open Innovations Association (FRUCT)*, Helsinki, Finland, 2017, pp. 79-87, DOI: 10.23919/FRUCT.2017.8250168.
[5] V. Yu. Deart and T. D. Fatkhulin, "Analysis of transport software-defined networks (T-SDN) with controlled optical layer to obtain a model providing assesment of the possibility of bandwidth on demand service," *T-Comm*, 2018, vol. 12, no.4, pp. 35-42. (in Russian).
[6] V. Yu. Deart and T. D. Fatkhulin, "Analysis of the process of creating a superchannel with the necessary capacity in the network organized according to transport software-defined networks (T-SDN) technology," *T-Comm*, 2018, vol. 12, no.10, pp. 23-30. (in Russian).
[7] Y. L. Leokhin and T. D. Fatkhulin, "Approach to Estimating the Probability of Providing "Cloud" Services in the SDN," *2020 Systems of Signals Generating and Processing in the Field of on Board Communications*, Moscow, Russia, 2020, pp. 1-9, DOI: 10.1109/IEEECONF48371.2020.9078593.
[8] Y. L. Leokhin and T. D. Fatkhulin, "Evaluation of Service Availability in Software-Defined Optical Network," *2021 Systems of Signals Generating and Processing in the Field of on Board Communications*, Moscow, Russia, 2021, pp. 1-6, DOI: 10.1109/IEEECONF51389.2021.9416122.
[9] V. M. Vishnevsky, Yu. L. Leokhin, T. D. Fatkhulin and A. V. Zanegin, "Machine learning methods in solving the problem of forecasting demand for specific types of goods," *T-Comm*, vol. 18, no. 10, pp. 34-43. (in Russian)
[10] N. A. Rinas, A. L. Zolkin, A. R. Kaberova and Yu. V. Skibin, "The Impact of Automation and Artificial Intelligence on Social Inequality," *Economy and Management: Problems, Solutions*, 2025, Vol. 7, No. 1(154), pp. 116-125. DOI 10.36871/ek.up.p.r.2025.01.07.015. (in Russian).
[11] V. V. Bespalova, A. R. Kaberova, D. B. Belinskaya [et al.], "Methodological approach to managing the sustainability of regional development," *Economy and Management: Problems, Solutions*, 2024, Vol. 11, No. 11(152), pp. 88-93. DOI 10.36871/ek.up.p.r.2024.11.11.011. (in Russian).
[12] V. V. Dragulenko, A. L. Zolkin, O. I. Esina and A. R. Kaberova, "The Impact of Population on Economic Growth and Development of Countries," *Economy and Management: Problems, Solutions*, 2024, Vol. 11, No. 9(150), pp. 67-75. DOI 10.36871/ek.up.p.r.2024.09.11.009. (in Russian).
[13] Yu. L. Leokhin, T. D. Fatkhulin, A. V. Zanegin, "The gradient boosting method modification to forecast demand for individual types of goods," H&ES Reserch. 2025. Vol. 17. No. 2, pp. 32-41. DOI: 10.36724/2409-5419-2025-17-2-32-41 (in Russian).
[14] Yu. L. Leokhin, S. S. Dymkova, T. D. Fatkhulin, "Research and development of image improvement tools," *T-Comm*, 2025, vol. 19, no. 4, pp. 45-56.
[15] Yu. L. Leokhin, S. S. Dymkova, T. D. Fatkhulin, "Machine learning methods in applied problems of forecasting dynamically changing data", *T-Comm*, 2025, vol. 19, no.8, pp. 49-63. (in Russian)
[16] S. S. Dymkova, I. S. Kretova and O. V. Varlamov, "Conference papers per-reviewing results: TIRVED-2024 scientometric research," *T-Comm*, vol. 18, no.12 pp. 19-26. (in Russian).
[17] Y. Leokhin, T. Fatkhulin and K. Boitsov, "Computer Vision Methods in Applied Problems of Classifying Objects in Images," *2025 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF)*, St. Petersburg, Russian Federation, 2025, pp. 1-10, DOI: 10.1109/WECONF65186.2025.11017109.
[18] Y. Leokhin, T. Fatkhulin, A. Zanegin and A. Rakhmatova, "Researching the Efficiency of Machine Learning Methods Used in Forecasting Demand for Certain Types of Goods," *2025 Systems of Signals Generating and Processing in the Field of on Board Communications*, Moscow, Russian Federation, 2025, pp. 1-8, DOI: 10.1109/IEEECONF64229.2025.10948113.
[19] T. Fatkhulin, Y. Leokhin, A. Zanegin and A. Rakhmatova, "Development and Research of a Modified Gradient Boosting Method Effectiveness to Solve Applied Problems of Time-Series Forecasting," *2025 Systems of Signals Generating and Processing in the Field of on Board Communications*, Moscow, Russian Federation, 2025, pp. 1-10, DOI: 10.1109/IEEECONF64229.2025.10948023.
[20] Yu. L. Leokhin, S. S. Dymkova, T. D. Fatkhulin and I. S. Zozulya, "Methods and algorithms of intellectual support for making management decisions in organizational systems of trading companies," *T-Comm*, 2025, vol. 19, no.12. pp. 44-50. (in Russian)

Information about authors:
Yuri L. Leokhin, *full professor, Dr. Sc. (Tech.), Moscow Technical University of Communications and Informatics, Moscow, Russia, orcid.org/0000-0003-3321-4497*
Svetlana S. Dymkova, *Candidate Sc. (Tech.), Moscow Technical University of Communications and Informatics, Moscow, Russia, orcid.org/0000-0003-0945-9850*
Timur D. Fatkhulin, *Dpt. of MC and IT, Docent, Candidate Sc. (Tech.), Moscow Technical University of Communications and Informatics, Moscow, Russia, orcid.org/0000-0003-0998-1055*
Albina A. Myalicheva, *Dpt. of MC and IT, master's student, Moscow Technical University of Communications and Informatics, Moscow, Russia, orcid.org/0009-0004-3267-4146*