РОССИЙСКАЯ ФЕДЕРАЦИЯ

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

(19) **RU** (11) **2 652 451** (13) **C2**

(51) МПК
G06F 21/54 (2013.01)

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(52) СПК
G06F 21/54 (2017.05); G06F 21/128 (2017.05); G06F 21/566 (2017.05)

(21)(22) Заявка: **2016136226, 08.09.2016**

(24) Дата начала отсчета срока действия патента:
08.09.2016

Дата регистрации:
26.04.2018

Приоритет(ы):
(22) Дата подачи заявки: **08.09.2016**

(43) Дата публикации заявки: **15.03.2018** Бюл. № 8

(45) Опубликовано: **26.04.2018** Бюл. № 12

Адрес для переписки:
125212, Москва, Ленинградское ш., 39а, стр. 3, АО "Лаборатория Касперского", Управление по интеллектуальной собственности, Надежда Васильевна Кащенко

(72) Автор(ы):
Купреев Олег Викторович (RU),
Гальченко Антон Борисович (RU),
Устинов Михаил Валерьевич (RU),
Кондратов Виталий Викторович (RU),
Кусков Владимир Анатольевич (RU)

(73) Патентообладатель(и):
Акционерное общество "Лаборатория Касперского" (RU)

(56) Список документов, цитированных в отчете о поиске: **RU 2446459 C1, 27.03.2012. RU 2571594 C2, 20.12.2015. US 8364811 B1, 29.01.2013. EP 2133809 A2, 16.12.2009. US 2012/0304296 A1, 29.11.2012.**

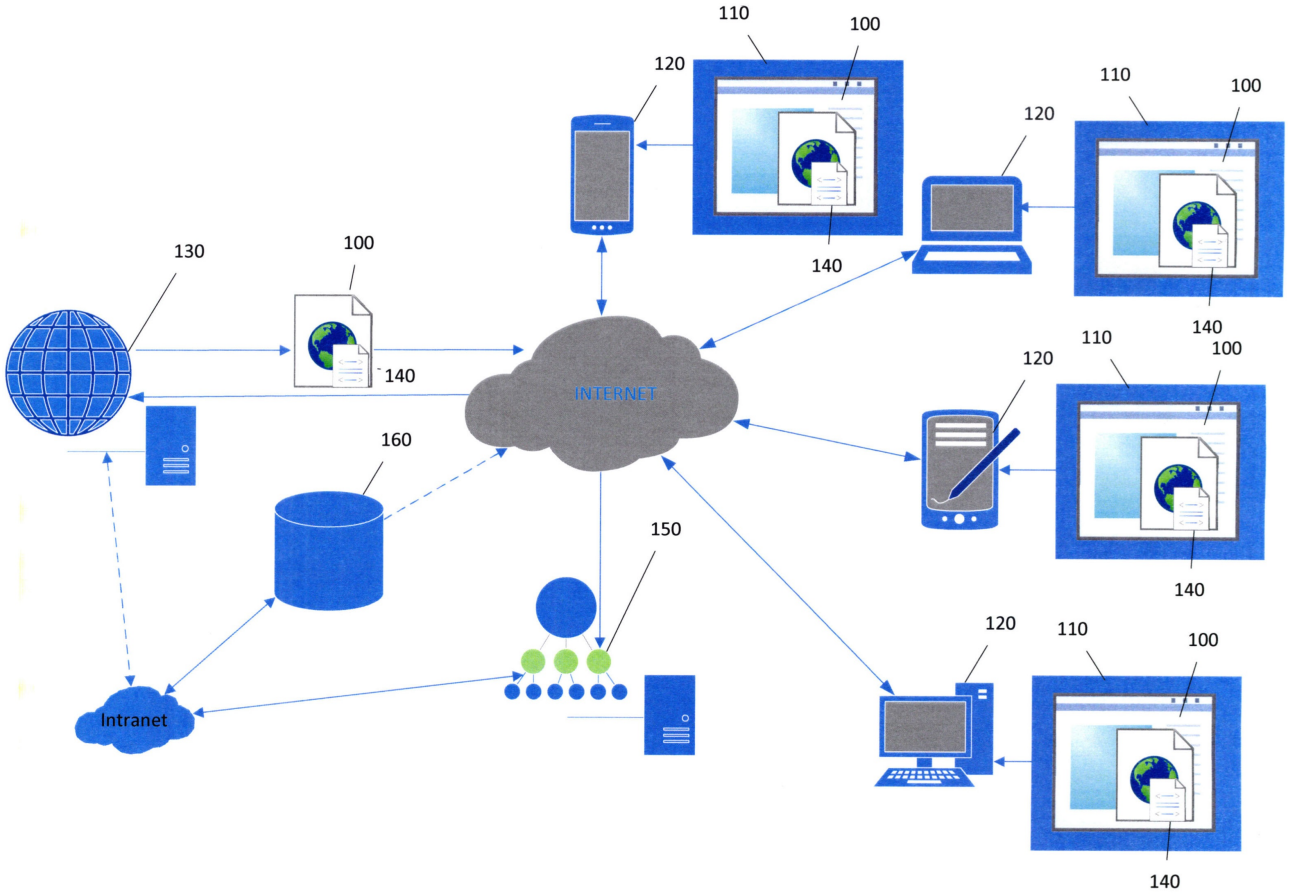
(54) Способы обнаружения аномальных элементов веб-страниц

(57) Реферат:

Изобретение относится к области обнаружения аномальных элементов веб-страницы. Технический результат заключается в обеспечении обнаружения аномальных элементов веб-страницы, возникших на стороне пользователя, без установки дополнительного программного обеспечения. Технический результат достигается путем использования кластерных статистических моделей веб-страниц, при этом сведения о содержимом элементов, используемые для обнаружения аномальных элементов и построения кластерной статистической модели, собираются скриптом на стороне веб-клиента, содержащимся непосредственно на веб-странице, сведения о содержимом элементов которой собираются. 16 з.п. ф-лы, 7 ил.

RU 2652451 C2

RU 2652451 C2



Фиг.1

RU 2652451 C2

RU 2652451 C2

RUSSIAN FEDERATION

FEDERAL SERVICE
FOR INTELLECTUAL PROPERTY

(19) **RU** (11) **2 652 451** (13) **C2**

(51) Int. Cl.
G06F 21/54 (2013.01)

(12) ABSTRACT OF INVENTION

(52) CPC
G06F 21/54 (2017.05); G06F 21/128 (2017.05); G06F 21/566 (2017.05)

(21)(22) Application: **2016136226, 08.09.2016**

(24) Effective date for property rights:
08.09.2016

Registration date:
26.04.2018

Priority:
(22) Date of filing: **08.09.2016**

(43) Application published: **15.03.2018** Bull. № **8**

(45) Date of publication: **26.04.2018** Bull. № **12**

Mail address:
125212, Moskva, Leningradskoe sh., 39a, str. 3, AO "Laboratoriya Kasperskogo", Upravlenie po intellektualnoj sobstvennosti, Nadezhda Vasilevna Kashchenko

(72) Inventor(s):
Kupreev Oleg Viktorovich (RU),
Galchenko Anton Borisovich (RU),
Ustinov Mikhail Valerevich (RU),
Kondratov Vitalij Viktorovich (RU),
Kuskov Vladimir Anatolevich (RU)

(73) Proprietor(s):
Aktsionernoe obshchestvo "Laboratoriya Kasperskogo" (RU)

(54) METHODS FOR ANOMALOUS ELEMENTS DETECTION ON WEB PAGES

(57) Abstract:

FIELD: information technology.

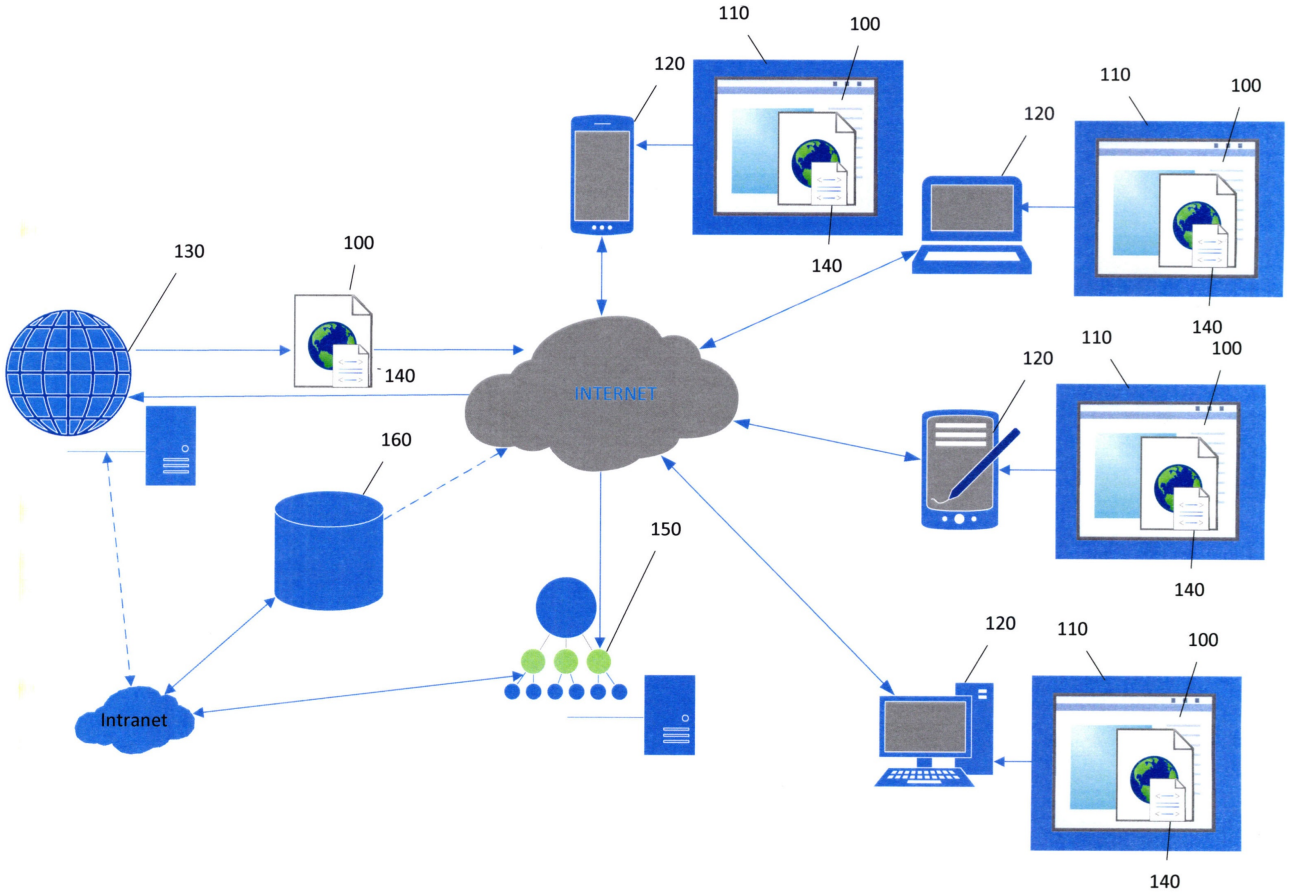
SUBSTANCE: cluster statistical models of web pages are used. Information about the content of elements used to detect anomalous elements and build a cluster statistical model is collected by a script on the side of the Web client contained directly on the web page, for which information about the elements content is collected.

EFFECT: detection of anomalous web page elements on the user side, without installation of additional software.

17 cl, 7 dwg

RU 2652451 C2

RU 2652451 C2



Фиг.1

RU 2652451 C2

RU 2652451 C2

Область техники

Изобретение относится к способам обнаружения аномальных элементов веб-страницы.

Уровень техники

В последнее время банки и другие финансовые организации активно внедряют в процесс банковского обслуживания системы веб-банкинга (интернет банкинга). Веб-банкинг - это общее название технологий дистанционного банковского обслуживания, а также доступа к счетам и операциям (по ним), предоставляется в любое время и с любого компьютера, имеющего доступ в Интернет. Для выполнения операций используется веб-клиент (например, браузер).

Широкое применение указанных технологий закономерно привлекает злоумышленников, которые заинтересованы в хищении средств со счетов пользователей систем дистанционного обслуживания. Одной из популярных атак на пользователя веб-банкинга является атака, при которой вредоносным программным обеспечением (далее ПО) подменяется содержимое веб-страницы, отображаемой пользователю. Вредоносное ПО производит внедрение HTML-кода в веб-страницу. Часто эту атаку называют «человек в браузере» (англ. main in the browser) или «внедрение веб-кода» (англ. web injection). Атака может начинаться с использования, например, троянского приложения, устанавливающего в браузер жертвы вредоносное расширение, запускающееся при перезапуске браузера. После происходит перехват трафика пользователя, направляемого на определенный веб-сайт (чаще всего банковский). Далее происходит изменение веб-страницы (на этапе загрузки или открытия), отображаемой пользователю, что позволяет модифицировать внешний вид того или иного элемента веб-страницы, похищать вводимые аутентификационные данные жертвы или перенаправлять переводимые пользователем средства на сторонний счет.

В настоящее время существуют решения, направленные на повышение безопасности работы пользователя в сети с учетом атак, внедряющих сторонний код в веб-страницу.

Так, публикация EP 2199940 описывает способ определения атаки «man in the browser» с помощью «отпечатка» (англ. fingerprint) транзакции, ассоциированного с веб-сайтом. Отпечатком в частном случае может являться количество ожидаемых транзакций вывода-вывода. Если есть отклонение, транзакция прерывается.

Публикация EP 2529304 описывает систему, которая сравнивает поведение пользователя в текущей сессии с усредненным поведением. Поддерживается определение атаки «man in the browser». Во время данной атаки выделяются атрибуты пользователя (например, логин и IP-адрес), и по ним в дальнейшем анализируется поведение пользователя.

Однако в настоящий момент уровень техники не содержит решений, которые могли бы эффективно определить, была ли изменена веб-страница вредоносным ПО, и отыскать аномальные элементы в версии веб-страницы на стороне пользователя без установки дополнительного программного обеспечения. В то же время дополнительное программное обеспечение, такое как различные клиенты безопасности, тонкие клиенты (англ. light agent) и другие антивирусные средства, не всегда возможно установить на стороне пользователя, что в результате приводит к ошибкам первого и второго рода в работе антивирусного приложения. Так, например, ошибкой первого рода является пропуск атаки типа «man in the browser» на вычислительную систему (компьютера) с целью захвата канала передачи данных и получения доступа ко всей передаваемой информации, а ошибкой второго рода является ошибочное определение легального изменения веб-страницы на стороне пользователя как аномальное.

Раскрытие изобретения

Настоящее изобретение предназначено для обнаружения аномальных элементов веб-страницы, возникших на клиентской стороне, без установки дополнительного (помимо уже установленного веб-клиента) программного обеспечения на стороне клиента.

Технический результат заключается в обеспечении обнаружения аномальных элементов веб-страницы, возникших на стороне пользователя, без установки дополнительного программного обеспечения. Технический результат достигается путем использования кластерных статистических моделей веб-страниц, при этом сведения о содержимом элементов, используемые для обнаружения аномальных элементов и построения кластерной статистической модели, собираются скриптом на стороне веб-клиента, содержащимся непосредственно на веб-странице, сведения о содержимом элементов которой собираются.

Объектом настоящего изобретения является способ - способ обнаружения аномального элемента веб-страницы на основании кластерной статистической модели веб-страницы, в котором строят статистическую модель веб-страницы и на основании указанной модели обнаруживают аномальные элементы веб-страницы.

Для построения кластерной статистической модели получают по меньшей мере одним, веб-клиентом, реализованным на компьютерном устройстве пользователя, веб-страницу от веб-сервера, при этом веб-страница содержит скрипт, который при выполнении собирает сведения о содержимом по меньшей мере одного элемента веб-страницы на стороне веб-клиента и оправляет собранные сведения с компьютерного устройства пользователя серверу управления. Далее выполняют вышеуказанный скрипт с помощью веб-клиента, который собирает сведения о содержимом по меньшей мере одного элемента веб-страницы на стороне веб-клиента и отправляет собранные сведения с компьютерного устройства пользователя серверу управления. Полученные сведения преобразуют с помощью сервера управления в по меньшей мере один N-мерный вектор, при этом N-мерный вектор характеризует содержимое по меньшей мере одного элемента веб-страницы. И, в заключение, создают с помощью сервера управления статистическую модель веб-страницы, которая представляет собой по крайней мере один кластер в N-мерном пространстве, при этом кластер содержит по крайней мере один N-мерный вектор.

На основании построенной кластерной статистической модели веб-страницы обнаруживают аномальный элемент веб-страницы, для этого получают по меньшей мере одним, веб-клиентом, реализованным на компьютерном устройстве пользователя, веб-страницу от веб-сервера, при этом веб-страница содержит скрипт, который при выполнении собирает сведения о содержимом по меньшей мере одного элемента веб-страницы на стороне веб-клиента и оправляет собранные сведения с компьютерного устройства пользователя серверу управления. Выполняют вышеуказанный скрипт с помощью веб-клиента, скрипт собирает сведения о содержимом по меньшей мере одного элемента веб-страницы на стороне веб-клиента и отправляет собранные сведения с компьютерного устройства пользователя серверу управления. Далее преобразуют с помощью сервера управления полученные сведения в по меньшей мере один N-мерный вектор, где N-мерный вектор характеризуют содержимое по меньшей мере одного элемента веб-страницы и сравнивают с помощью сервера управления полученный N-мерный вектор с кластерами статистической модели веб-страницы, где определяют расстояние между полученным N-мерным вектором элемента и центрами всех кластеров статистической модели. В результате сравнения обнаруживают с помощью сервера

управления аномальный элемент веб-страницы, где аномальным признается элемент, когда выполняется по крайней мере одно из следующих условий:

- расстояние между полученным N-мерным вектором и центрами всех кластеров статистической модели в N-мерном пространстве больше радиусов этих кластеров;
- мера близости между полученным N-мерным вектором и центрами всех кластеров модели в N-мерном пространстве больше порогового значения;
- мера близости между полученным N-мерным вектором и наиболее удаленными от центра кластеров N-мерными векторами кластеров статистической модели в N-мерном пространстве больше порогового значения.

В частном случае получаемая веб-страница при построении статистической модели веб-страницы заведомо не содержит аномальных элементов.

Элементами веб-страницы, о содержимом которых собирают сведения, являются элементы, по меньшей мере, следующих видов:

- объекты:
- апплеты;
- скрипты;
- машинный код (англ. native code);
- формы.

В частном случае сведения собираются о содержимом по меньшей мере двух элементов веб-страницы, при этом элементы относятся к разным видам элементов или элементы относятся к одному виду элементов.

Для создания кластеров могут использоваться иерархические методы, например, кластер создают агломеративным методом, в котором наиболее близкие (по расстоянию) N-мерные векторы элементов выделяются в кластеры или наиболее близкие (по расстоянию) кластеры объединяют в один кластер. При применении этого метода используется расстояние: линейное, или евклидово, или обобщенное степенное Минковского, или Чебышева, или Манхэттенское. А наиболее близкими признаются векторы, имеющие наименьшее взаимное расстояние, и кластер могут выделять до тех пор, пока радиус кластера максимально не приблизится к пороговому значению радиуса, где максимальным приближенным является радиус, который при следующем акте выделения кластера превысит пороговое значение радиуса. В другом случае выделяют кластер до тех пор, пока не останется кластеров или векторов с допустимой мерой близости, где допустимой мерой близости считается мера, не превышающая установленное пороговое значение. Наиболее близкими признаются кластеры, имеющие наименьшее расстояние между центрами.

В другом частном случае кластеры создают дивизимным методом, где кластер образуют векторы, взаимное расстояние которых меньше предельно допустимого расстояния, при этом предельная допустимость расстояния определяется пороговым значением, а кластеры отделяют, например, до тех пор, пока радиус кластера не станет равным или меньше порогового значения радиуса.

Для создания кластера могут использоваться и неиерархические методы. Аномальным может признаваться элемент или группа элементов, N-мерный вектор которых не соответствует построенной статистической модели веб-страницы, а именно не принадлежит ни одному из кластеров модели.

Краткое описание чертежей

Сопровождающие чертежи включены для обеспечения дополнительного понимания изобретения и составляют часть этого описания, показывают варианты осуществления изобретения и совместно с описанием служат для объяснения принципов изобретения.

Заявленное изобретение поясняется чертежами, где:

на Фиг. 1 изображена система обнаружения аномалий, предназначенная для построения статистических моделей веб-страниц 100 и для обнаружения аномальных элементов веб-страницы;

на Фиг. 2 изображен пример N-мерного пространства со статистическими моделями и метриками кластера;

на Фиг. 3 изображены способы, осуществляемые системой обнаружения аномалий;

на Фиг. 4 изображены визуализации статистических моделей;

на Фиг. 5 изображена компьютерная система общего назначения.

Хотя изобретение может иметь различные модификации и альтернативные формы, характерные признаки, показанные в качестве примера на чертежах, будут описаны подробно. Следует понимать, однако, что цель описания заключается не в ограничении изобретения конкретным его воплощением. Наоборот, целью описания является охват всех изменений, модификаций, входящих в рамки данного изобретения, как это определено в приложенной формуле.

Осуществление изобретения

Объекты и признаки настоящего изобретения, способы для достижения этих объектов и признаков станут очевидными посредством отсылки к примерным вариантам осуществления. Однако настоящее изобретение не ограничивается примерными вариантами осуществления, раскрытыми ниже, оно может воплощаться в различных видах. Приведенное описание предназначено для помощи специалисту в области техники для исчерпывающего понимания изобретения, которое определяется только в объеме приложенной формулы.

Веб-страница - данные (код), созданные веб-сервером для обработки веб-клиентом (браузером) и организованные с применением языков гипертекстовой разметки (HTML, XHTML, XML, WML, VML, PGML, SVG, XBRL и др.) и сценарных языков (JScript, JavaScript, ActionScript, Tcl, Lua, Perl, PHP, Python, REBOL, Ruby и др.).

Контент - содержимое веб-страницы.

Скрипт (сценарий) - исполняемая процедура, написанная на сценарном языке, которая запускается на исполнение на стороны сервера или клиента по запросу, поступившему при отображении строго определенной веб-страницы.

Встроенный скрипт (inline скрипт) - скрипт, исполняемый код которого (тело) является частью контента веб-страницы. В частном случае располагается между тегами <script></script>.

Тег (метка) - специальная конструкция языка разметки гипертекста. Представляет собой текст, заключенный в угловые скобки <имя_тега>. Каждый тег несет определенную команду браузеру, как его (тег) и последующее содержимое отобразить. Теги в частном случае имеют атрибуты, которые уточняют тег, расширяют возможности тега и позволяют более гибко управлять, например, содержимым тега-контейнера. Например, <script src="URL">…</script>. Атрибут src указывает на расположение тела скрипта.

Тег-контейнер - парный тег, имеет открывающий и закрывающий теги. Может содержать как текст, так и другие элементы гипертекстового языка.

Элемент веб-страницы (элемент языка разметки) - комбинация начального тега, конечного тега (в некоторых случаях начальный и конечные теги совпадают, например в случае тега
) и содержимого между тегами. Совокупность элементов веб-страницы образуют содержимое веб-страницы. Существуют, по меньшей мере, следующие виды элементов, которые отличаются именами соответствующих тегов:

- гиперссылки;
- текстовые блоки;
- форматирование текста;
- списки;
- объекты:
  - медиафайлы;
  - апплеты; о скрипты;
  - машинный код (англ. native code);
  - и др;
  - изображения;
- карта изображений;
- таблицы;
- формы;
- символы.

N-мерный вектор элемента - упорядоченный набор из n действительных чисел, где числа есть координаты вектора. Количество координат вектора называется размерностью вектора. Координаты определяют положение соответствующего элемента (например, скрипта) или группы элементов одного вида (например, элементов форм) веб-страницы в N-мерном пространстве (на Фиг. 2 приведен пример двумерного пространства). Вектор получают преобразованием сведений о содержимом элемента или группы элементов. Вектор отражает некоторую информацию о содержимом элемента или группы элементов. В частном случае каждая координата отражает одну из характеристик содержимого элемента, например, одна координата характеризует число операторов в скрипте, другая - число операторов eval. Также числа могут отражать лексикографический порядок строковых параметров содержания элементов или расстояние Левенштейна между строковыми параметрами разных элементов. Например, на Фиг. 2 изображены примеры векторов, в частности двумерные векторы с координатами (1666,1889) и (1686,1789)

Кластер - совокупность допустимых значений координат векторов для строго определенного элемента или группы элементов в N-мерном пространстве. Рассматриваемый элемент или группа элементов относится к некоторому кластеру, если расстояние от N-мерного вектора элемента до центра данного кластера меньше радиуса кластера в направлении N-мерного вектора. На Фиг. 2 показан пример кластера 210'. В частном случае элемент относится к некоторому кластеру, если значение расстояния (на Фиг. 2 «d'») от N-мерного вектора элемента до ближайшего N-мерного вектора элемента данного кластера меньше предельно допустимого (порогового значения расстояния [d']) или если значение расстояния (на Фиг. 2 «d») от N-мерного вектора элемента до центра данного кластера меньше радиуса этого кластера. Например, расстояние от вектора (1666,1889) до центра кластера меньше радиуса кластера, следовательно, элемент или группа элементов, содержание которых отражает вектор, принадлежат данному кластеру и напротив - расстояние от вектора (1686,1789) до центра кластера больше и радиуса кластера, и расстояния до ближайшего N-мерного вектора больше порогового значения, следовательно, элемент или группа элементов, содержание которых отражает вектор, не принадлежат данному кластеру. Варианты расстояний для оценки близости:

- линейное расстояние;
- евклидово расстояние;
- квадрат евклидова расстояния;

- обобщенное степенное расстояние Минковского;
- расстояние Чебышева;
- Манхэттенское расстояние.

Мера близости (степень сходства, коэффициент сходства) - безразмерный показатель для определения сходства элементов веб-страницы. Для определения меры близости используются меры:

- Охаи;
- Жаккара;
- Сокала-Снита;
- Кульчинского;
- симметричная Дайса.

Центр кластера (центроид) - это среднее геометрическое место N-мерных векторов в N-мерном пространстве. Для кластеров, состоящих из одного вектора, данный вектор будет являться центром кластера.

Радиус кластера (на Фиг. 2 «R») - максимальное расстояние N-мерных векторов, входящих в кластер, от центра кластера.

Для кластеризации используют различные известные алгоритмы и подходы, в том числе иерархические (агломеративные и дивизивные) и неиерархические.

Статистическая модель элементов веб-страницы (модель элементов веб-страницы) - совокупность кластеров 210 для элементов одного вида или групп элементов одного вида. Например, статистическая модель скриптов веб страницы, статистическая модель форм веб-страницы. На Фиг. 2 статистические модели элементов веб-страницы обозначены 220. Для моделей, состоящих из одного кластера, данный кластер будет являться моделью элементов.

Статистическая модель веб-страницы (модель веб-страницы) - совокупность кластеров элементов веб-страницы всех видов и/или групп элементов (в том числе групп элементов, содержащих элементы разных видов). Например, статистическая модель страницы авторизации. Иными словами, статистическая модель веб-страницы 230 есть совокупность моделей элементов веб-страницы 220. По аналогии статистической моделью веб-сайта будет совокупность кластеров элементов веб-страницы всех видов и/или групп элементов всех веб-страниц веб-сайта. Иными словами, статистическая модель веб-сайта (на чертежах не указана) есть совокупность моделей веб-страниц 230.

Аномальный элемент веб-страницы - элемент веб-страницы, вектор которого не относится ни к одному из кластеров статистической модели, построенной для элементов данного типа, или имеет статистическую значимость ниже пороговой.

Статистическая значимость элемента - значение отношения числа встречаемости оцениваемого элемента в контенте веб-страниц к общему числу полученных для построения модели веб-страниц или к числу полученных для построения модели веб-страниц на некотором участке (участке оценивания), где длина участка определяется числом полученных для построения модели веб-страниц с некоторого момента, например момента начала наблюдения за элементом. Например, если получено 100 страниц и оцениваемый элемент встретился 30 раз, то статистическая значимость составит 30%.

Статистическая значимость кластера - значение отношения количества элементов, векторы которых образуют оцениваемый кластер, в контенте веб-страницы к общему числу полученных для построения модели веб-страниц; или к числу полученных для построения модели веб-страниц на некотором участке, где длина участка определяется числом полученных для построения модели веб-страниц с некоторого момента, например момента начала наблюдения за кластером.

Пороговое значение статистической значимости - значение статистической значимости элемента или кластера, при превышение которого элемент или кластер (и элементы кластера) признается статистически значимым, в том случае, если значение статистической значимости элемента кластера ниже установленного порогового значения, то элемент или кластер считаются аномальными.

Для создания кластеров могут использоваться иерархические методы, например, кластер создают агломеративным методом, в котором наиболее близкие (по расстоянию) N-мерные векторы элементов выделяются в кластеры или наиболее близкие (по расстоянию) кластеры объединяют в один кластер. При применение этого метода используется расстояние: линейное, или евклидово, или обобщенное степенное Минковского, или Чебышева, или Манхэттенское. А наиболее близкими признаются векторы, имеющие наименьшее взаимное расстояние, и кластер могут выделять до тех пор, пока радиус кластера максимально не приблизится к пороговому значению радиуса, где максимальным приближенным является радиус, который при следующем акте выделения кластера превысит пороговое значение радиуса. В другом случае выделяют кластер до тех пор, пока не останется кластеров или векторов с допустимой мерой близости, где допустимой мерой близости считается мера, не превышающая установленное пороговое значение. Наиболее близкими признаются кластеры, имеющие наименьшее расстояние между центрами.

В другом частном случае кластеры создают дивизимным методом, где кластер образуют векторы, взаимное расстояние которых меньше предельно допустимого расстояния, при этом предельная допустимость расстояния определяется пороговым значением, а кластеры отделяют, например, до тех пор, пока радиус кластера не станет равным или меньше порогового значения радиуса.

На Фиг. 1 изображена система обнаружения аномалий, предназначенная для построения статистических моделей веб-страниц 100 и для обнаружения аномальных элементов веб-страницы. Система включает в себя: устройство пользователя 120, с установленным на нем веб-клиентом 110; веб-сервер 130; сервер управления 150 и базу данных 160.

На устройстве пользователя 120 реализован веб-клиент 110, в частном случае это браузер. Веб-клиент 110 предназначен для запроса, обработки, манипулирования и отображения содержания веб-сайтов, где веб-сайт является совокупностью логически связанных между собой веб-страниц 100. Веб-клиент 110 отправляет запросы на получение ресурсов, обозначенных, например, URL (uniform resource locator) адресами веб-серверу 130 и получает ответы, как правило, вместе с веб-страницей 100 или элементом веб-страницы от веб-сервера 130. Веб-сервер 130 по запросу от веб-клиента 110 выдает готовую веб-страницу 100 или формирует страницу динамически, в описываемом изобретении веб-сервером 130 к каждой веб-странице 100, отправляемой клиенту, дополнительно к обычному содержанию добавляется скрипт 140. Назначение скрипта 140, по меньшей мере, собирать на стороне веб-клиента 110 данные веб-страницы 100 (сведения об элементах или группе элементов веб-страницы, сведения об элементе, в частном случае содержимое элемента), которая данный скрипт 140 содержит. В частном случае сведением об элементе веб-страницы 100 является содержимое данного элемента. Как упоминалось в уровне техники, элементы веб-страницы 100 и содержимое этих элементов веб-страницы 100 на стороне веб-клиента 100 могут отличаться от элементов и содержимого этих элементов той же версии веб-страницы 100 на стороне веб-сервера 130, по причине динамического обновления веб-страницы на стороне веб-клиента 110 или в результате атаки «man in the browser».

Сервер управления 150 получает собранные скриптом сведения об элементах или группе элементов веб-страницы. При этом скрипт может отправлять собранные данные как в «сыром» (англ. "raw"), так и в преобразованном виде, формат отправляемых данных определяется функциональностью скрипта 140, который добавлен веб-сервером 130 на веб-страницу 100, а именно:

- скрипт в процессе выполнения отправляет строго определенные сведения об элементах веб-страницы 100 в строго заданном виде, которые заданы функционалом скрипта; или
- скрипт отправляет данные веб-серверу 130 или серверу управления о своем успешном запуске на стороне клиента 110 и получает в ответ команду о том, о каких элементах веб-страницы 100 и в каком виде нужно собрать и отправить сведения приемнику (веб-серверу 130 или непосредственно серверу управления 150).

Основными способами трансформирования (преобразования) данных являются:

- квантование;
- сортировка;
- слияние (склеивание);
- группировка;
- настройка набора данных;
- табличная подстановка значений;
- вычисляемые значения;
- кодирование данных;
- нормализация (масштабирование).

В частном случае в результате преобразования данных данные приобретают свойства информации.

Одним из способов преобразования скриптов является построение абстрактного синтаксического дерева и передача приемнику (веб-серверу 130 или непосредственно серверу управления 150) только значимых операторов и конструкций, которые заранее предопределяются настройками скрипта 140 или командами от приемника.

Все собранные скриптом 140 данные передаются, в итоге, серверу управления 150. Сервер управления 150 может получать данные напрямую от веб-клиентов 110 либо через веб-сервер 130. В частном случае сервер управления 150 может находиться в одной сети с веб-сервером 130. Собранные данные сервером управления 150 используются для построения статистической модели веб-страницы 230 и обнаружения аномальных элементов веб-страниц. На сервере управления 150 реализован ряд средств (на чертежах не указаны). Средство обработки, реализованное на сервере управления 150, преобразует собранные скриптами 140 данные в N-мерные векторы, полученные векторы хранятся в базе данных 160.

Средство анализа, реализованное на сервере управления 150, предназначено для формирования кластеров 210 из полученных векторов и обнаружения аномальных элементов или групп элементов, содержимое которых отражают полученные векторы, данное назначение реализуется за счет взаимного сравнения N-мерных векторов и сформированных кластеров 210 в N-мерном пространстве.

База данных 160 хранит построенные модели и векторы.

Описанная система осуществляет несколько способов: способ построения статистической модели веб-страницы 230 и способ обнаружения аномальных элементов веб-страницы 100 с помощью построенной модели веб-страницы 230, изображенные на Фиг. 3.

Способ построения статистической модели веб-страницы 230 осуществляется

следующим образом. На этапе 300 пользователь со своего устройства получает доступ к веб-сайту, где веб-клиент 110 по запросу к веб-серверу 130 получает от веб-сервера 130 веб-страницу 100 сайта, на веб-страницу 100 веб-сервером 130 при этом добавляется скрипт 140. На этапе 310 скрипт выполняется на стороне веб-клиента 110, собирая данные, содержащиеся в веб-странице 100. Данные, собираемые скриптом 140, могут содержать различные сведения, в частном случае скрипт 140 собирает содержимое по меньшей мере одного элемента веб-страницы (скрипта, формы и т.д.). Данные, собранные скриптом 140, при необходимости трансформируются, данные трансформируются либо самим скриптом 140, либо средством обработки на сервере управления 150 и на этапе 320 собранные данные преобразуются в по меньшей мере один N-мерный вектор, который сохраняется на этапе 330. Из по меньшей мере одного вектора на этапе 350 создают по меньшей мере один кластер 210. На основании по меньшей мере одного созданного кластера 210 строят на этапе 360 статистическую модель веб-страницы 230.

В частном случае, после сохранения полученного N-мерного вектора, на этапе 300' получают веб-страницу 100 другим веб-клиентом 110 и на основании собранных данных с этой веб страницы получают дополнительно на этапе 320 N-мерные векторы, только после этого создают кластеры.

В другом частном случае после создания кластеров 210 и построения модели 230 на этапе 300" получают веб-страницу 100 другим веб-клиентом 110 и на основании собранных данных скриптом 140 с этой веб-страницы получают N-мерные векторы, и на основании полученных N-мерных векторов корректируют (обновляют) ранее созданные кластеры 210 (изменяют радиус, центр/центроид) или создают новые кластеры 210, тем самым уточняя (скорректированными кластерами 210) и дополняя (вновь созданными кластерами 210) статистическую модель веб-страницы 230. При этом данные, собираемые скриптом 140, могут отличаться от данных, собираемых скриптом 140 на предыдущей итерации, например, собираются сведения о других элементах веб-страницы 100.

Способ обнаружения аномальных элементов на основании статистической модели веб-страницы 230. На этапе 300 пользователь со своего устройства получает доступ к веб-сайту, где веб-клиент 110 по запросу к веб-серверу 130 получает от веб-севера 130 веб-страницу 100 сайта, на веб-страницу 100 веб-сервером 130 при этом добавляется скрипт 140. На этапе 310 скрипт выполняется на стороне веб-клиента, собирая данные, содержащиеся в веб-странице 100. Данные, собираемые скриптом 140, могут содержать различные сведения, в частном случае скрипт собирает содержимое по меньшей мере одного элемента веб-страницы (скрипта, формы и т.д.). Данные, собранные скриптом 140, при необходимости трансформируются, при этом данные трансформируются либо самим скриптом 140, либо средством обработки сервера управления 150 и на этапе 320 собранные данные преобразуются в по меньшей мере один N-мерный вектор, который сохраняется на этапе 330. Полученный вектор на этапе 370 сравнивается (путем определения взаимного расстояния, например, между полученным вектором и центром кластера) с кластерами построенной статистической модели веб-страницы 230 и/или N-мерными векторами данной модели 230. На этапе 370 в результате сравнения анализируемый элемент признается аномальным, элемент признается аномальным, когда:

- расстояние между N-мерным вектором элемента и центрами всех кластеров модели, в N-мерном пространстве, больше радиусов этих кластеров; или
- мера близости между N-мерным вектором элемента и центрами всех кластеров

модели, в N-мерном пространстве, больше порогового значения; или

- мера близости между N-мерным вектором элемента и наиболее удаленными от центра кластеров N-мерными векторами кластеров модели, в N-мерном пространстве, больше порогового значения.

В частном случае, если элемент не признан аномальным, на этапе 350' N-мерный вектор данного элемента добавляется к статистической модели веб-страницы 230.

В частном случае при обнаружении аномального элемента веб-страницы 100 веб-сервер 130 разрывает соединение с веб-клиентом 110 и устройством пользователя 120 или соединение сохраняется, но веб-сервер 130 перестает отвечать на запросы клиента 110 (передача данных по соединению приостанавливается). В момент приостановки передачи данных обнаруженный аномальный элемент веб-страницы проверяется антивирусными средствами (на чертежах не указаны) сервера управления 150 на наличие вредоносного функционала (опасности) или производят наблюдение за данным элементом и, если вокруг него сформируется кластер со статистической значимостью выше пороговой, обнаруженный аномальный элемент признается безопасным и соединение размораживается, и сессия продолжается.

В частном случае, когда модель строится на основании веб-страниц, о которых заведомо неизвестно, содержат они аномальные элементы или нет, возможна коллизия: N-мерный вектор элемента не попадает ни в один из кластеров модели и возникает дилемма - создавать новый кластер на базе данного вектора или признавать элемент, содержимое которого отражает данный вектор, аномальным. Коллизия может быть разрешена на основании оценки статистической значимости элемента или кластера, который возможно создать на основании элементов, подобных (близких) оцениваемому на участке оценивания. А именно на основании отношения числа веб-страниц, содержащих оцениваемый элемент (или близкие элементы, элементы, расстояние между N-мерными векторами которых в N-мерном пространстве меньше некоторого порогового значения), к общему числу веб-страниц, используемых при построении модели на оцениваемом участке, где длина участка измеряется в количестве страниц или итераций. Если значение статистической значимости оцениваемого элемента веб-страницы на участке оценки близко (близость определяется пороговым значением) значению статистической значимости других элементов (или среднему значению статистической значимости других элементов) на данном участке или превышает некоторое пороговое значение, например, в 20%, то элемент признается статистически значимым, иначе (если не превышает) - аномальным. Например, на этапе построения модели, появился некоторый элемент веб-страницы 100, вектор которого не попадает ни в один из ранее созданных кластеров 210, необходимо определить, является ли данный элемент аномальным. Для этого определим его статистическую значимость на участке, где длина участка составляет 200 веб-страниц 100. При этом пороговое значение статистической значимости для данного вида элемента равно 20%. При проверке выясняют, что на данной длине элементы, близкие оцениваемому, встретились 4 раза, что соответствует значению статистической значимости в 2%, что ниже порогового значения, следовательно, оцениваемый элемент и близкие ему (кластер, который образуется вокруг оцениваемого элемента) являются аномальными. Пороговое значение статистической значимости, в частном случае, определяется как минимальное значение статистической значимости кластера для элемента того же вида. Например, модель содержит кластеры скриптов со значениями статистической значимости 25%, 32%, 47%, 95%, следовательно, пороговое значение для данного вида элементов устанавливается равным 25%.

Статистическая значимость может использоваться также при обнаружении аномальных элементов. Это, например, используется, когда статистическая модель не построена или решается дилемма, описанная выше. На первом этапе получают веб-клиенты 110 и реализованные на устройствах пользователей 120 веб-страницы 100 от веб-сервера 130, при этом веб-страницы 100 содержат скрипт 140, который при выполнении собирает сведения о содержимом по меньшей мере одного элемента веб-страницы 100 на стороне веб-клиента 110 и отправляет собранные сведения с устройства пользователя 120. Далее выполняют вышеуказанный скрипт с помощью веб-клиента 110 для сбора сведений о содержимом по меньшей мере одного элемента веб-страницы 100 на стороне веб-клиента 110 и отправки собранных сведений с устройств пользователей 120, веб-клиенты 110 которых получили веб-страницу 100. На стороне сервера управления 150 преобразуют сведения о содержимом, полученные с устройств 120, в N-мерные векторы элементов, затем кластеризуют полученные N-мерные векторы любым известным из уровня техники способом. N-мерные векторы могут формироваться для каждого элемента веб-страницы, для группы элементов, при этом для группы элементов как одного вида, так и в группу могут входить элементы разных видов. После того как кластеры 210 сформированы, при этом кластер 210 может включать по меньшей мере один вектор, определяют статистическую значимость полученных кластеров 210, где статистическая значимость определяется как отношение числа N-мерных векторов в кластере 210 к числу веб-страниц 100, с которых собраны и отправлены сведения о содержимом их элементов серверу управления 150 или веб-серверу 130. В результате аномальными элементами признаются элементы, N-мерные векторы которых образуют кластер со статистической значимостью меньше пороговой. Пороговая значимость задается способами, описанными выше, а также может зависеть от видов элементов, способов кластеризации, длины участка оценивания и т.д.

Приведем пример работы описанного выше изобретения. Пользователь запрашивает веб-страницу сайта веб-банка - https://my.KasperskyBank.ru/. На запрошенную веб-страницу добавляется скрипт 140, и страница 100 отправляется веб-клиенту 110, реализованному на устройстве пользователя 120. Скрипт 140 на стороне пользователя собирает имеющиеся на веб-странице элементы <script>:

```
<script>document.documentElement.id="js";var
…./Kasperskybank/";</script>

<script                    src="//static.kaspersky.ru/dist/kfs/kfs.js"
crossorigin="anonymous"></script>

<script              src="https://              static.kaspersky.ru
/ib/prod/2842c77095d860e412d7a8cf30231fd53c89fb4e/    Kasperskybank    /
Kasperskybank.js" crossorigin="anonymous"></script>

<script async="" src="/kfs/kfs"></script>

<script>!function(){var              e=document.getElementById("before-
init__noscript");e&&(e.className="ui-browser__holder-block-hide");var
o=function(){try{return"withCredentials"in                          new
XMLHttpRequest}catch(e){return!1}}();if(o){var
t=function(){if(navigator.cookieEnabled)return!0;document.cookie="cooki
etest=1";var         e=-1!=document.cookie.indexOf("cookietest=");return
document.cookie="cookietest=1;    expires=Thu,    01-Jan-1970    00:00:01
GMT",e}();if(t)document.body.removeChild(document.getElementById("befor
e-init"));else{var                    n=document.getElementById("before-
init__nocookies");n&&(n.className="ui-browser__holder-block")}}else{var
r=document.getElementById("before-init__old-
browser");r&&(r.className="ui-browser__holder-block")}}();</script>
```

Для элементов <script>, имеющих атрибут src, выполняется загрузка и нормализация тела скрипта, для inline-скриптов - только нормализация. Например, для приведенных выше inline-скриптов, нормализованная форма может быть следующей (сохранены только значимые конструкции языка и стандартные объекты/методы, литералы «обезличены»):

```
document.documentElement.i0=v0;vari1=window.i1||{};i1.i2=v1,i1.i3=v2,i1
.i4=v3,i1.i5=v4,i1.i6={i7:v5,i8:v6},i1.i9=v7;

!function(){vari0=document.getElementById(v0);i0&&(i0.i1=v1);vari2=func
tion(){try{returnv2innewXMLHttpRequest}catch(i0){return!v3}}();if(i2){v
ari3=function(){if(navigator.i4)return!v4;document.cookie=v5;vari0=-
v3!=document.cookie.indexOf(v6);returndocument.cookie=v7,i0}();if(i3)do
cument.body.removeChild(document.getElementById(v8));else{vari5=documen
t.getElementById(v9);i5&&(i5.i1=v10)}}else{vari6=document.getElementByI
d(v11);i6&&(i6.i1=v10)}}();
```

Далее скрипт 140 собирает имеющиеся на странице элементы <input>:

```
<input autocomplete="off" autocorrect="off" autocapitalize="off"
class="m-login__form-field-input ng-pristine ng-invalid ng-invalid-
required ng-touched" type="text" … ng-blur="login.focus = false"
placeholder="Пароль">
```

Скрипт 140 преобразует собранные данные элементов <input>, выполняя нормализацию, например, так (атрибуты сортируются по алфавиту, имя тэга вырезается, пробелы в значениях атрибутов вырезаются, атрибуты перечисляются через «;»):

```
<autocapitalize=off;autocomplete=off;autocorrect=off;class=m-
login__form-field-inputng-pristineng-invalidng-invalid-requiredng-
touched;name=lg;ng-blur=login.focus=false;warmUp();;ng-
change=input(true);ng-disabled=false;ng-
keydown=login.focus=true&&$event.keyCode===13&&authUser();ng-
keyup=fix(login.form.lg,$event);ng-

model=login.lg;placeholder=Логин;spellcheck=false;style=padding:0px;;ty
pe=text;ui-focus=login.setFocus;validator=validator.lg>

<autocapitalize=off;autocomplete=off;autocorrect=off;class=m-
login__form-field-inputng-pristineng-untouchedng-invalidng-invalid-
required;name=pw;ng-blur=login.focus=false;ng-change=input();ng-
disabled=false;ng-
keydown=login.focus=true&&$event.keyCode===13&&authUser();ng-
keyup=fix(login.form.pw,$event);ng-
model=login.pw;placeholder=Пароль;spellcheck=false;type=password;valida
tor=validator.pw>
```

Скрипт 140 отправляет собранные данные на сервер управления 150. Сервер управления 150 обрабатывает собранные данные элементов <script> в контексте соответствующей модели (единой для всех элементов script - статистическая модель элементов 220 вида скрипт) следующим образом:

- для каждого скрипта получают числовой вектор (для простоты в примере считаем, что вектор двумерный), где вектор рассчитывается из кодов символов строк (для получения кодов символов может использоваться любая известная кодировка, например ASCII), составляющих собранные данные (для inline-скриптов эти данные - содержимое нормализованного скрипта, для остальных - содержание атрибута src). Тогда для элементов <script>, содержащихся в полученной веб-странице 100, могут быть получены следующие векторы:
  - 16314,10816
  - 2254,2598
  - 16084,15036
  - 356,822
  - 20010,51838
- каждый вектор сохраняется в двумерном пространстве модели 230, в данном случае аномалии отсутствуют, все векторы попадают в сложившиеся ранее кластеры (т.е.

совпадают с данными, поступившими от скрипта 140 с тех версий веб-страниц ранее). Для наглядности на Фиг. 4а приведена визуализация модели, где точками изображены анализируемые элементы <script>, закрашенными областями - созданные ранее кластеры 210 модели 220 как части модели 230:

Сервер управления 150 обрабатывает собранные данные элементов <input> аналогичным образом, в результате визуализация имеет вид, представленный на Фиг. 4б. Поскольку аномальных элементов не выявлено, обработка на этом завершается.

Теперь предположим, что у кого-то из пользователей на той же самой странице - https://my.KasperskyBank.ru/ появился вредоносный инжект (англ. inject) в виде дополнительного элемента <script>:

```
<script    src="https://static.kasperskyBank.ru/ib/prod/bank/malware.js"
crossorigin="anonymous"></script>
```

И для него вектор, рассчитанный способом, описанным выше, равен (4560,3192) и модель приобретает вид, представленный на Фиг. 4в (красным отмечен текущий вектор, отражающий содержание инжекта, это аномалия). Обнаруженный аномальный элемент будет обработан антивирусными средствами сервера управления 150, а само соединение будет заморожено, параллельно за элементом в пространстве модели будет осуществляться наблюдение для определения его статистической значимости.

Под веб-сервером, веб-клиентом, базой данных, сервером управления с реализованными на нем средством анализа и средством обработки в настоящем изобретении понимаются реальные устройства, системы, компоненты, группы компонентов, реализованные с использованием аппаратных средств, таких как интегральные микросхемы (англ. application-specific integrated circuit, ASIC) или программируемые вентильные матрицы (англ. field-programmable gate array, FPGA) или, например, в виде комбинации программных и аппаратных средств, таких как микропроцессорная система и набор программных инструкций, а также на нейроморфных чипах (англ. neurosynaptic chips). Функциональность указанных элементов системы может быть реализована исключительно аппаратными средствами, а также в виде комбинации, где часть функциональности элементов системы реализована программными средствами, а часть аппаратными. В некоторых вариантах реализации часть элементов или все элементы могут быть исполнены на процессоре компьютера общего назначения (например, который изображен на Фиг. 5).

Фиг. 5 представляет пример компьютерной системы общего назначения, персональный компьютер или сервер 20, содержащий центральный процессор 21, системную память 22 и системную шину 23, которая содержит разные системные компоненты, в том числе память, связанную с центральным процессором 21. Системная шина 23 реализована как любая известная из уровня техники шинная структура, содержащая в свою очередь память шины или контроллер памяти шины, периферийную шину и локальную шину, которая способна взаимодействовать с любой другой шинной архитектурой. Системная память содержит постоянное запоминающее устройство (ПЗУ) 24, память с произвольным доступом (ОЗУ) 25. Основная система ввода/вывода (BIOS) 26 содержит основные процедуры, которые обеспечивают передачу информации между элементами персонального компьютера 20, например, в момент загрузки операционной системы с использованием ПЗУ 24.

Персональный компьютер 20 в свою очередь содержит жесткий диск 27 для чтения и записи данных, привод магнитных дисков 28 для чтения и записи на сменные магнитные диски 29 и оптический привод 30 для чтения и записи на сменные оптические

диски 31, такие как CD-ROM, DVD-ROM и иные оптические носители информации. Жесткий диск 27, привод магнитных дисков 28, оптический привод 30 соединены с системной шиной 23 через интерфейс жесткого диска 32, интерфейс магнитных дисков 33 и интерфейс оптического привода 34 соответственно. Приводы и соответствующие компьютерные носители информации представляют собой энергонезависимые средства хранения компьютерных инструкций, структур данных, программных модулей и прочих данных персонального компьютера 20.

Настоящее описание раскрывает реализацию системы, которая использует жесткий диск 27, сменный магнитный диск 29 и сменный оптический диск 31, но следует понимать, что возможно применение иных типов компьютерных носителей информации 56, которые способны хранить данные в доступной для чтения компьютером форме (твердотельные накопители, флеш карты памяти, цифровые диски, память с произвольным доступом (ОЗУ) и т.п.), которые подключены к системной шине 23 через контроллер 55.

Компьютер 20 имеет файловую систему 36, где хранится записанная операционная система 35, а также дополнительные программные приложения 37, другие программные модули 38 и данные программ 39. Пользователь имеет возможность вводить команды и информацию в персональный компьютер 20 посредством устройств ввода (клавиатуры 40, манипулятора «мышь» 42). Могут использоваться другие устройства ввода (не отображены): микрофон, джойстик, игровая консоль, сканнер и т.п. Подобные устройства ввода по своему обычаю подключают к компьютерной системе 20 через последовательный порт 46, который в свою очередь подсоединен к системной шине, но могут быть подключены иным способом, например при помощи параллельного порта, игрового порта или универсальной последовательной шины (USB). Монитор 47 или иной тип устройства отображения также подсоединен к системной шине 23 через интерфейс, такой как видеоадаптер 48. В дополнение к монитору 47 персональный компьютер может быть оснащен другими периферийными устройствами вывода (не отображены), например колонками, принтером и т.п.

Персональный компьютер 20 способен работать в сетевом окружении, при этом используется сетевое соединение с другим или несколькими удаленными компьютерами 49. Удаленный компьютер или компьютеры 49 являются такими же персональными компьютерами или серверами, которые имеют большинство или все упомянутые элементы, отмеченные ранее при описании существа персонального компьютера 20, представленного на Фиг. 5. В вычислительной сети могут присутствовать также и другие устройства, например маршрутизаторы, сетевые станции, пиринговые устройства или иные сетевые узлы.

Сетевые соединения могут образовывать локальную вычислительную сеть (LAN) 50 и глобальную вычислительную сеть (WAN). Такие сети применяются в корпоративных компьютерных сетях, внутренних сетях компаний и, как правило, имеют доступ к сети Интернет. В LAN- или WAN-сетях персональный компьютер 20 подключен к локальной сети 50 через сетевой адаптер или сетевой интерфейс 51. При использовании сетей персональный компьютер 20 может использовать модем 54 или иные средства обеспечения связи с глобальной вычислительной сетью, такой как Интернет. Модем 54, который является внутренним или внешним устройством, подключен к системной шине 23 посредством последовательного порта 46. Следует уточнить, что сетевые соединения являются лишь примерными и не обязаны отображать точную конфигурацию сети, т.е. в действительности существуют иные способы установления соединения техническими средствами связи одного компьютера с другим.

В заключение следует отметить, что приведенные в описании сведения являются примерами, которые не ограничивают объем настоящего изобретения, определенного формулой. Специалисту в данной области становится понятным, что могут существовать и другие варианты осуществления настоящего изобретения, согласующиеся с сущностью и объемом настоящего изобретения.

(57) Формула изобретения

1. Способ обнаружения аномального элемента веб-страницы на основании статистической модели веб-страницы, в котором:

а) строят статистическую модель веб-страницы, где:

- получают по меньшей мере одним веб-клиентом, реализованным на компьютерном устройстве пользователя, веб-страницу от веб-сервера, при этом веб-страница содержит скрипт, который при выполнении собирает сведения о содержимом по меньшей мере одного элемента веб-страницы на стороне веб-клиента и оправляет собранные сведения с компьютерного устройства пользователя серверу управления;

- выполняют вышеуказанный скрипт с помощью веб-клиента, который собирает сведения о содержимом по меньшей мере одного элемента веб-страницы на стороне веб-клиента и отправляет собранные сведения с компьютерного устройства пользователя серверу управления;

- преобразуют с помощью сервера управления полученные сведения в по меньшей мере один N-мерный вектор, где N-мерный вектор характеризует содержимое по меньшей мере одного элемента веб-страницы;

- создают с помощью сервера управления статистическую модель веб-страницы, которая представляет собой по крайней мере один кластер в N-мерном пространстве, при этом кластер содержит по крайней мере один N-мерный вектор;

б) обнаруживают аномальный элемент веб-страницы на основании построенной статистической модели веб-страницы, где:

- получают по меньшей мере одним веб-клиентом, реализованным на компьютерном устройстве пользователя, веб-страницу от веб-сервера, при этом веб-страница содержит скрипт, который при выполнении собирает сведения о содержимом по меньшей мере одного элемента веб-страницы на стороне веб-клиента и оправляет собранные сведения с компьютерного устройства пользователя серверу управления;

- выполняют вышеуказанный скрипт с помощью веб-клиента, который собирает сведения о содержимом по меньшей мере одного элемента веб-страницы на стороне веб-клиента и отправляет собранные сведения с компьютерного устройства пользователя серверу управления;

- преобразуют с помощью сервера управления полученные сведения в по меньшей мере один N-мерный вектор, где N-мерный вектор характеризует содержимое по меньшей мере одного элемента веб-страницы;

- сравнивают с помощью сервера управления полученный N-мерный вектор с кластерами статистической модели веб-страницы, где определяют расстояние между полученным N-мерным вектором элемента и центрами всех кластеров статистической модели;

- обнаруживают с помощью сервера управления в результате сравнения аномальный элемент веб-страницы, где аномальным признается элемент, когда выполняется по крайней мере одно из следующих условий:

● расстояние между полученным N-мерным вектором и центрами всех кластеров статистической модели в N-мерном пространстве больше радиусов этих кластеров;

● мера близости между полученным N-мерным вектором и центрами всех кластеров модели в N-мерном пространстве больше порогового значения;

● мера близости между полученным N-мерным вектором и наиболее удаленными от центра кластеров N-мерными векторами кластеров статистической модели в N-мерном пространстве больше порогового значения.

2. Способ по п. 1, в котором получаемая веб-страница при построении статистической модели веб-страницы заведомо не содержит аномальных элементов.

3. Способ по п. 1, в котором элементами веб-страницы, о содержимом которых собирают сведения, являются элементы, по меньшей мере, следующих видов:

а) объекты:

- апплеты;

- скрипты;

- машинный код;

б) формы.

4. Способ по п. 3, в котором собираются сведения о содержимом по меньшей мере двух элементов веб-страницы.

5. Способ по п. 4, в котором элементы относятся к разным видам элементов.

6. Способ по п. 4, в котором элементы относятся к одному виду элементов.

7. Способ по п. 1, в котором для создания кластера используют иерархические методы.

8. Способ по п. 7, в котором кластер создают агломеративным методом, в котором наиболее близкие по расстоянию N-мерные векторы элементов выделяются в кластеры или наиболее близкие по расстоянию кластеры объединяют в один кластер.

9. Способ по п. 8, по которому расстояние: линейное, или евклидово, или обобщенное степенное Минковского, или Чебышева, или Манхэттенское.

10. Способ по п. 8, в котором наиболее близкими признаются векторы, имеющие наименьшее взаимное расстояние.

11. Способ по п. 8, в котором выделяют кластер до тех пор, пока радиус кластера максимально не приблизится к пороговому значению радиуса, где максимальным приближенным является радиус, который при следующем акте выделения кластера превысит пороговое значение радиуса.

12. Способ по п. 8, в котором выделяют кластер до тех пор, пока не останется кластеров или векторов с допустимой мерой близости, где допустимой мерой близости считается мера, не превышающая установленное пороговое значение.

13. Способ по п. 8, в котором наиболее близкими признаются кластеры, имеющие наименьшее расстояние между центрами.

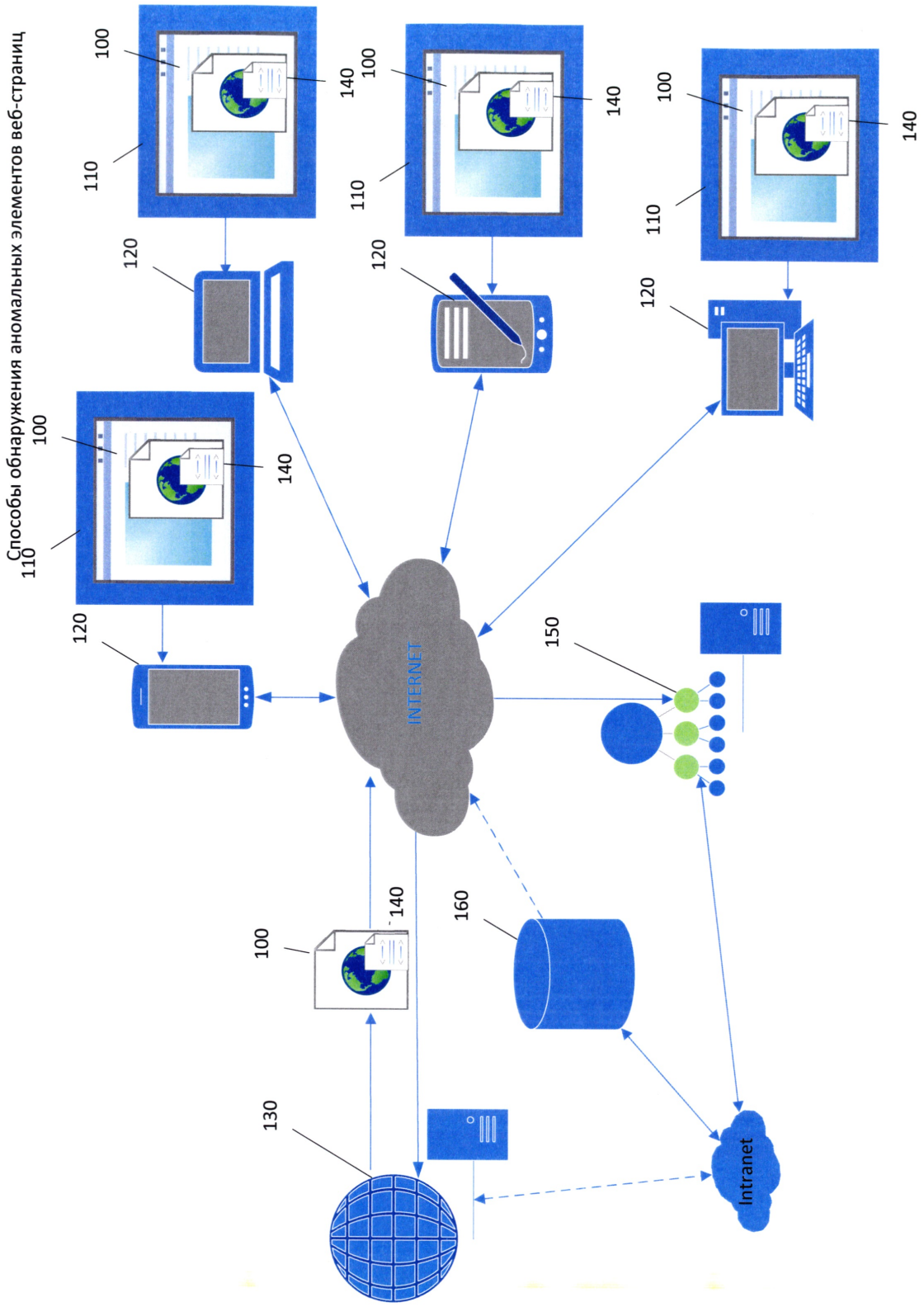
14. Способ по п. 7, в котором кластер создают дивизимным методом, где кластер образуют векторы, взаимное расстояние которых меньше предельно допустимого расстояния, при этом предельная допустимость расстояния определяется пороговым значением.

15. Способ по п. 14, в котором отделяют кластеры до тех пор, пока радиус кластера не станет равным или меньше порогового значения радиуса.

16. Способ по п. 1, в котором для создания кластера используют неиерархические методы.

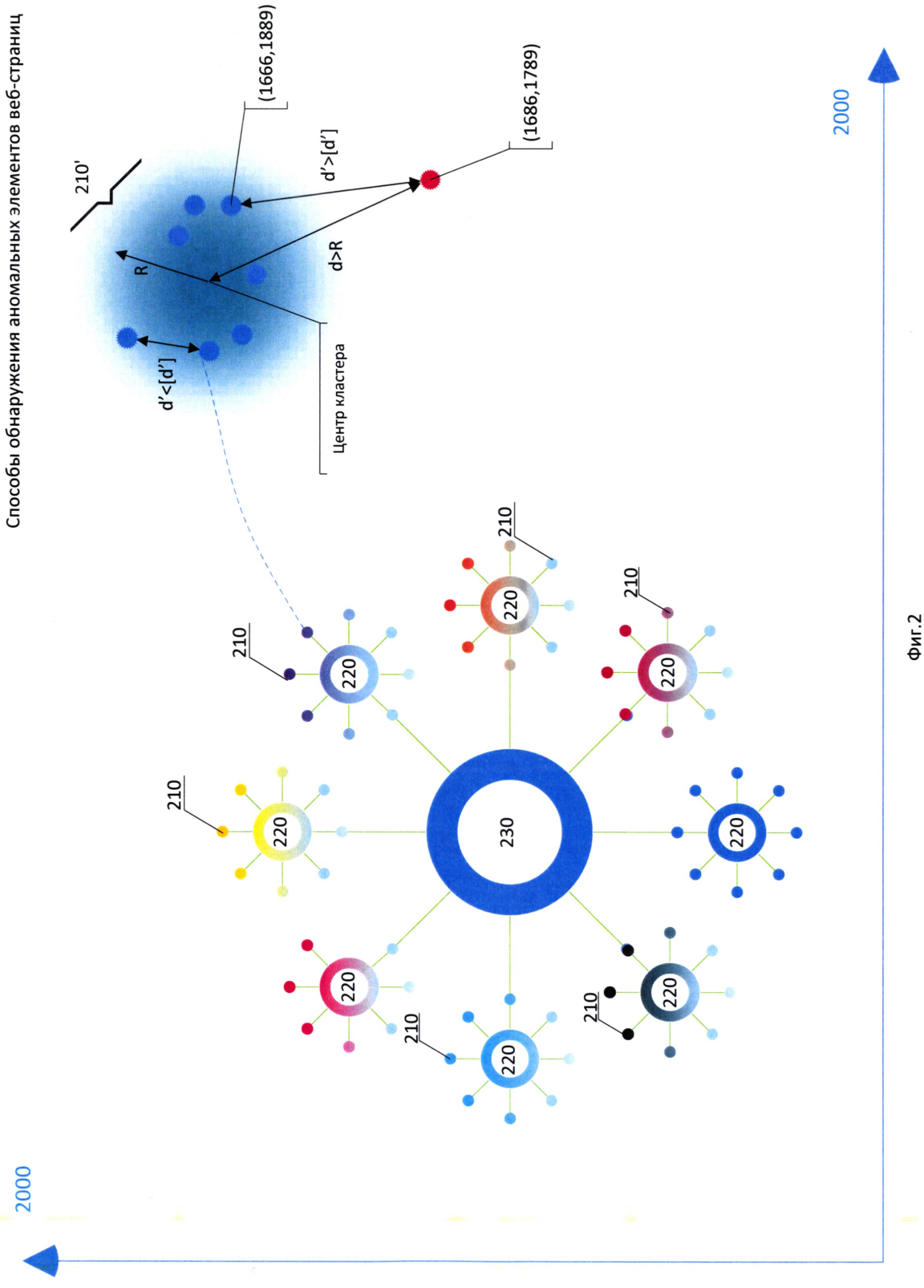
17. Способ по п. 1, в котором аномальным является элемент или группа элементов, N-мерный вектор которых не соответствует построенной статистической модели веб-страницы, а именно не принадлежит ни одному из кластеров модели.

1


Способы обнаружения аномальных элементов веб-страниц
100
110
120
140
150
160
130
INTERNET
Intranet

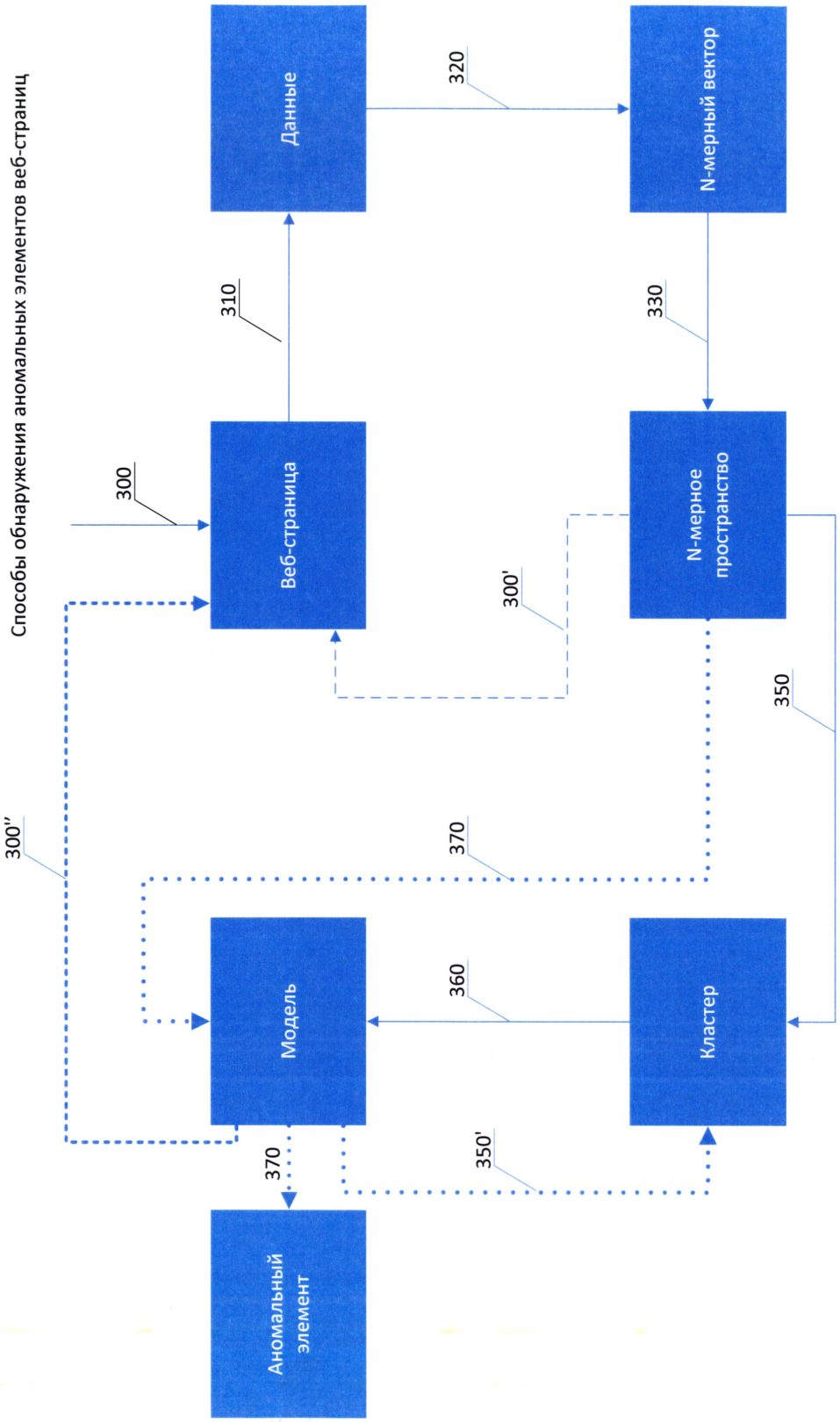
Фиг.1

2


Способы обнаружения аномальных элементов веб-страниц
2000
2000
210'
(1666,1889)
(1686,1789)
d'>[d']
d>R
R
d'<[d']
Центр кластера
210
220
230

Фиг.2

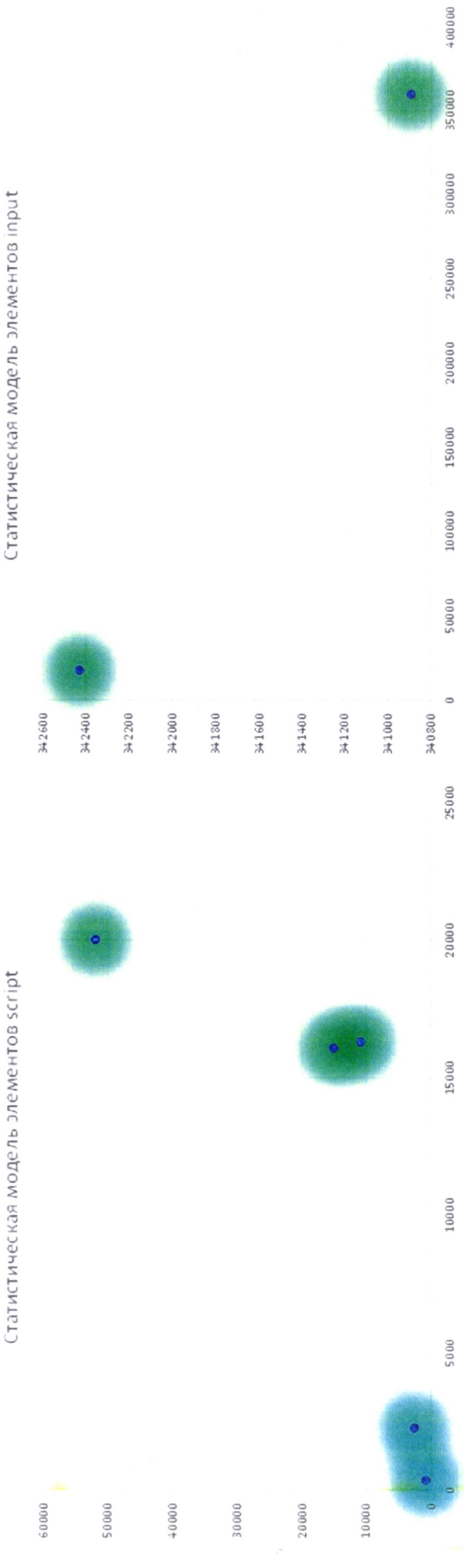
3


Способы обнаружения аномальных элементов веб-страниц
Данные
320
N-мерный вектор
310
330
300
Веб-страница
N-мерное пространство
300'
350
300''
370
Модель
360
Кластер
370
350'
Аномальный элемент

Фиг.3

4

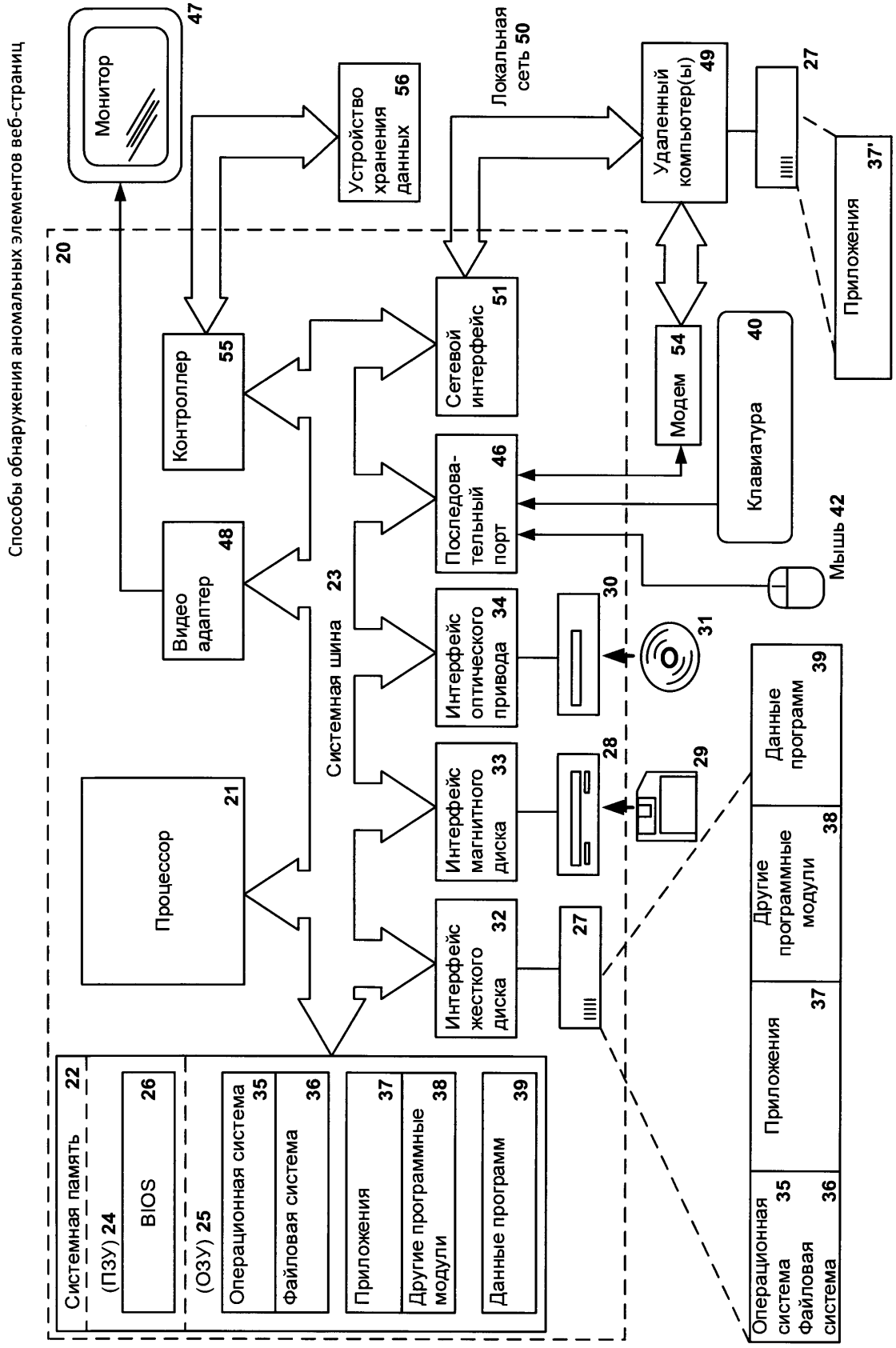
Способы обнаружения аномальных элементов веб-страниц



Фиг.4а

Фиг.4б

Фиг.4в


Монитор 47
Устройство хранения данных 56
Локальная сеть 50
Удаленный компьютер(ы) 49
27
Приложения 37'
20
Контроллер 55
Сетевой интерфейс 51
Модем 54
Клавиатура 40
Видео адаптер 48
Последова-тельный порт 46
Мышь 42
Системная шина 23
Интерфейс оптического привода 34
30
31
Процессор 21
Интерфейс магнитного диска 33
28
29
Данные программ 39
Другие программные модули 38
Интерфейс жесткого диска 32
27
Приложения 37
Операционная система 35
Файловая система 36
Системная память 22
(ПЗУ) 24
BIOS 26
(ОЗУ) 25
Операционная система 35
Файловая система 36
Приложения 37
Другие программные модули 38
Данные программ 39

Фиг.5