(19) Евразийское патентное ведомство

(11) 039867 (13) B1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2022.03.22

(21) Номер заявки
202092870

(22) Дата подачи заявки
2020.12.23

(51) Int. Cl. ***G06F 11/36*** (2006.01)
G06N 20/00 (2019.01)
G06F 17/16 (2006.01)

(54) СПОСОБ И СИСТЕМА ОЦЕНКИ ВЕРОЯТНОСТИ ВОЗНИКНОВЕНИЯ КРИТИЧЕСКИХ ДЕФЕКТОВ ПО КИБЕРБЕЗОПАСНОСТИ НА ПРИЕМО-СДАТОЧНЫХ ИСПЫТАНИЯХ РЕЛИЗОВ ПРОДУКТОВ

(31) **2020131498**
(32) **2020.09.24**
(33) **RU**
(43) **2022.03.18**

(71)(73) Заявитель и патентовладелец:
ПУБЛИЧНОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО "СБЕРБАНК РОССИИ" (ПАО СБЕРБАНК) (RU)

(72) Изобретатель:
Кудияров Дмитрий Сергеевич, Биферт Виталий Оттович, Демьянова Елена Анатольевна, Глотов Геннадий Геннадьевич, Анистратенко Александр Артурович (RU)

(74) Представитель:
Герасин Б.В. (RU)

(56) TIANCHI ZHOUA et al., "Improving defect prediction with deep forest", 05.06.2019, 13 л., [онлайн] [найдено 14.05.2021]. Найдено в <https://xin-xia.github.io/publication/ist192.pdf>

MONIKA YADAV et al., "Deep Learning For Software Defect Prediction in time", 5th IEEE International Conference on Parallel, Distributed and Grid Compu-ting(PDGC-2018), 2018, 12 л., [онлайн] [найдено 14.05.2021]. Найдено в <https://www.researchgate.net/publication/334080551_Deep_Learning_for_Software_Defect_Prediction_in_time>

CN-A-103257921

US-A1-20170192880

(57) Заявленное техническое решение относится к автоматизированному способу и системе оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов с помощью алгоритмов машинного обучения. Техническим результатом от реализации заявленного способа является повышение скорости и точности оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов. Указанный технический результат достигается благодаря осуществлению компьютерно-реализуемого способа оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов, выполняемого с помощью по меньшей мере одного процессора и содержащего этапы, на которых получают данные, содержащие информацию, по меньшей мере, о релизах программных продуктов, находящихся в статусах разработки, предшествующих приемо-сдаточным испытаниям; осуществляют обработку полученных данных с помощью модели машинного обучения (МО), причем в ходе указанной обработки выполняют разделение импортированных данных на категориальные и численные переменные; преобразование полученных переменных, при котором выполняется векторизация категориальных переменных и нормализация численных переменных; конкатенацию преобразованных переменных и построение на их основе вектора, соответствующего оцениваемым релизам, находящимся в статусах, предшествующих приемо-сдаточным испытаниям; классификацию с помощью полученного вектора каждого релиза с присвоением степени вероятности возникновения критических дефектов по кибербезопасности на приемосдаточных испытаниях релизов продуктов.

039867 B1

039867 B1

Область техники

Заявленное техническое решение в общем относится к области вычислительной техники, а в частности к автоматизированному способу и системе оценки вероятности возникновения критических дефектов по кибербезопасности на приемосдаточных испытаниях релизов продуктов с помощью алгоритмов машинного обучения.

Уровень техники

Разработка программного обеспечения для крупных финансовых организаций (например, банков) всегда трудоемкий и кропотливый труд. Кроме того, при разработке программного продукта необходимо учесть все риски возникновения дефектов кибербезопасности. Для данных проверок привлекаются эксперты по кибербезопасности, которые вручную проверяют наличие критических дефектов в разрабатываемом программном продукте. Эксперты кибербезопасности, работающие с командами разработчиков (Agile-командами), разрабатывающими банковские продукты, могут привлекаться к подготовке релиза на разных этапах: идея, проектирование, кодирование, тестирование, приемо-сдаточные испытания. Чем раньше эксперт будет привлечен к подготовке релиза, тем менее трудоемко для команды выполнение требований кибербезопасности и ниже риски кибербезопасности. Однако далеко не всегда есть возможность привлечь экспертов на ранних этапах. В случае обнаружения серьезных нарушений требований кибербезопасности на испытаниях, эксперт кибербезопасности указывает соответствующее критическое замечание в протоколе испытаний, что запрещает команде внедрять релиз в промышленной среде. Подобные ситуации увеличивают время разработки продуктов.

Привлечение эксперта кибербезопасности исключительно на приемосдаточные испытания в условиях большого числа синхронных agile-спринтов у разных команд и неосведомленности эксперта о реализации в релизах функций по кибербезопасности влекут существенные пики в нагрузке на экспертов, что приводит к увеличению времени проверки и как следствие происходит сдвиг релиза программного продукта.

Из уровня техники известен патент US 8631384B2 "Creating a test progression plan", патентообладатель: IBM, опубликовано: 01.12.2011. В данном решении описывается автоматизированный процесс составления планов тестирования программных продуктов. Известное решение обеспечивает автоматическое создание плана выполнения теста программного обеспечения путем вычисления для каждой единицы периода тестирования x усилий по выполнению тестовых блоков ATTx и усилий по завершению выполнения тестового блока CCx. В вычислении вводятся три переменные, характеризующие стратегию тестирования: эффективность, которая представляет эффективность группы тестирования, коэффициент плотности дефектов и значение коэффициента проверки. Выбирая стратегию тестирования, менеджер тестов определяет значения трех переменных, которые влияют на план развития. Во время выполнения теста кумулятивная кривая "попытка" значений ATTx и кумулятивная кривая "завершение" значений CCx позволяют менеджеру тестирования сравнить уже предпринятые усилия с ожидаемыми усилиями, предпринятыми для испытательных блоков, которые были предприняты и для испытательных единиц, которые были закончены, то есть, когда дефекты, найденные в коде, были исправлены.

Недостатком известного решения в данной области техники является отсутствие возможности автоматизированной оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов.

Раскрытие изобретения

В заявленном техническом решении предлагается новый подход к оценке вероятности возникновения критических дефектов по кибербезопасности на приемосдаточных испытаниях релизов продуктов. В данном решении используется алгоритм машинного обучения, который позволяет автоматизировать процесс проверки релизов на критические дефекты по кибербезопасности и с высокой точностью оценивает их возникновение.

Таким образом, решается техническая проблема автоматизированной оценки вероятности возникновения критических дефектов по кибербезопасности на приемосдаточных испытаниях релизов продуктов.

Техническим результатом, достигающимся при решении данной проблемы, является повышение скорости и точности оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов.

Указанный технический результат достигается благодаря осуществлению компьютерно-реализуемого способа оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов, выполняемого с помощью по меньшей мере одного процессора и содержащего этапы, на которых получают данные, содержащие информацию, по меньшей мере, о релизах программных продуктов, находящихся в статусах разработки, предшествующих приемо-сдаточным испытаниям; осуществляют обработку полученных данных с помощью модели машинного обучения (МО), причем в ходе указанной обработки выполняют: разделение импортированных данных на категориальные и численные переменные; преобразование полученных переменных, при котором выполняется векторизация категориальных переменных и нормализация численных переменных; конкатенацию преобразованных переменных и построение на их основе вектора, соответствующего оце-

ниваемым релизам, находящимся в статусах, предшествующих приемо-сдаточным испытаниям; классификацию с помощью полученного вектора каждого релиза с присвоением степени вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов.

В одном из частных вариантов реализации способа обработка полученных данных осуществляется с помощью модели машинного обучения на базе классификатора случайного леса (random forest).

В другом частном варианте реализации способа модель МО предварительно обучают на исторических данных о получении критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов программных продуктов.

В другом частном варианте реализации способа обработка полученных данных осуществляется с помощью ансамбля нейронных сетей.

В другом частном варианте реализации способа получаемые данные дополнительно содержат информацию о записях о задачах категории релиз в системе управления задачами на разработку, включающую в себя: номер, тип, важность, статус, время создания, время взятия в работу, время отметки как решенного, список зависимых записей о задачах и типы зависимостей, список зависящих записях о задачах и типы зависимостей.

В другом частном варианте реализации способа получаемые данные дополнительно содержат информацию о количестве критических дефектов по кибербезопасности, выявленных на приемо-сдаточных испытаниях всех предыдущих релизов программных продуктов.

Кроме того, заявленный технический результат достигается за счет системы оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов содержащей: по меньшей мере один процессор; по меньшей мере одну память, соединенную с процессором, которая содержит машиночитаемые инструкции, которые при их выполнении по меньшей мере одним процессором обеспечивают выполнение способа оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов.

Краткое описание чертежей

Признаки и преимущества настоящего изобретения станут очевидными из приводимого ниже подробного описания изобретения и прилагаемых чертежей.

Фиг. 1 иллюстрирует блок-схему выполнения заявленного способа.

Фиг. 2 иллюстрирует ROC-кривую (кривая ошибок) для классификатора релизов, основанного на случайном лесе.

Фиг. 3 иллюстрирует матрицу ошибок (без нормализации) для классификатора релизов, основанного на случайном лесе.

Фиг. 4 иллюстрирует пример общего вида вычислительной системы, которая обеспечивает реализацию заявленного решения.

Осуществление изобретения

Ниже будут описаны понятия и термины, необходимые для понимания данного технического решения.

Модель в машинном обучении (МО) - совокупность методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач.

F-1 мера представляет собой совместную оценку точности и полноты.

ROC-кривая - графическая характеристика качества бинарного классификатора, отражающая зависимость доли истинно-положительных классификаций от доли ложно-положительных классификаций при варьировании порога решающего правила.

Матрица ошибок - это способ разбить классифицируемые объекты на четыре категории в зависимости от комбинации реального класса и ответа классификатора.

Коннекторы - программные компоненты, осуществляющие сбор данных от источников информации (Система управления задачами /Система для совместной работы над релизами /Система управления версиями /Система управления проектами /Система управления сервисами предприятия /и др .) и приведение данных к необходимым структуре и формату.

Хранилище - система для хранения больших объемов собранных и обработанных коннекторами данных, а также генерируемой иными компонентами системы.

Данное техническое решение может быть реализовано на компьютере, в виде автоматизированной информационной системы (АИС) или машиночитаемого носителя, содержащего инструкции для выполнения вышеупомянутого способа.

Техническое решение может быть реализовано в виде распределенной компьютерной системы.

В данном решении под системой подразумевается компьютерная система, ЭВМ (электронно-вычислительная машина), ЧПУ (числовое программное управление), ПЛК (программируемый логический контроллер), компьютеризированные системы управления и любые другие устройства, способные выполнять заданную, чётко определённую последовательность вычислительных операций (действий, инструкций).

Под устройством обработки команд подразумевается электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (программы)/

Устройство обработки команд считывает и выполняет машинные инструкции (программы) с одного или более устройства хранения данных, например таких устройств, как оперативно запоминающие устройства (ОЗУ) и/или постоянные запоминающие устройства (ПЗУ). В качестве ПЗУ могут выступать, но, не ограничиваясь, жесткие диски (HDD), флеш-память, твердотельные накопители (SSD), оптические носители данных (CD, DVD, BD, MD и т.п.) и др.

Программа - последовательность инструкций, предназначенных для исполнения устройством управления вычислительной машины или устройством обработки команд.

Обучение модели МО производится на заранее размеченных данных. Всего был доступен на момент создания модели 431 релиз, созданный в заданный временной диапазон, например, 5-6 месяцев. Для оценки качества модели набор данных был разбит на 2 части: тренировочную и контрольную выборки. Разбиение происходило следующим образом (сортировка и выбор для теста последних обращений (тикетов) по дате создания обусловлена особенностями задачи): внутри набора данных релизные тикеты группировались по проектам в системе управления задачами, например, Jira (https://ru.wikipedia.org/wiki/Jira), и сортировались по дате создания; последние 20% внутри каждой группы (по каждому проекту) откладывались для контрольной выборки, если в группе было более 2 тикетов; в контрольную и тестовую выборки добавлялось по одному тикету, если тикетов было 2.

Взвешенная f-1 мера для классификатора составляет около 0,8, точность - около 0,8.

На фиг. 2 приведена ROC-кривая (кривая ошибок) для классификатора релизов, основанного на случайном лесе.

На фиг. 3 приведена матрица ошибок (без нормализации) для классификатора релизов, основанного на случайном лесе

Коннекторы получают необходимую информацию источников (путем загрузки файлов, запросов в БД, к API, анализа web-страниц, чтения журналов событий и т.п.), сохранить ее в хранилище.

Коннекторы выделяют из загруженных данных значимые параметры для дальнейших вычислений, выполняют их предобработку и формируют в хранилище таблицу значений указанных параметров (или признаков) по релизам со следующими столбцами:

Число тикетов типа Bug в релизе;

Число тикетов типа Feature в релизе;

Число тикетов с приоритетом minor в релизе;

Число тикетов с приоритетом major в релизе;

Число тикетов с приоритетом critical в релизе;

Число коммуникаций между членами команды и экспертом кибербезопасности;

Среднее время от создания тикета типа Release до отметки его как решенного;

Среднее время от создания тикета типа Feature до отметки его как решенного;

Среднее время от создания тикета типа Bug до отметки его как решенного;

Число выпущенных продуктовой командой релизов;

Число любых тикетов в одном Epic;

Набор параметров (или признаков), представляющих собой произведения значений всех пар, указанных выше параметров (или признаков);

Количество критичных замечаний по всем релизам продукта в прошлом.

Алгоритм машинного обучения на основе содержащихся в таблице значений параметров релизов осуществляет маркировку присутствующих в ней релизов на несущие в высокой, средней и низкой степени риски кибербезопасности. Результаты маркировки сохраняются в виде таблицы в хранилище.

Способ оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов (100) состоит из нескольких этапов, выполняемых по меньшей мере одним процессором.

На этапе (101) на вход модели машинного обучения подаются данные, содержащие информацию, по меньшей мере, о релизах программных продуктов, находящихся в статусах разработки, предшествующих приемо-сдаточным испытаниям. Также данные могут содержать информацию о: тикетах релизов в системе управления задачами на разработку (номер, тип, важность, статус, время создания, время взятия в работу, время отметки как решенного, список зависимых тикетов и типы зависимостей, список зависящих тикетов и типы зависимостей, ответственная agile-команда, ответственный член команды); количестве критических дефектов не по кибербезопасности, выявленных на приемо-сдаточных испытаниях всех предыдущих релизов программных продуктов; данных об agile-командах и их членах, разрабатывающих релизы: команды, сотрудники-члены команд, их роли в команде, должности, пройденное обучение, сданные экзамены и их результаты, данные о предшествующих переходах сотрудников между agile-командам и изменение должностей, перечень релизов, над которыми работали сотрудники; количестве критических дефектов по кибербезопасности, выявленных на приемосдаточных испытаниях всех предыдущих релизов программных продуктов; данных о документации на релизы (ее объем и иерархия страниц, число попыток и даты ее согласовании экспертами кибербезопасности и иными сотрудниками);

данных об исходном коде релизов (использованные языки, количество модулей, объем кода, количество функций, методов, классов, переменных, файлов); данных о кодировании релизов (число попыток сборки, количество возникавших ошибок и предупреждений при попытках сборки, объем кода, отправляемого на сборку, количество функций, методов, классов, переменных, файлов); данных о тестировании релизов (число попыток прохождения автотестов, нагрузочного и функционального тестирования, объем кода, отправляемого на тестирование, количество функций, методов, классов, переменных, файлов); данных о прохождении проверок системой статического и динамического анализа на предмет наличия уязвимостей в релизах (число и типы обнаруженных уязвимостей, результаты их отметки разработчиками релизов в системе как true-positive/false-positive, объемы кода, отправляемого на сборку, количество функций, методов, классов, переменных, файлов); данных об обнаруженных после вывода в промышленную эксплуатацию уязвимостях в предыдущих релизах программных продуктов (номер релиза, дата обнаружения, создавший уязвимый код разработчик, тип уязвимости, критичность уязвимости, кто обнаружил уязвимость).

Далее на этапе (102) осуществляется обработка полученных данных с помощью модели машинного обучения (МО), например, но не ограничиваясь, с помощью алгоритма машинного обучения на базе классификатора случайного леса (англ. random forest).

В ходе обработки алгоритм машинного обучения выполняет: на этапе (103) разделение импортированных данных на категориальные и численные переменные на этапе (104) преобразование полученных переменных, при котором выполняется векторизация категориальных переменных и нормализация численных переменных; на этапе (105) конкатенацию преобразованных переменных и построение на их основе вектора, соответствующего оцениваемым релизам, находящимся в статусах, предшествующих приемо-сдаточным испытаниям; на этапе (106) классификацию с помощью полученного вектора каждого релиза с присвоением степени вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов. Для соответствующего релизу вектора, вычисляется численная оценка вероятности возникновения дефекта (число от 0 до 1). Далее численной оценке добавляется качественная: высокая/средняя/низкая вероятность (сопоставление происходит в соответствии с диапазонами, например, 0,8-1=высокая).

Заявленное техническое решение обеспечивает новую возможность автоматизированной оценки уровней риска кибербезопасности, порождаемых деятельностью продуктовых agile-команд, и их классификации на соблюдающих в высокой, средней и низкой степени требования по кибербезопасности при разработке ими программных продуктов, позволяет автоматически формировать список релизов, отсортированный по степени порождаемого ими риска кибербезопасности.

Использование такого списка экспертом кибербезопасности для приоритезации своих задач, приводит к выявлению рискованных релизов на более ранних стадиях их создания, экономии трудозатрат экспертов кибербезопасности и членов agile-команд при одновременном снижении уровня рисков кибербезопасности предприятия, порождаемых деятельностью продуктовых agile-команд, снижению времени разработки программных продуктов.

На фиг. 4 представлен пример общего вида вычислительной системы (300), которая обеспечивает реализацию заявленного способа или является частью компьютерной системы, например, сервером, персональным компьютером, частью вычислительного кластера, обрабатывающим необходимые данные для осуществления заявленного технического решения.

В общем случае, система (300) содержит объединенные общей шиной информационного обмена один или несколько процессоров (301), средства памяти, такие как ОЗУ (302) и ПЗУ (303), интерфейсы ввода/вывода (304), устройства ввода/вывода (1105), и устройство для сетевого взаимодействия (306).

Процессор (301) (или несколько процессоров, многоядерный процессор и т.п.) может выбираться из ассортимента устройств, широко применяемых в настоящее время, например, таких производителей, как: Intel™, AMD™, Apple™, Samsung Exynos™, MediaTEK™, Qualcomm Snapdragon™ и т.п. Под процессором или одним из используемых процессоров в системе (300) также необходимо учитывать графический процессор, например, GPU NVIDIA или Graphcore, тип которых также является пригодным для полного или частичного выполнения способа, а также может применяться для обучения и применения моделей машинного обучения в различных информационных системах.

ОЗУ (302) представляет собой оперативную память и предназначено для хранения исполняемых процессором (301) машиночитаемых инструкций для выполнение необходимых операций по логической обработке данных. ОЗУ (302), как правило, содержит исполняемые инструкции операционной системы и соответствующих программных компонент (приложения, программные модули и т.п.). При этом, в качестве ОЗУ (302) может выступать доступный объем памяти графической карты или графического процессора.

ПЗУ (303) представляет собой одно или более устройств постоянного хранения данных, например, жесткий диск (HDD), твердотельный накопитель данных (SSD), флэш-память (EEPROM, NAND и т.п.), оптические носители информации (CD-R/RW, DVD-R/RW, BlueRay Disc, MD) и др.

Для организации работы компонентов системы (300) и организации работы внешних подключаемых устройств применяются различные виды интерфейсов В/В (304).

Выбор соответствующих интерфейсов зависит от конкретного исполнения вычислительного устройства, которые могут представлять собой, не ограничиваясь: PCI, AGP, PS/2, IrDa, FireWire, LPT, COM, SATA, IDE, Lightning, USB (2.0, 3.0, 3.1, micro, mini, type C), TRS/Audio jack (2.5, 3.5, 6.35), HDMI, DVI, VGA, Display Port, RJ45, RS232 и т.п.

Для обеспечения взаимодействия пользователя с вычислительной системой (300) применяются различные средства (305) В/В информации, например, клавиатура, дисплей (монитор), сенсорный дисплей, тач-пад, джойстик, манипулятор мышь, световое перо, стилус, сенсорная панель, трекбол, динамики, микрофон, средства дополненной реальности, оптические сенсоры, планшет, световые индикаторы, проектор, камера, средства биометрической идентификации (сканер сетчатки глаза, сканер отпечатков пальцев, модуль распознавания голоса) и т.п.

Средство сетевого взаимодействия (306) обеспечивает передачу данных посредством внутренней или внешней вычислительной сети, например, Интранет, Интернет, ЛВС и т.п. В качестве одного или более средств (306) может использоваться, но не ограничиваться: Ethernet карта, GSM модем, GPRS модем, LTE модем, 5G модем, модуль спутниковой связи, NFC модуль, Bluetooth и/или BLE модуль, Wi-Fi модуль и др.

Представленные материалы заявки раскрывают предпочтительные примеры реализации технического решения и не должны трактоваться как ограничивающие иные, частные примеры его воплощения, не выходящие за пределы испрашиваемой правовой охраны, которые являются очевидными для специалистов соответствующей области техники.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Компьютерно-реализуемый способ оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов, выполняемый с помощью по меньшей мере одного процессора и содержащий этапы, на которых получают данные, содержащие информацию, по меньшей мере, о релизах программных продуктов, находящихся в статусах разработки, предшествующих приемо-сдаточным испытаниям; осуществляют обработку полученных данных с помощью модели машинного обучения (МО), причем в ходе указанной обработки выполняют разделение импортированных данных на категориальные и численные переменные; преобразование полученных переменных, при котором выполняется векторизация категориальных переменных и нормализация численных переменных; конкатенацию преобразованных переменных и построение на их основе вектора, соответствующего оцениваемым релизам, находящимся в статусах, предшествующих приемо-сдаточным испытаниям; классификацию с помощью полученного вектора каждого релиза с присвоением степени вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов.

2. Способ по п.1, характеризующийся тем, что обработка полученных данных осуществляется с помощью модели машинного обучения на базе классификатора случайного леса (random forest).

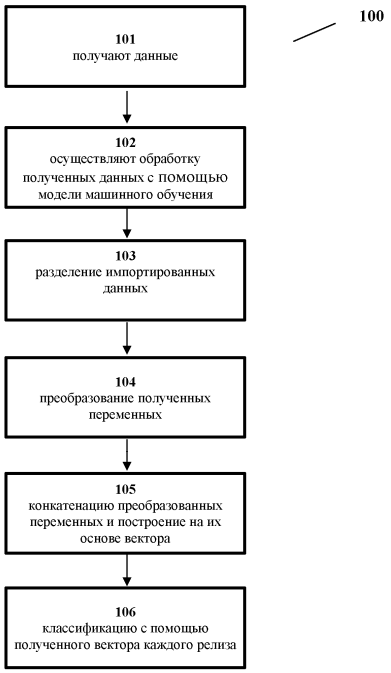
3. Способ по п.1, характеризующийся тем, что модель МО предварительно обучают на исторических данных о получении критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов программных продуктов.

4. Способ по п.1, характеризующийся тем, что обработка полученных данных осуществляется с помощью ансамбля нейронных сетей.

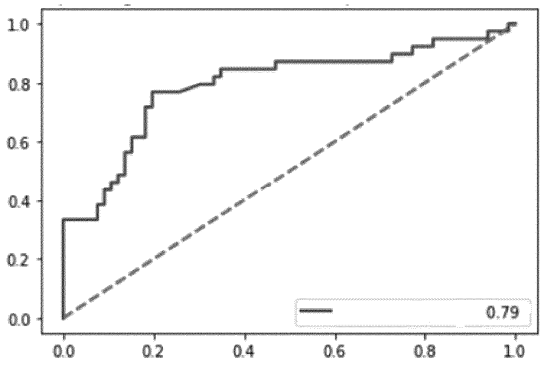
5. Способ по п.1, характеризующийся тем, что получаемые данные дополнительно содержат информацию о записях о задачах категории релиз в системе управления задачами на разработку, включающую в себя номер, тип, важность, статус, время создания, время взятия в работу, время отметки как решенного, список зависимых записей о задачах и типы зависимостей, список зависящих записях о задачах и типы зависимостей.

6. Способ по п.1, характеризующийся тем, что получаемые данные дополнительно содержат информацию о количестве критических дефектов по кибербезопасности, выявленных на приемо-сдаточных испытаниях всех предыдущих релизов программных продуктов.

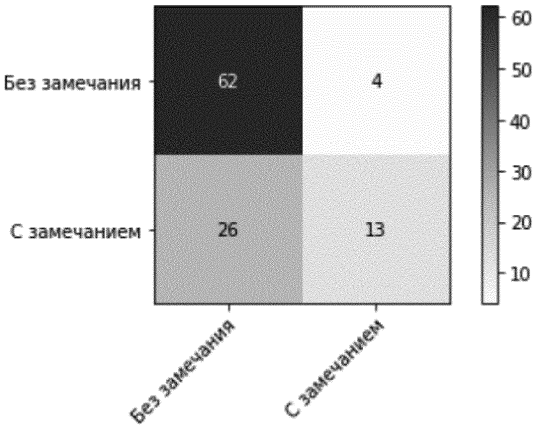
7. Система оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов, содержащая по меньшей мере один процессор; по меньшей мере одну память, соединенную с процессором, которая содержит машиночитаемые инструкции, которые при их выполнении по меньшей мере одним процессором обеспечивают выполнение способа по любому из пп. 1-6.



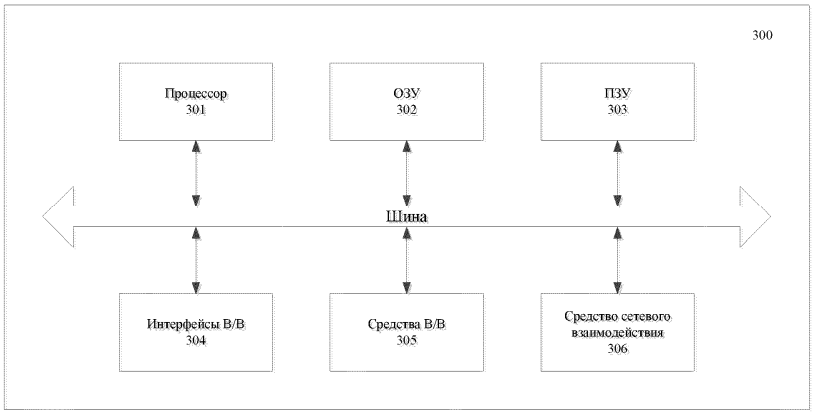
Фиг. 1



Фиг. 2



Фиг. 3


300
Процессор
301
ОЗУ
302
ПЗУ
303
Шина
Интерфейсы В/В
304
Средства В/В
305
Средство сетевого
взаимодействия
306

Фиг. 4