(19) Евразийское патентное ведомство

(11) 038038 (13) B1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2021.06.28

(21) Номер заявки
201892371

(22) Дата подачи заявки
2018.11.19

(51) Int. Cl. ***G06F 9/455*** (2006.01)
G06N 20/00 (2006.01)

(54) СПОСОБ И СИСТЕМА ВЫЯВЛЕНИЯ ЭМУЛИРУЕМОЙ МОБИЛЬНОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

(31) **2018140416**
(32) **2018.11.15**
(33) **RU**
(43) **2020.05.31**

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

(72) Изобретатель:
Балашов Александр Викторович, Давидов Дмитрий Георгиевич (RU)

(74) Представитель:
Герасин Б.В. (RU)

(56) US-A1-20180199199
US-B1-9667613
CN-A-103260173
KR-B1-101530533

(57) Данное техническое решение относится к области вычислительной техники, а в частности к способам и системам выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения, с целью предотвращения возможности запуска приложения, либо для ограничения функционала приложения в эмулируемой среде, либо использования данного признака для увеличения итогового риска транзакции (в системах фрод-мониторинга), выполненных через мобильный банковский клиент. Технический результат - снижение финансового фрода за счет повышения качества анализа транзакций, выполненных в банковских приложениях, запущенных в эмулируемой операционной системе. Указанный результат достигается благодаря способу выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения, в котором осуществляют сбор данных на мобильном приложении, которое установлено на мобильное устройство связи пользователя; направляют полученные на предыдущем шаге данные из мобильного устройства связи пользователя на сервер; осуществляют выделение полученных данных посредством синтаксического анализатора, содержащих собранные признаки, в соответствии с наименованием полей; направляют выделенные признаки в обученную статистическую модель данных, расположенную на сервере, содержащую группы признаков, которые включают группу статистических признаков, динамических признаков, пользовательских метрик, файловых метрик и модифицированных метрик; принимают решение посредством обученной статистической модели данных установить мобильное приложение на реальном устройстве или эмулируемой мобильной операционной системе.

038038 B1

038038 B1

Область техники

Данное техническое решение относится к области вычислительной техники, а в частности к способам и системам выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения, с целью предотвращения возможности запуска приложения, либо для ограничения функционала мобильного приложения в эмулируемой среде.

Уровень техники

В настоящее время использование смартфонов в повседневной жизни не ограничивается голосовыми звонками и СМС. Возможность загружать и выполнять программы, а также мобильный доступ в Интернет привели к появлению большого количества мобильных приложений. Функциональность современного смартфона составляют браузеры, клиентские программы социальных сетей, офисные приложения и всевозможные сервисы, работающие в Сети.

В настоящий момент возможен беспрепятственный запуск мобильного приложения, например, банковские клиенты, на множестве программных эмуляторов операционной системы Android. Злоумышленники используют эмулируемые устройства для совершения мошеннических действий, т.к. в этом случае не требуется покупка нового телефона для совершения очередной попытки мошенничества, достаточно переустановить эмулятор, либо изменить ряд параметров в конфигурационных файлах.

Как раскрыто в статьях [1, 2], существуют способы определения эмулятора, основанные на данных, получаемых от датчиков оборудования, таких как акселерометр, гироскоп, GPS, датчик света, силы тяжести. Выходные значения этих датчиков основаны на информации, собранной из окружающей среды, и, следовательно, реалистичная их эмуляция является сложной задачей. Присутствие такого рода датчиков - основное различие между смартфонами и настольными компьютерами. Увеличивающееся число датчиков на смартфонах дает новые возможности для идентификации мобильных устройств.

Однако данные технологии могут не сработать или привести к неверным результатам, если вредоносная часть кода программы не присутствует в антивирусных базах или код содержит скрытое и неочевидное вредоносное поведение. В этих случаях скрытый код может быть отправлен аналитику для рассмотрения и принятия решения о вредоносности программы. Процесс просмотра аналитиком дает точные результаты, но при этом занимает время и не эффективен с данной точки зрения.

Анализ предшествующего уровня техники позволяет сделать вывод о неэффективности и в некоторых случаях о невозможности применения предшествующих технологий, недостатки которых решаются настоящим изобретением.

Сущность технического решения

Технической проблемой или технической задачей, решаемой в данном техническом решении, является выявление эмулируемой мобильной операционной системы с использованием методов машинного обучения. Техническим результатом, достигаемым при решении вышеуказанной задачи, является повышение точности выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения. Дополнительно проявляющимся техническим результатом является предотвращение возможности многократно выполнять "первый" заказ в мобильных приложениях. Многие приложения предоставляют скидки/бонусы/баллы за начало использование их сервиса на первый заказ услуги/продукта. Эмулируемая операционная система позволяет многократно выполнять "первый" заказ одним человеком.

Указанный технический результат достигается благодаря осуществлению способа выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения, в котором осуществляют сбор данных на мобильном приложении, которое установлено на мобильное устройство связи пользователя; направляют полученные на предыдущем шаге данные из мобильного устройства связи пользователя на сервер; осуществляет выделение полученных данных посредством синтаксического анализатора, содержащих собранные признаки, в соответствии с наименованием полей; направляют выделенные признаки в обученную статистическую модель данных, расположенную на сервере, содержащую группы признаков, которые включают группу статистических признаков, динамических признаков, пользовательских метрик, файловых метрик и модифицированных метрик; принимают решение посредством обученной статистической модели данных установлено мобильное приложение на реальном устройстве или эмулируемой мобильной операционной системе.

В некоторых вариантах реализации технического решения сбор данных на мобильном приложении осуществляются в JSON-файл и/или XML, и/или YML, и/или TXT.

В некоторых вариантах реализации технического решения сервер располагается в облачном хранилище данных или локально.

В некоторых вариантах реализации технического решения статистической моделью является логистическая регрессия, или дерево решений, или случайный лес, или наивный байесовский классификатор, или метод опорных векторов.

В некоторых вариантах реализации технического решения для определения эффективности статистической модели используется кросс-валидация.

В некоторых вариантах реализации технического решения синтаксический анализатор осуществляет разбор элементов, содержащих собранные признаки, в соответствии с наименованием их полей.

В некоторых вариантах реализации технического решения синтаксический анализатор после разбора файла осуществляет маппинг признаков в поля таблицы в базу данных.

В некоторых вариантах реализации технического решения маппинг данных в поля таблицы в базу данных происходит путем выполнения INSERT-запроса.

В некоторых вариантах реализации технического решения при выделении посредством синтаксического анализатора признаков делят их на статические и динамические.

В некоторых вариантах реализации технического решения к динамическим данным относятся показания акселерометра и/или гироскопа, и/или датчика освещенности, и/или датчика магнитного поля, и/или датчика движения, и/или датчика расстояния, и/или датчика вращения.

Каждый вариант осуществления настоящей технологии включает по меньшей мере одну из вышеупомянутых целей и/или объектов, но наличие всех не является обязательным. Следует иметь в виду, что некоторые объекты данной технологии, полученные в результате попыток достичь вышеупомянутой цели, могут не удовлетворять этой цели и/или могут удовлетворять другим целям, отдельно не указанным здесь.

Дополнительные и/или альтернативные характеристики, аспекты и преимущества вариантов осуществления настоящей технологии станут очевидными из последующего описания, прилагаемых чертежей и прилагаемой формулы технологии.

Краткое описание чертежей

Признаки и преимущества настоящего технического решения станут очевидными из приводимого ниже подробного описания и прилагаемых чертежей, на которых

На фиг. 1 показан вариант реализации архитектуры системы выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения;

на фиг. 2 показана структура приложения для сбора данных с устройства;

на фиг. 3 показан пример коэффициентов для модели логистической регрессии;

на фиг. 4 показан пример значимости коэффициентов для статистической модели случайного леса;

на фиг. 5 показан объем потребляемого заряда батареи в зависимости от используемых функций устройства;

на фиг. 6 показан вариант реализации разбиения набора данных и построение модели на одной части данных;

на фиг. 7 показан вариант реализации дерева решений;

на фиг. 8 показан вариант реализации системы выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения;

на фиг. 9 показан вариант реализации системы выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения в виде блок-схемы.

Подробное описание

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, которая может быть установлена на централизованном сервере (наборе серверов). В некоторых вариантах реализации модель так же может функционировать локально на телефоне в рамках мобильного приложения. Доступ пользователей к системе возможен как из сети Интернет, так и из внутренней сети предприятия/организации посредством мобильного устройства связи, на котором установлено программное обеспечение с соответствующим графическим интерфейсом пользователя, или персонального компьютера с доступом к веб-версии системы с соответствующим графическим интерфейсом пользователя.

Ниже будут описаны термины и понятия, необходимые для реализации настоящего технического решения.

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. В роли устройства хранения данных могут выступать, но, не ограничиваясь, жесткие диски (HDD), флеш-память, ПЗУ (постоянное запоминающее устройство), твердотельные накопители (SSD), оптические приводы.

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

В контексте настоящего описания, если четко не указано иное, "сервер" подразумевает под собой компьютерную программу, работающую на соответствующем оборудовании, которая способна получать

запросы (например, от клиентских устройств) по сети и выполнять эти запросы или инициировать выполнение этих запросов. Оборудование может представлять собой один физический компьютер или одну физическую компьютерную систему, но ни то, ни другое не является обязательным для данной технологии. В контексте настоящей технологии использование выражения "сервер" не означает, что каждая задача (например, полученные инструкции или запросы) или какая-либо конкретная задача будет получена, выполнена или инициирована к выполнению одним и тем же сервером (то есть одним и тем же программным обеспечением и/или аппаратным обеспечением); это означает, что любое количество элементов программного обеспечения или аппаратных устройств может быть вовлечено в прием/передачу, выполнение или инициирование выполнения любого запроса или последствия любого запроса, связанного с клиентским устройством, и все это программное и аппаратное обеспечение может являться одним сервером или несколькими серверами, оба варианта включены в выражение "по меньшей мере один сервер". В контексте настоящего описания, если четко не указано иное, "клиентское устройство" или "мобильное устройство связи пользователя" подразумевает под собой аппаратное устройство, способное работать с программным обеспечением, подходящим к решению соответствующей задачи. Таким образом, примерами клиентских устройств (среди прочего) могут служить персональные компьютеры (настольные компьютеры, ноутбуки, нетбуки и т.п.), смартфоны, планшеты, а также сетевое оборудование, такое как маршрутизаторы, коммутаторы и шлюзы. Следует иметь в виду, что устройство, ведущее себя как клиентское устройство в настоящем контексте, может вести себя как сервер по отношению к другим клиентским устройствам. Использование выражения "клиентское устройство" не исключает возможности использования множества клиентских устройств для получения/отправки, выполнения или инициирования выполнения любой задачи или запроса, или же последствий любой задачи или запроса, или же этапов любого вышеописанного метода.

В контексте настоящего описания, если четко не указано иное, термин "база данных" подразумевает под собой любой структурированный набор данных, не зависящий от конкретной структуры, программного обеспечения по управлению базой данных, аппаратного обеспечения компьютера, на котором данные хранятся, используются или иным образом оказываются доступны для использования. База данных может находиться на том же оборудовании, которое выполняет процесс, который сохраняет или использует информацию, хранящуюся в базе данных, или же она может находиться на отдельном оборудовании, например, выделенном сервере или множестве серверов.

В контексте настоящего описания, если четко не указано иное, термин "информация" включает в себя любую информацию, которая может храниться в базе данных. Таким образом, информация включает в себя, среди прочего, аудиовизуальные произведения (изображения, видео, звукозаписи, презентации и т.д.), данные (данные о местоположении, цифровые данные и т.д.), текст (мнения, комментарии, вопросы, сообщения и т.д.), документы, таблицы и т.д.

В контексте настоящего описания, если четко не указано иное, термин "компонент" подразумевает под собой программное обеспечение (соответствующее конкретному аппаратному контексту), которое является необходимым и достаточным для выполнения конкретной (ых) указанной (ых) функции(й).

В контексте настоящего описания, если четко не указано иное, термин "используемый компьютером носитель компьютерной информации" подразумевает под собой носитель абсолютно любого типа и характера, включая ОЗУ, ПЗУ, диски (компакт диски, DVD-диски, дискеты, жесткие диски и т.д.), USB флеш-накопители, твердотельные накопители, накопители на магнитной ленте и т.д.

В контексте настоящего описания, если четко не указано иное, слова "первый", "второй", "третий" и т.д. используются в виде прилагательных исключительно для того, чтобы отличать существительные, к которым они относятся, друг от друга, а не для целей описания какой-либо конкретной связи между этими существительными. Так, например, следует иметь в виду, что использование терминов "первый сервер" и "третий сервер" не подразумевает какого-либо порядка, отнесения к определенному типу, хронологии, иерархии или ранжирования (например) серверов/между серверами, равно как и их использование (само по себе) не предполагает, что некий "второй сервер" обязательно должен существовать в той или иной ситуации.

Перечень ключевых компонент системы выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения, представленной на фиг. 8, включает в себя нижеуказанные элементы.

Сборщиком данных является мобильное приложение, которое устанавливается на устройство под управлением ОС Android и выполняет сбор данных, как показано на фиг. 2, которые представляют собой различные метрики устройства, подробно описанные ниже. В некоторых вариантах реализации получаемые данные делятся на группы признаков, которые позволяют в дальнейшем более точно классифицировать принадлежность признака к той или иной группе.

Группа статистических признаков может включать наличие Bluetooth модуля и наличие Vibracall модуля в мобильном устройстве связи пользователя. Группа динамических признаков может включать показания акселерометра, гироскопа, датчика освещения, датчика линейного ускорения, уровень заряда батареи, показания датчика магнитного поля, показания датчика движения, показания датчика расстояния, показания датчика вращения. Группа пользовательских метрик может включать такие признаки, как

количество SMS/MMS, количество контактов, количество выполненных звонков, количество фотографий (пользовательских файлов), история в браузере и т.д. Группа файловых метрик может включать, например, наличие файла наличие токена "network_thoughput" в файле /proc/mosc, наличие токена "Off\0:" в файле /proc/ioports, наличие файла /proc/uid_stat, наличие токена "MINOR=5" в файле /sys/device/virtual/misc/cpu_dma_latency/uevent. Группа модифицированных метрик, включающая набор различных токенов в параметрах, например, токен "userdebug" в параметре Build.DISPLAY, токены "userdebug", "vbox86" или "remix" в параметре Build.FINGERPRINT и т.д.

Парсер текста или синтаксический анализатор находится на серверной части системы, осуществляет разбор строки данных и маппинг в базу данных полей из JSON файла, в котором могут находиться признаки, в поля БД. В качестве парсера JSON может быть использован любой Open Source продукт, решающий эту задачу, или парсер собственной разработки. Передача данных средствами JSON-файла не является единственным возможным и ограничивающим вариантом реализации. Так же для передачи данных могут использоваться прямые запросы (команда Insert) в БД, либо осуществляться передача данных путем формирования и передачи XML, YML, TXT и других форматов текстовых файлов, не ограничиваясь. База данных находится на серверной части, осуществляет долгосрочное хранение данных. База данных используется в процессе исследования механизмов ОС (например, Android) для сбора и хранения списка всех метрик, потенциально свидетельствующих о наличии эмулируемой среды. База данных хранит итоговый список метрик.

Статистической моделью, которая находятся на серверной части, может быть логистическая регрессия, или дерево решений, или случайный лес, или наивный байесовский классификатор, или метод опорных векторов и другие алгоритмы машинного обучения. Данная статистическая модель осуществляет обработку и анализ данных с последующим формированием выходного результата (является ли устройство эмулятором, либо реальным устройством). Предварительно происходит формирование обучающей выборки и ее разметка. В некоторых вариантах реализации дополнительно с обучающей выборкой используют тестовую выборку для контролирования качества обучения. Для обучения статистической модели данные собираются из множества реальных устройств и множества доступных в настоящий момент эмуляторов. Затем происходит процесс обучения модели. Например, для дерева решений процесс обучения заключается в построении оптимального дерева, путем поиска признаков, наилучшим образом разделяющих всю выборку на два класса - эмуляторы и реальные устройства. В конкретном варианте реализации данного технического решения необходимо ответить от одного до трех вопросов, как показано на фиг. 7, чтобы однозначно сказать, используется эмулятор или реальное устройство. В качестве примерного варианта реализации могут использоваться следующие вопросы:

есть ли файл uid_stat в директории /proc/?

есть ли файл switch в директории /sys/device/virtual/?

изменились ли значения датчика гироскопа с момента последнего сбора данных?

На следующем шаге проверяется эффективность полученной модели на различных наборах данных. В некоторых вариантах реализации для этого используется кросс-валидация, согласно которой происходит разбивка набора данных и построение статистической модели на одной части данных (называемой тренировочным набором или обучающей выборкой), затем происходит валидация результатов работы модели на другой части (называемой тестовым набором или тестовой выборкой). Таким образом в примерном варианте реализации было выполнено 20 циклов кросс-валидации. Чтобы уменьшить разброс результатов, разные циклы кросс-валидации проводились на разных разбиениях, как показано на фиг. 6. Результаты по каждому циклу кросс-валидации представлены ниже. В качестве примера реализации ниже описан алгоритм работы одной из статистических моделей, которая может использоваться для выявления эмулируемой среды. Метод дерево решений (англ. "DecisionTree") является одним из наиболее популярных способов решения задач классификации и прогнозирования в области машинного обучения. Итак, в конкретном варианте реализации, как показано на фиг. 7, основными элементами дерева решений являются:

Корнем дерева решений является признак "f3" - наличие файла /proc/uid_stat. Внутренним узлом дерева или узлом проверки является признак "f6" - наличие файла /sys/device/virtual/switch. Листом (конечным узлом дерева) является узел решения или вершина "эмулятор", "реальное устройство". Ветвью дерева (случаи ответа) может быть "Да" или "Нет".

Для формирования обучающей выборки осуществляют подготовку инфраструктуры, включающую работу сборщика данных, синтаксического анализатора JSON-файла и БД. Сборщик данных реализован в виде мобильного приложения, которое устанавливается на множество реальных устройств или в эмулируемую операционную систему и с заданным интервалом времени (например, один раз в 5 с) собирает данные по исследуемым метрикам. Синтаксический анализатор JSON-файла и БД устанавливаются на отдельный сервер, который может располагаться, как в облачном хранилище данных (в конкретном примере реализации в MS Azure), так и локально. Синтаксический анализатор осуществляет разбор элементов JSON-файла, содержащих собранные признаки, в соответствии с наименованием их полей. Затем происходит маппинг в одноименные с JSON-файлом поля в БД. Нижеприведенный пример показывает возможное содержимое JSON-файла, для случая использования метода деревьев решении, описывающе-

го эмулируемую операционную систему:

"f3": "0",

"f6": "0",

"gyroscope_value_m": "0".

Алгоритм синтаксического разбора такого JSON-файла включает следующие шаги:

1) чтение JSON-файла;

2) поиск первого токена - f3 и запись данных в переменную A;

3) поиск второго токена - f6 и запись данных в переменную B;

4) поиск третьего токена - gyroscope_value_m и запись данных в переменную C.

Затем осуществляется маппинг значений переменных A, B и C в поля таблицы в БД для долгосрочного хранения и передача значений переменных в модель для анализа и принятия решения. Маппинг данных в поля таблицы в БД происходит путем выполнения INSERT-запроса.

В данном техническом решении для осуществления изобретения был собран набор данных, включивший в себя данные по 125 признакам, собранным с 631 устройства (246 - эмуляторов, 385 - реальных телефонов). После очистки и обработки данных были выбраны наиболее значимые признаки, которые использовались для обучения финальной статистической модели. Значимость признака определяется на основании эффективности разделения набора данных, т.е. каждый из 125 признаков последовательно проверяется на способность достоверно разделить сформированный набор данных на множество данных с реальных устройств и множество данных с эмуляторов.

Для сбора данных были использованы различные конфигурации следующих эмуляторов, приведенных в качестве примера, а не ограничения: Andy OS, BlueStacks, Genymotion, KO Player, LeapDroid, MEmu и т.д.

А также были использованы следующие модели реальных мобильных устройств связи, не ограничиваясь: Samsung Galaxy S4, Samsung Galaxy A5, Motorola XT1541, Lenovo A600 Plus.

После проведения анализа собранных данных и отбора наиболее значимых параметров, были выделены следующие метрики, не требующие дополнительных модификаций: наличие и доступность модуля Bluetooth, наличие функции виброзвонка, наличие токена "network_thoughput" в файле /proc/mosc, наличие токена "Off\0:" в файле /proc/ioports, наличие файла /proc/uid_stat, наличие токена "MINOR=5" в файле /sys/device/virtual/misc/cpu_dma_latency/uevent, наличие файла /sys/device/virtual/ppp, наличие файла /sys/device/virtual/switch, наличие файла /sys/module/alarm/parameters, наличие файла /sys/device/system/cpu/cpuO/cpufreq, наличие файла /sys/device/virtual/misc/android_adb, наличие файла /proc/sys/net/ipv4/tcp_syncookies и т.д., не ограничиваясь.

Отбор значимых признаков происходит после формирования набора данных. Отбор значимых признаков происходит путем анализа собранных данных и выбора признаков, позволяющих максимально эффективно разделить множество устройств на реальные устройства и эмуляторы. Анализ заключается в подсчете показателя эффективности разделения множества устройств на эмуляторы и реальные устройства по каждому из 125 признаков последовательно. Затем происходит оценка возможности модификации признака и проводится повторный анализ аналогичным образом. Оценка возможности модификации признака носит аналитический характер и осуществляется на основе данных, характерных для конкретного признака. Например, данными возвращаемыми функцией telephonyManager.getAIICelllnfo(), является список идентификаторов GSM, LTE и WCDMA вышек. В исходном виде эти данные не являются эффективным признаком для разделения всего набора данных на две группы: 1 - данные с эмуляторов, 2 - данные с реальных устройств. Но после анализа этих данных, а именно подсчета количества каждого типа вышек в данных с эмуляторов и в данных с реальных устройств удается эффективно разделить весь набор данных по следующему критерию: наличие вышек типов LTE и WCDMA с высокой вероятностью свидетельствует о том, что данные получены с реального устройства. На основе этой аналитики были сформированы два модифицированных признака: n_cells_lte - количество вышек LTE, n_cells_wcdma - количество вышек WCDMA в выводе функции telephonyManager.getAIICelllnfo().

В отдельную группу вошли показания различных датчиков. Из-за невысокой точности получаемых с них показаний в подавляющем большинстве случаев значения, взятые с интервалом хотя бы в 5 с, различаются, даже если сами устройства находятся в неподвижном состоянии. Эти различия, как правило, невелики, но их может быть достаточно, чтобы отличить реальное устройство от эмулятора, где данные статичны, пока не будут изменены вручную. Для проверки корректности данного предположения были добавлены модифицированные признаки, сформированные на основе показаний с датчиков. Значение признака по конкретному датчику принимало значение 1, если показания датчика изменились с последнего измерения, и 0, если значение не изменилось.

Затем по этим признакам был выполнен повторный анализ на эффективность разделения всего набора данных на две группы - данные с эмуляторов и данные с реальных устройств. Некоторые признаки успешно разделили набор данных и были использованы в моделях. Например, в вышеописанной модели "дерево решений" успешно используется признак gyroscope_value_m.

К динамическим данным относятся показания со следующих датчиков: акселерометр, гироскоп, датчик освещенности, датчик магнитного поля, датчик движения, датчик расстояния, датчик вращения и

другие.

Так как любые манипуляции с датчиками устройства подразумевают повышенный расход батареи устройства, предварительно используют только статические признаки, а затем добавляют для использования динамические признаки. Для тестирования в конкретном примере реализации были выбраны пять классификаторов: логистическая регрессия, или дерево решений, или случайный лес, или наивный байесовский классификатор, или метод опорных векторов, а также другие алгоритмы машинного обучения.

При анализе статических показателей в набор данных вошли 25 признаков, перечисленных выше. Ниже приведены данные по 20 итерациям кросс - валидации на основе StratifiedKFold и метрики roc_auc (метрика, оценивающая качество работы классификатора, вычисляющего вероятность принадлежности объекта к положительному классу).

Номер разбиения	Логистическая регрессия	Дерево решений	Случайный лес	Наивный байесовский классификатор	Метод опорных векторов
1	1.0	0.96	1.0	0.99	1.0
2	1.0	1.0	1.0	1.0	1.0
3	1.0	0.98	1.0	0.98	1.0
4	1.0	0.99	1.0	0.99	1.0
5	1.0	0.97	1.0	0.97	1.0
6	1.0	1.0	1.0	1.0	1.0
7	0.99	0.99	1.0	0.99	1.0
8	1.0	0.99	1.0	0.99	1.0
9	1.0	0.99	1.0	0.99	1.0
10	1.0	0.98	1.0	0.98	1.0
11	1.0	1.0	1.0	1.0	1.0
12	1.0	0.99	1.0	1.0	1.0
13	1.0	0.97	1.0	0.97	1.0
14	0.99	0.98	1.0	0.98	1.0
15	1.0	0.99	1.0	1.0	1.0
16	1.0	1.0	1.0	1.0	1.0
17	0.99	0.99	1.0	0.99	1.0
18	1.0	0.98	1.0	0.98	1.0
19	1.0	0.98	1.0	0.98	1.0
20	1.0	1.0	1.0	1.0	1.0

В таблице представлены показатели эффективности результатов работы моделей без использования данных с датчиков.

StratifiedKFold - класс библиотеки scikit-learn, позволяющий выполнить разделение набора данных таким образом, чтобы в полученных наборах соотношение количеств объектов, относящихся к разным классам (эмуляторы и реальные устройства), соответствовало этому соотношению для исходного набора данных.

При добавлении к набору данных динамических метрик общее число рассматриваемых признаков увеличилось до 35. Оценка моделей проводилась аналогичным описанному выше способом.

Номер разбиения	Логистическая регрессия	Дерево решений	Случайный лес	Наивный байесовский классификатор	Метод опорных векторов
1	0.99	0.98	1.0	0.98	1.0
2	0.99	0.96	1.0	0.99	1.0
3	0.99	0.98	1.0	0.98	1.0
4	0.99	0.98	1.0	0.98	1.0
5	1.0	0.99	1.0	0.99	1.0
6	0.99	0.99	1.0	1.0	1.0
7	0.99	0.99	1.0	1.0	1.0
8	0.99	0.99	1.0	0.99	1.0
9	0.99	0.99	1.0	0.99	1.0
10	1.0	0.99	1.0	1.0	1.0
11	0.99	0.99	1.0	0.99	1.0
12	1.0	0.99	1.0	0.99	1.0
13	1.0	0.99	1.0	1.0	1.0
14	1.0	0.95	1.0	1.0	1.0
15	0.99	0.99	1.0	0.99	1.0
16	1.0	0.95	1.0	1.0	1.0
17	0.99	0.99	1.0	1.0	1.0
18	1.0	0.99	1.0	0.99	1.0
19	0.99	0.99	1.0	1.0	1.0
20	0.99	0.99	1.0	1.0	1.0

Как видно из приведенных данных, отобранные признаки позволили построить статистические модели, демонстрирующие близкие к идеальным результатам. По результатам проведенных тестов наилучшие показатели демонстрируют случай лес, метод опорных векторов и логистическая регрессия, достигая на имеющихся данных вероятности выявления эмулируемой среды близкой к 1. На фиг. 3-4 показаны примеры коэффициентов для этих моделей. Все рассмотренные статистические модели стабильно демонстрируют высокую точность даже без использования данных с датчиков. Последние хоть и увели-

чивают точность отдельных моделей, но, как правило, требуют большего расхода заряда батареи устройства, что накладывает свои ограничения. На фиг. 5 представлена информацию по потреблению энергии, в зависимости от используемых функций мобильного устройства связи пользователя. Процесс принятия решения состоит из следующих шагов:

сбор данных;

отправка данных в статистическую модель для анализа;

формирование результата моделью;

ответ мобильному приложению.

После сбора данных и работы синтаксического анализатора, соответствующие поля передаются в статистическую модель, где подвергаются анализу. Пример процесса анализа описан выше на примере метода деревьев решений. Результатом работы такой модели является вердикт, однозначно определяющий на реальном устройстве или в эмулируемой операционной системе запущено мобильное приложение. Результат работы статистической модели может передаваться в дальнейшем в систему фрод-мониторинга для последующей обработки, либо мобильному приложению для ограничения функционала или возможности запуска. Процесс передачи результата в приложение может осуществляться посредством одного из известных в уровне техники программных интерфейсов для обеспечения обмена данными между процессами, например, Socket, и/или WebSocket, и/или WCF, и/или REST и т.д.

Например, приложение может выполнять проверку операционной системы на предмет эмулируемой среды в момент запуска, либо при наступлении определенных событий - регистрация нового пользователя, смена пароля и другие. Архитектура всей системы выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения показана на фиг. 8.

На фиг. 9 представлен пример вычислительного устройства 800, которое может применяться для выполнения функций по логической обработке необходимых данных для реализации заявленного способа выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения. Необходимо отметить, что указанные в материалах настоящей заявки компьютерное устройство пользователя и сервер, могут представлять собой один из вариантов воплощения вычислительного устройства 800.

В общем случае, вычислительное устройство 800 содержит объединенные общей шиной 810 один или несколько процессоров 801, средства памяти, такие как ОЗУ 802 и ПЗУ 803, интерфейсы ввода/вывода 804, средства ввода/вывода 805 и средство для сетевого взаимодействия 806.

Процессор 801 (или несколько процессоров, многоядерный процессор) может выбираться из ассортимента устройств, широко применяемых в текущее время, например, компаний Intel™, AMD™, Apple™, Samsung Exynos™, MediaTEK™, Qualcomm Snapdragon™ и т.п.

ОЗУ 802 представляет собой оперативную память и предназначено для хранения исполняемых процессором 801 машиночитаемых инструкций, для выполнения необходимых операций по логической обработке данных. ОЗУ 802, как правило, содержит исполняемые инструкции операционной системы и соответствующих программных компонент (приложения, программные модули и т.п.).

ПЗУ 803 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 устройства 800 и организации работы внешних подключаемых устройств применяются различные виды интерфейсов В/В 804. 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 вычислительным устройством 800 применяются различные средства 805 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 806 обеспечивает передачу данных устройством 800 посредством внутренней или внешней вычислительной сети, например, Интранет, Интернет, ЛВС и т.п. В качестве одного или более средств 806 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.

Дополнительно могут применяться также средства спутниковой навигации, например, как GPS, ГЛОНАСС, BeiDou, Galileo и т.д.

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 специа-

листов соответствующей области техники.

Следует принять во внимание, что настоящее изобретение не ограничивается точной конструкцией или реализацией, которая была описана выше и проиллюстрирована на прилагаемых чертежах, и различные модификации и изменения могут быть выполнены без отступления от его объема. Предполагается, что объем настоящего изобретения ограничивается только прилагаемой формулой изобретения.

Используемые источники информации

1. Petsas T. et al. Rage against the virtual machine: hindering dynamic analysis of android malware // Proceedings of the Seventh European Workshop on System Security. ACM, 2014. P. 5.

2. Jing Y. et al. Morpheus: automatically generating heuristics to detect Android emulators // Proceedings of the 30th Annual Computer Security Applications Conference. ACM, 2014. P. 216—225.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения, включающий следующие шаги:

осуществляют сбор данных на мобильном приложении, которое установлено на мобильное устройство связи пользователя;

направляют полученные на предыдущем шаге данные из мобильного устройства связи пользователя на сервер;

осуществляют выделение полученных данных посредством синтаксического анализатора, содержащих собранные признаки, в соответствии с наименованием полей;

направляют выделенные признаки в обученную статистическую модель данных, расположенную на сервере, содержащую группы признаков, которые включают группу статистических признаков, включающих наличие Bluetooth модуля и наличие Vibracall модуля (виброзвонка) в мобильном устройстве связи пользователя, динамических признаков, включающих показания акселерометра, гироскопа, датчика освещения, датчика линейного ускорения, уровень заряда батареи, показания датчика магнитного поля, показания датчика движения, показания датчика расстояния, показания датчика вращения, пользовательских метрик, файловых метрик и модифицированных метрик, включающих набор токенов в параметрах мобильного устройства связи пользователя;

принимают решение посредством обученной статистической модели данных установить мобильное приложение на реальном устройстве или в эмулируемой мобильной операционной системе.

2. Способ по п.1, характеризующийся тем, что сбор данных на мобильном приложении осуществляется в JSON-файл и/или XML, и/или YML, и/или TXT.

3. Способ по п.1, характеризующийся тем, что сервер располагается в облачном хранилище данных или локально.

4. Способ по п.1, характеризующийся тем, что статистической моделью является логистическая регрессия, или дерево решений, или случайный лес, или наивный байесовский классификатор, или метод опорных векторов.

5. Способ по п.1, характеризующийся тем, что для определения эффективности статистической модели используется кросс-валидация.

6. Способ по п.1, характеризующийся тем, что синтаксический анализатор осуществляет разбор элементов, содержащих собранные признаки, в соответствии с наименованием их полей.

7. Способ по п.1, характеризующийся тем, что синтаксический анализатор после разбора файла осуществляет маппинг признаков в поля таблицы в базу данных.

8. Способ по п.7, характеризующийся тем, что маппинг данных в поля таблицы в базу данных происходит путем выполнения INSERT-запроса.

9. Способ по п.1, характеризующийся тем, что при выделении посредством синтаксического анализатора признаков делят их на статические и динамические.

10. Способ по п.1, характеризующийся тем, что к динамическим данным относятся показания акселерометра и/или гироскопа, и/или датчика освещенности, и/или датчика магнитного поля, и/или датчика движения, и/или датчика расстояния, и/или датчика вращения.

11. Система выявления эмулируемой мобильной операционной системы с использованием методов машинного обучения, содержащая

по меньшей мере одно мобильное устройство связи пользователя, на которое установлено мобильное приложение, выполненное с возможностью осуществления сбора данных;

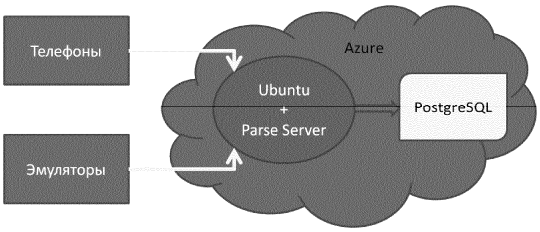
направление полученных данных из мобильного устройства связи пользователя на по меньшей мере один сервер;

по меньшей мере один сервер, выполненный с возможностью осуществления выделения полученных данных из мобильного устройства связи пользователя посредством синтаксического анализатора,

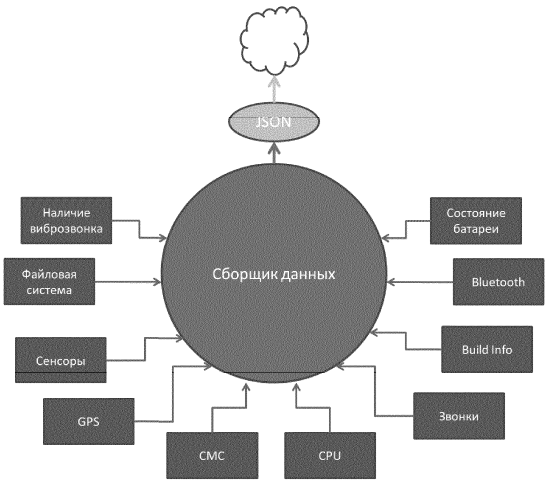
причем данные содержат собранные признаки, в соответствии с наименованием полей;

направление синтаксическим анализатором выделенных признаков в обученную статистическую модель данных, содержащую группы признаков, которые включают группу статистических признаков, включающих наличие Bluetooth модуля и наличие Vibracall модуля (виброзвонка) в мобильном устройстве связи пользователя, динамических признаков, включающих показания акселерометра, гироскопа, датчика освещения, датчика линейного ускорения, уровень заряда батареи, показания датчика магнитного поля, показания датчика движения, показания датчика расстояния, показания датчика вращения, пользовательских метрик, файловых метрик и модифицированных метрик, включающих набор токенов в параметрах мобильного устройства связи пользователя;

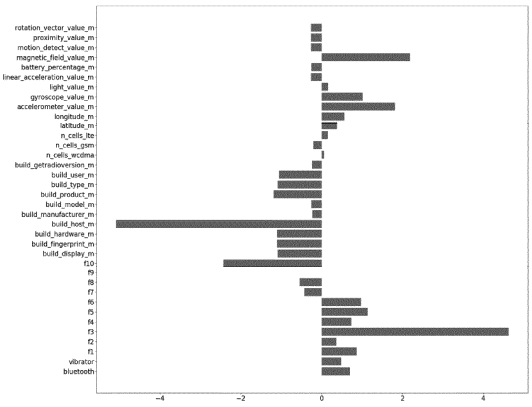
принятие решения посредством обученной статистической модели данных установить мобильное приложение на реальном устройстве или эмулируемой мобильной операционной системе.



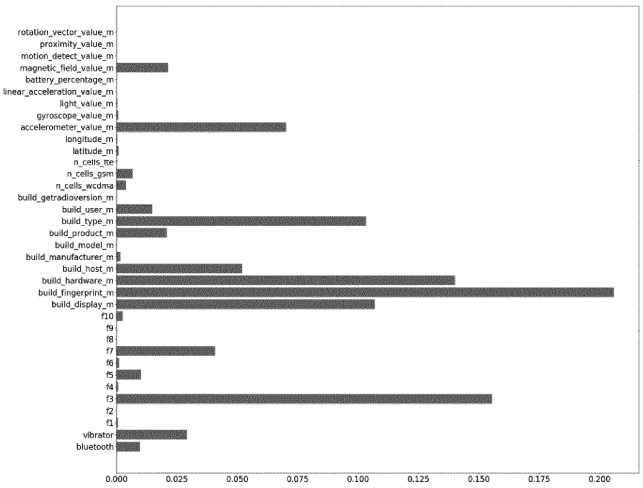
Фиг. 1



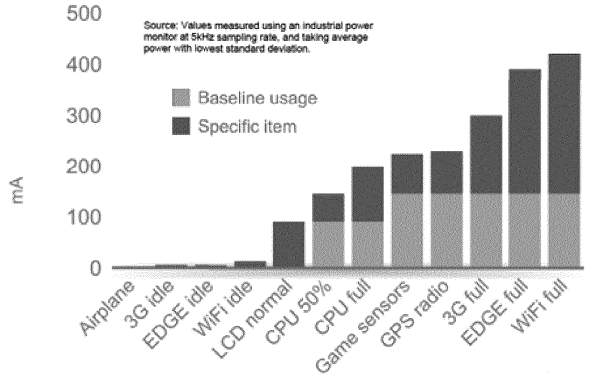
Фиг. 2



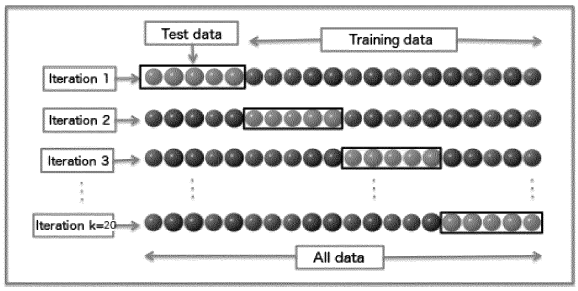
Фис. 3



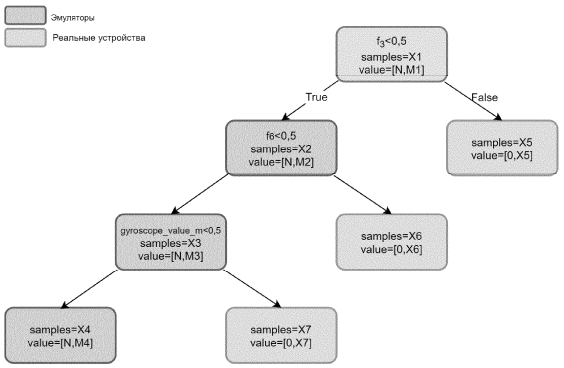
Фиг. 4



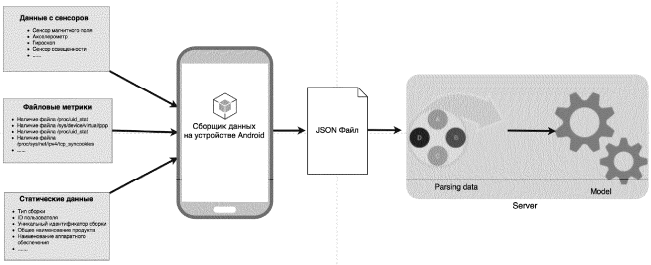
Фиг. 5



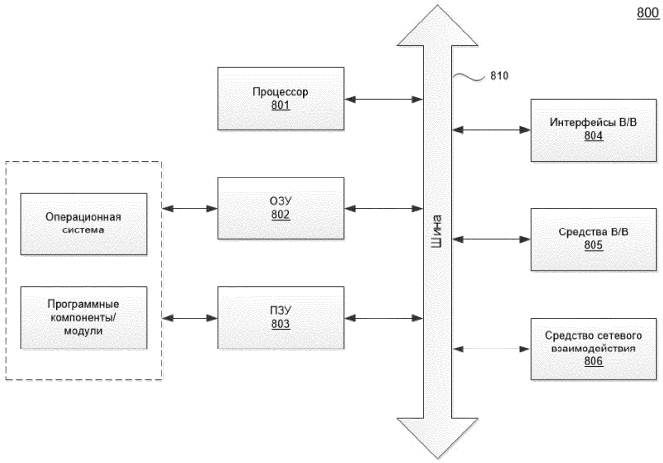
Фиг. 6



Фиг. 7


Данные с сенсоров
Файловые метрики
Статические данные
Сборщик данных на устройстве Android
JSON Файл
Parsing data
Model
Server

Фиг. 8


800
810
Процессор 801
ОЗУ 802
ПЗУ 803
Интерфейсы В/В 804
Средства В/В 805
Средство сетевого взаимодействия 806
Шина
Операционная система
Программные компоненты/ модули

Фиг. 9