



(21) 申请号 201810714303.8

(22) 申请日 2018.07.03

(65) 同一申请的已公布的文献号
申请公布号 CN 110737726 A

(43) 申请公布日 2020.01.31

(73) 专利权人 北京京东尚科信息技术有限公司
地址 100195 北京市海淀区杏石口路65号
西杉创意园四区11号楼东段1-4层西
段1-4层

专利权人 北京京东世纪贸易有限公司

(72) 发明人 孔祥云 王彪 张琪 王少华

(74) 专利代理机构 中原信达知识产权代理有限
责任公司 11219

专利代理师 张一军 张效荣

(51) Int.Cl.

G06F 16/28 (2019.01)

G06F 11/22 (2006.01)

(56) 对比文件

CN 104899141 A, 2015.09.09

CN 107609020 A, 2018.01.19

CN 107861870 A, 2018.03.30

US 2017351746 A1, 2017.12.07

审查员 张皓静

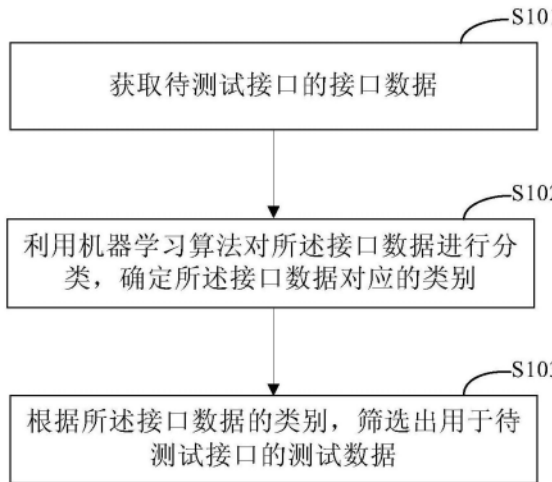
权利要求书2页 说明书9页 附图4页

(54) 发明名称

一种确定待测试接口的测试数据的方法和装置

(57) 摘要

本发明公开了确定待测试接口的测试数据的方法和装置,涉及计算机技术领域。该方法的一具体实施方式包括:获取待测试接口的接口数据;利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别;根据所述接口数据的类别,筛选出用于待测试接口的测试数据。该实施方式克服了现有技术依赖日志确定测试数据导致测试数据的数量大、不真实的技术缺陷,进而达到减少测试所用时间,覆盖典型的测试场景的技术效果。



1. 一种确定待测试接口的测试数据的方法,其特征在于,包括:
获取待测试接口的接口数据;
利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别;
根据所述接口数据的类别,筛选出用于在不同需求或不同业务场景下测试待测试接口的测试数据;

利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别,包括:对所述接口数据做聚类;将聚类后的每一个簇所包含的接口数据对应为一个类别;每个簇对应一个典型场景;

根据所述接口数据的类别,筛选出用于在不同需求或不同业务场景下测试所述待测试接口的测试数据,包括:从每一个类别中筛选出目标时刻采集的接口数据,再从筛选后的每个簇中筛选出一个或多个接口数据,作为用于在对应需求或对应业务场景下测试所述待测试接口的测试数据;将所有类别中筛选后的接口数据,确定为用于测试所述待测试接口的测试数据。

2. 根据权利要求1所述的方法,其特征在于,采用如下方法获取待测试接口的接口数据:

安装Java归档文件包对待测试接口进行采集、或调用第三方工具对待测试接口进行采集。

3. 根据权利要求1所述的方法,其特征在于,对接口数据聚类的模型包括以下至少之一:Spark框架、Tensorflow框架、Scikit-Learn类库。

4. 根据权利要求1所述的方法,其特征在于,根据所述接口数据的类别,筛选出用于在不同需求或不同业务场景下测试所述待测试接口的测试数据之后,还包括:

验证筛选得到的测试数据是否有效。

5. 根据权利要求4所述的方法,其特征在于,测试数据包括:传入数据和传出数据;

验证筛选得到的测试数据是否有效,包括:

将测试数据中的传入数据回放,确定回放后的传出数据;

验证回放后的传出数据与对应的测试数据中的传出数据是否一致;

若回放后的传出数据与对应的测试数据中的传出数据不一致,则确定回放后的传出数据无效。

6. 一种确定待测试接口的测试数据的装置,其特征在于,包括:

接口数据获取模块,用于获取待测试接口的接口数据;

分类模块,用于利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别;

筛选模块,用于根据所述接口数据的类别,筛选出用于在不同需求或不同业务场景下测试待测试接口的测试数据;

利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别,包括:对所述接口数据做聚类;将聚类后的每一个簇所包含的接口数据对应为一个类别;

根据所述接口数据的类别,筛选出用于在不同需求或不同业务场景下测试所述待测试接口的测试数据,包括:从每一个类别中筛选出目标时刻采集的接口数据,再从筛选后的每个簇中筛选出一个或多个接口数据,作为用于在对应需求或对应业务场景下测试所述待测

试接口的测试数据;将所有类别中筛选后的接口数据,确定为用于测试所述待测试接口的测试数据。

7.根据权利要求6所述的装置,其特征在于,采用如下方法获取待测试接口的接口数据:

安装Java归档文件包对待测试接口进行采集、或调用第三方工具对待测试接口进行采集。

8.根据权利要求6所述的装置,其特征在于,其中,对接口数据聚类的模型包括以下至少之一:Spark框架、Tensorflow框架、Scikit-Learn类库。

9.根据权利要求6所述的装置,其特征在于,还包括:验证模块,用于筛选出用于在不同需求或不同业务场景下测试所述待测试接口的测试数据之后验证筛选得到的测试数据是否有效。

10.根据权利要求9所述的装置,其特征在于,测试数据包括:传入数据和传出数据;

验证筛选得到的测试数据是否有效,包括:

将测试数据中的传入数据回放,确定回放后的传出数据;

验证回放后的传出数据与对应的测试数据中的传出数据是否一致;

若回放后的传出数据与对应的测试数据中的传出数据不一致,则确定回放后的传出数据无效;

其中,测试数据包括:传入数据和传出数据。

11.一种确定待测试接口的测试数据的电子设备,其特征在于,包括:

一个或多个处理器;

存储装置,用于存储一个或多个程序,

当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如权利要求1-5中任一所述的方法。

12.一种计算机可读介质,其上存储有计算机程序,其特征在于,所述程序被处理器执行时实现如权利要求1-5中任一所述的方法。

一种确定待测试接口的测试数据的方法和装置

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种确定待测试接口的测试数据的方法和装置。

背景技术

[0002] 在测试接口前,接口的测试数据一般从已有的日志中获取或测试人员自己编写。对于较为复杂的接口,确定接口的测试数据所用的时间比重较大,然而测试过程或者编写测试程序的时间占得比重较小。

[0003] 在实现本发明过程中,发明人发现现有技术中至少存在如下问题:

[0004] 1. 依赖日志确定用于测试接口的测试数据,导致得到的测试数据的量很大,并且重复的测试数据与业务场景耦合度较高。

[0005] 2. 获取测试数据所用的时间较长,造成测试的时间成本偏高。

发明内容

[0006] 有鉴于此,本发明实施例提供一种确定待测试接口的测试数据的方法,能够克服了现有技术依赖日志确定接口数据导致测试数据的数量大、不真实的技术缺陷,进而达到减少测试所用时间,覆盖典型测试场景的技术效果。

[0007] 为实现上述目的,根据本发明实施例的一个方面,提供了一种确定待测试接口的测试数据的方法,包括:

[0008] 获取待测试接口的接口数据;

[0009] 利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别;

[0010] 根据所述接口数据的类别,筛选出用于待测试接口的测试数据。

[0011] 可选地,采用如下方法获取待测试接口的接口数据:

[0012] 安装Java归档文件包对待测试接口进行采集、或调用第三方工具对待测试接口进行采集。

[0013] 可选地,利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别,包括:

[0014] 对所述接口数据做聚类;

[0015] 将聚类后的每一个簇所包含的接口数据对应为一个类别;

[0016] 其中,对接口数据聚类的模型包括以下至少之一:Spark框架、Tensorflow框架、Scikit-Learn类库。

[0017] 可选地,根据所述接口数据的类别,筛选出用于测试所述待测试接口的测试数据,包括:

[0018] 从每一个类别中筛选出一个或多个接口数据;

[0019] 将所有类别中筛选后的接口数据,确定为用于测试所述待测试接口的测试数据。

[0020] 可选地,根据所述接口数据的类别,筛选出用于测试所述待测试接口的测试数据

之后,还包括:

- [0021] 验证筛选得到的测试数据是否有效。
- [0022] 可选地,测试数据包括:传入数据和传出数据;
- [0023] 验证筛选得到的测试数据是否有效,包括:
- [0024] 将测试数据中的传入数据回放,确定回放后的传出数据;
- [0025] 验证回放后的传出数据与对应的测试数据中的传出数据是否一致;
- [0026] 若回放后的传出数据与对应的测试数据中的传出数据不一致,则确定回放后的传出数据无效。
- [0027] 根据本发明实施例的另一个方面,提供一种确定待测试接口的测试数据的装置,包括:
- [0028] 接口数据获取模块,用于获取待测试接口的接口数据;
- [0029] 分类模块,用于利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别;
- [0030] 筛选模块,用于根据所述接口数据的类别,筛选出用于待测试接口的测试数据。
- [0031] 可选地,采用如下方法获取待测试接口的接口数据:
- [0032] 安装Java归档文件包对待测试接口进行采集、或调用第三方工具对待测试接口进行采集。
- [0033] 可选地,利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别,包括:
- [0034] 对所述接口数据做聚类;
- [0035] 将聚类后的每一个簇所包含的接口数据对应为一个类别;
- [0036] 其中,对接口数据聚类的模型包括以下至少之一:Spark框架、Tensorflow框架、Scikit-Learn类库。
- [0037] 可选地,根据所述接口数据的类别,筛选出用于测试所述待测试接口的测试数据,包括:
- [0038] 从每一个类别中筛选出一个或多个接口数据;
- [0039] 将所有类别中筛选后的接口数据,确定为用于测试所述待测试接口的测试数据。
- [0040] 可选地,还包括:验证模块,用于筛选出用于测试所述待测试接口的测试数据之后验证筛选得到的测试数据是否有效。
- [0041] 可选地,测试数据包括:传入数据和传出数据;
- [0042] 验证筛选得到的测试数据是否有效,包括:
- [0043] 将测试数据中的传入数据回放,确定回放后的传出数据;
- [0044] 验证回放后的传出数据与对应的测试数据中的传出数据是否一致;
- [0045] 若回放后的传出数据与对应的测试数据中的传出数据不一致,则确定回放后的传出数据无效。
- [0046] 根据本发明实施例的另一个方面,提供一种确定待测试接口的测试数据的电子设备,包括:
- [0047] 一个或多个处理器;
- [0048] 存储装置,用于存储一个或多个程序,

[0049] 当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现本发明确定待测试接口的测试数据的方法。

[0050] 根据本发明实施例的另一个方面,提供一种计算机可读介质,其上存储有计算机程序,程序被处理器执行时实现本发明确定待测试接口的测试数据的方法。

[0051] 上述发明中的一个实施例具有如下优点或有益效果:

[0052] 本申请采用在待测试接口直接采集接口数据用于测试的技术手段,达到获得真实的接口数据用于测试,使得确定出的测试数据更真实的技术效果,并且由于采集接口数据的时间短,使得获得测试数据的整个过程的时间短,进而达到降低测试时间成本的技术效果,克服了现有技术根据日志确定测试数据导致的测试数据不真实、操作步骤多、确定测试数据的时间成本高的技术缺陷。

[0053] 本申请采用还采用机器学习算法对待测试接口的接口数据进行分类用于测试的技术手段,克服了现有技术采用日志获取测试数据导致的测试数据重复、与业务场景耦合度较高的技术缺陷,进而达到减少测试数据的选择范围、并且覆盖典型的测试场景的技术效果。

[0054] 上述的非惯用的可选方式所具有的进一步效果将在下文中结合具体实施方式加以说明。

附图说明

[0055] 附图用于更好地理解本发明,不构成对本发明的不当限定。其中:

[0056] 图1是根据本发明实施例的确定待测试接口的测试数据的方法的主要流程的示意图;

[0057] 图2是根据本发明实施例的确定待测试接口的测试数据的方法的具体实施例的示意图;

[0058] 图3是根据本发明实施例的确定待测试接口的测试数据的装置的主要模块的示意图;

[0059] 图4是本发明实施例可以应用于其中的示例性系统架构图;

[0060] 图5是适于用来实现本发明实施例的终端设备或服务器的计算机系统的结构示意图。

具体实施方式

[0061] 以下结合附图对本发明的示范性实施例做出说明,其中包括本发明实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识到,可以对这里描述的实施例做出各种改变和修改,而不会背离本发明的范围和精神。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。

[0062] 图1是根据本发明实施例的确定待测试接口的测试数据的方法的主要流程的示意图,如图1所示,为实现上述目的,根据本发明实施例的一个方面,提供了一种确定待测试接口的测试数据的方法,包括:

[0063] 步骤S101、获取待测试接口的接口数据;

[0064] 步骤S102、利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应

的类别；

[0065] 步骤S103、根据所述接口数据的类别，筛选出用于待测试接口的测试数据。

[0066] 本发明提供了一种确定待测试接口的测试数据的方法，其目的在于：测试接口之前，为所述待测试接口准备测试所需要的测试数据。具体地，本实施例的方法可以为待测试接口提供在不同需求或不同业务场景下的测试数据。

[0067] 由于待测试接口一般已经上线、或已经运行过一段时间，故所述待测试接口已经存在一些接口数据。通过将已有的接口数据选为测试数据，可使得测试数据更为真实。其中，在获取接口数据时，可以同时获取接口数据的属性信息。所述属性信息，包括但不限于：获取时间信息、获取测试数据的单位。

[0068] 若将所有接口数据都用于测试，会造成测试成本过高的情况。故在实际测试过程中，仅需采用覆盖典型场景的接口数据进行测试，进而无需将所有的接口数据都用于测试，达到减少测试时间成本的有益效果。由于接口数据的数量总数较大，并且有许多重复、业务场景混杂情况出现，故在可以通过机器学习的算法，将接口数据分类。利用机器学习的算法从分类中筛选出适用于典型场景的接口数据进行测试，达到显著地减少测试数据的选择范围、减少测试成本、减少测试所用时间、覆盖典型的测试场景的技术效果。

[0069] 可选地，采用如下方法获取待测试接口的接口数据：

[0070] 安装Java归档文件包对待测试接口进行采集、或调用第三方工具对待测试接口进行采集。

[0071] 与现有技术从日志中调取接口数据用于测试相比，本发明采用预设的工具直接采集接口数据的技术手段，克服了现有技术过分依赖日志导致操作步骤多、获取的接口数据不真实的技术缺陷，达到获取真实的接口数据的技术效果。

[0072] 可选地，在得到接口数据后，通过异步的传输方式将接口数据发送到数据服务中心，可以避免现有技术采用同步的方式传输接口数据造成对系统的消耗大、传输成本高的技术效果；

[0073] 可选地，可对所述接口数据做标准化处理，使得接口数据更方便机器学习。所述标准化处理，包括但不限于：统一接口数据的单位、剔除接口数据中的异常数据。

[0074] 可选地，利用机器学习算法对所述接口数据进行分类，确定所述接口数据对应的类别，包括：

[0075] 对所述接口数据做聚类；

[0076] 将聚类后的每一个簇所包含的接口数据对应为一个类别；

[0077] 其中，对接口数据聚类的模型包括以下至少之一：Spark框架、Tensorflow框架、Scikit-Learn类库。

[0078] 在可选实施中，采用机器学习的算法对接口数据进行聚类，使得同一类别的接口数据聚集成一个簇(Cluster)。所述Spark框架、Tensorflow框架、Scikit-Learn类库均为现有的、可用于对接口数据聚类的模型，达到方便确定接口数据的分类的技术效果。在本实施例中，通过机器学习将接口数据聚类，进而确定接口数据的分类技术手段，克服了现有技术无法处理接口数据之间存在重复、业务情景混乱的技术问题。

[0079] 可选地，根据所述接口数据的类别，筛选出用于测试所述待测试接口的测试数据，包括：

[0080] 从每一个类别中筛选出一个或多个接口数据;

[0081] 将所有类别中筛选后的接口数据,确定为用于测试所述待测试接口的测试数据。

[0082] 由于测试接口是针对典型的场景进行测试。故可选地,从每个簇中筛选一个或多个接口数据用于代表一个簇所对应的场景。在测试接口时,仅需采用筛选后的测试数据测试接口,就可以覆盖典型的测试场景。本实施例中,通过从每个簇中选择出一个或多个接口数据,大大减少了用于测试的接口数据,克服了现有技术所用的测试数据的数量过大、消耗过大的技术缺陷,达到既可以使得测试更加全面,又可以降低测试成本的技术效果。

[0083] 可选地,将获取接口数据的时间作为筛选测试数据的另一条件。具体地,从每一类接口数据中筛选出目标时刻采集的接口数据,再从筛选后的每个簇中,筛选出一个或多个接口数据作为测试数据。

[0084] 可选地,根据所述接口数据的类别,筛选出用于测试所述待测试接口的测试数据之后,还包括:

[0085] 验证筛选得到的测试数据是否有效。

[0086] 由于待测试接口的逻辑可能会发生变化,故直接采集的接口数据可能不适用于变化后的接口。可选地,在固定的时间段进行验证筛选后的测试数据是否仍然有效,例如,设定验证操作在夜间12点自动启动,可以减小对接口正常运行的影响;也可以将验证操作设定为筛选得到测试数据之后,立即对筛选后的测试数据进行验证。

[0087] 可选地,测试数据包括:传入数据和传出数据;

[0088] 验证筛选得到的测试数据是否有效,包括:

[0089] 将测试数据中的传入数据回放,确定回放后的传出数据;

[0090] 验证回放后的传出数据与对应的测试数据中的传出数据是否一致;

[0091] 若回放后的传出数据与对应的测试数据中的传出数据不一致,则确定回放后的传出数据无效。

[0092] 所述回放是指将测试数据中的传入数据重新放置到接口中运行的过程。通过回放操作,验证回放后的传出数据与测试数据中的传出数据是否一致,进而确定筛选得到测试数据是否出现异常。如果回放后的传出数据发生了变化,则确定该传出数据无效。

[0093] 可选地,标记测试数据中无效的传出数据。然后将标记后的传出数据与正常的传出数据同时作为测试数据,用于方便在测试接口时计算容差率;也可以设定标记后的传出数据为不可用,进而不参与到测试中。

[0094] 下面以一测试接口为具体实施例详细说明本发明的实施方式。

[0095] 图2是根据本发明实施例的确定待测试接口的测试数据的方法的具体实施例的示意图。在确定待测试接口A的测试数据之前,可以在接口程序中安装一个Java归档文件包用于采集接口数据。

[0096] 具体的实施方式如图2所示,包括:

[0097] 步骤S201、利用Java归档文件包,获取待测试接口A的100条接口数据。

[0098] 步骤S202、将所述100条接口数据通过异步传输的方式传输到数据服务中心。

[0099] 步骤S203、在数据服务中心中,对所述100条接口数据进行标准化,并将标准化后的接口数据存入MySQL数据库。其中,MySQL数据库为数据管理提供场所。

[0100] 然后通过步骤S204将MySQL数据库中的对应待测试接口A的接口数据同步到大数

据集市。其中,大数据集市与机器学习训练中心的作用相同,都是为接口数据聚类提供场所。

[0101] 步骤S205、利用Spark模型对接口数据进行聚类。具体地,假设对所述100条接口数据聚类生成10个簇,每个簇对应一个典型场景,并且每个簇中包含10条接口数据。

[0102] 步骤S206、将聚类的结果保存到MySQL数据库。

[0103] 然后,通过步骤S207、从聚类后的每个簇中筛选出测试数据。具体地,将每一个簇中心的接口数据作为测试数据,即可得到10条测试数据,并且每条测试数据都代表一个典型的场景。

[0104] 还包括,步骤S208、验证筛选得到的10条测试数据是否有效。具体地,从MySQL数据库中调用所述筛选后的10条测试数据中的传入数据,将传入数据回放到待测试接口A中,确定该传入数据对应的传出数据是否发生变化。

[0105] 假设验证结果为:所述10条传出数据中有3条与原传出数据不同,则将所述3条异常的传出数据标记。然后,通过步骤S209将验证的结果存储到MySQL数据库中。在对待测试接口A进行测试时,可以直接调用数据库中的剩余的7条有效的测试数据进行测试。

[0106] 图3是根据本发明实施例的确定待测试接口的测试数据的装置的主要模块的示意图,如图3所示,根据本发明实施例的另一个方面,提供一种确定待测试接口的测试数据的装置300,包括:

[0107] 接口数据获取模块301,用于获取待测试接口的接口数据;

[0108] 分类模块302,用于利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别;

[0109] 筛选模块303,用于根据所述接口数据的类别,筛选出用于待测试接口的测试数据。

[0110] 可选地,采用如下方法获取待测试接口的接口数据:

[0111] 安装Java归档文件包对待测试接口进行采集、或调用第三方工具对待测试接口进行采集。

[0112] 可选地,利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别,包括:

[0113] 对所述接口数据做聚类;

[0114] 将聚类后的每一个簇所包含的接口数据对应为一个类别;

[0115] 其中,对接口数据聚类的模型包括以下至少之一:Spark框架、Tensorflow框架、Scikit-Learn类库。

[0116] 可选地,根据所述接口数据的类别,筛选出用于测试所述待测试接口的测试数据,包括:

[0117] 从每一个类别中筛选出一个或多个接口数据;

[0118] 将所有类别中筛选后的接口数据,确定为用于测试所述待测试接口的测试数据。

[0119] 可选地,还包括:验证模块,用于筛选出用于测试所述待测试接口的测试数据之后验证筛选得到的测试数据是否有效。

[0120] 可选地,测试数据包括:传入数据和传出数据;

[0121] 验证筛选得到的测试数据是否有效,包括:

[0122] 将测试数据中的传入数据回放,确定回放后的传出数据;

[0123] 验证回放后的传出数据与对应的测试数据中的传出数据是否一致;

[0124] 若回放后的传出数据与对应的测试数据中的传出数据不一致,则确定回放后的传出数据无效;

[0125] 图4示出了可以应用本发明实施例的确定待测试接口的测试数据的方法或确定待测试接口的测试数据的装置的示例性系统架构400。

[0126] 如图4所示,系统架构400可以包括终端设备401、402、403,网络404和服务器405。网络404用以在终端设备401、402、403和服务器405之间提供通信链路的介质。网络404可以包括各种连接类型,例如有线、无线通信链路或者光纤电缆等等。

[0127] 用户可以使用终端设备401、402、403通过网络404与服务器405交互,以接收或发送消息等。终端设备401、402、403上可以安装有各种通讯客户端应用,例如购物类应用、网页浏览器应用、搜索类应用、即时通信工具、邮箱客户端、社交平台软件等(仅为示例)。

[0128] 终端设备401、402、403可以是具有显示屏并且支持网页浏览的各种电子设备,包括但不限于智能手机、平板电脑、膝上型便携计算机和台式计算机等等。

[0129] 服务器405可以是提供各种服务的服务器,例如对用户利用终端设备401、402、403所浏览的购物类网站提供支持的后台管理服务器(仅为示例)。后台管理服务器可以对接收到的产品信息查询请求等数据进行分析等处理,并将处理结果(例如目标推送信息、产品信息--仅为示例)反馈给终端设备。

[0130] 需要说明的是,本发明实施例所提供的确定待测试接口的测试数据方法一般由服务器405执行,相应地,确定待测试接口的测试数据装置一般设置于服务器405中。

[0131] 应该理解,图4中的终端设备、网络和服务器的数目仅仅是示意性的。根据实现需要,可以具有任意数目的终端设备、网络和服务器。

[0132] 下面参考图5,其示出了适于用来实现本发明实施例的终端设备的计算机系统500的结构示意图。图5示出的终端设备仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0133] 如图5所示,计算机系统500包括中央处理单元(CPU)501,其可以根据存储在只读存储器(ROM)502中的程序或者从存储部分508加载到随机访问存储器(RAM)503中的程序而执行各种适当的动作和处理。在RAM 503中,还存储有系统500操作所需的各种程序和数据。CPU 501、ROM 502以及RAM 503通过总线504彼此相连。输入/输出(I/O)接口505也连接至总线504。

[0134] 以下部件连接至I/O接口505:包括键盘、鼠标等的输入部分506;包括诸如阴极射线管(CRT)、液晶显示器(LCD)等以及扬声器等的输出部分507;包括硬盘等的存储部分508;以及包括诸如LAN卡、调制解调器等的网络接口卡的通信部分509。通信部分509经由诸如因特网的网络执行通信处理。驱动器510也根据需要连接至I/O接口505。可拆卸介质511,诸如磁盘、光盘、磁光盘、半导体存储器等等,根据需要安装在驱动器510上,以便于从其上读出的计算机程序根据需要被安装入存储部分508。

[0135] 特别地,根据本发明公开的实施例,上文参考流程图描述的过程可以被实现为计算机软件程序。例如,本发明公开的实施例包括一种计算机程序产品,其包括承载在计算机可读介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。在

这样的实施例中,该计算机程序可以通过通信部分509从网络上被下载和安装,和/或从可拆卸介质511被安装。在该计算机程序被中央处理单元(CPU)501执行时,执行本发明的系统中限定的上述功能。

[0136] 需要说明的是,本发明所示的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是一——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器(CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本发明中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本发明中,计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:无线、电线、光缆、RF等等,或者上述的任意合适的组合。

[0137] 附图中的流程图和框图,图示了按照本发明各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,上述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意,框图或流程图中的每个方框、以及框图或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0138] 描述于本发明实施例中所涉及到的模块可以通过软件的方式实现,也可以通过硬件的方式来实现。所描述的模块也可以设置在处理器中,例如,可以描述为:一种处理器包括发送模块、获取模块、确定模块和第一处理模块。其中,这些模块的名称在某种情况下并不构成对该模块本身的限定,例如,发送模块还可以被描述为“向所连接的服务端发送图片获取请求的模块”。

[0139] 作为另一方面,本发明还提供了一种计算机可读介质,该计算机可读介质可以是上述实施例中描述的设备中所包含的;也可以是单独存在,而未装配入该设备中。上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被一个该设备执行时,使得该设备包括:

[0140] 获取待测试接口的接口数据;

[0141] 利用机器学习算法对所述接口数据进行分类,确定所述接口数据对应的类别;

[0142] 根据所述接口数据的类别,筛选出用于待测试接口的测试数据。

[0143] 根据本发明实施例的技术方案,从而达到如下有益效果:

[0144] 本申请采用在待测试接口直接采集接口数据用于测试的技术手段,达到获得真实的接口数据用于测试,使得确定出的测试数据更真实的技术效果,并且由于采集接口数据的时间短,使得获得测试数据的整个过程的时间短,进而达到降低测试时间成本的技术效果,克服了现有技术根据日志确定测试数据导致的测试数据不真实、操作步骤多、确定测试数据的时间成本高的技术缺陷。

[0145] 本申请采用还采用机器学习算法对待测试接口的接口数据进行分类用于测试的技术手段,克服了现有技术采用日志获取测试数据导致的测试数据重复、与业务场景耦合度较高的技术缺陷,进而达到减少测试数据的选择范围、并且覆盖典型的测试场景的技术效果。

[0146] 上述具体实施方式,并不构成对本发明保护范围的限制。本领域技术人员应该明白的是,取决于设计要求和因素,可以发生各种各样的修改、组合、子组合和替代。任何在本发明的精神和原则之内所作的修改、等同替换和改进等,均应包含在本发明保护范围之内。

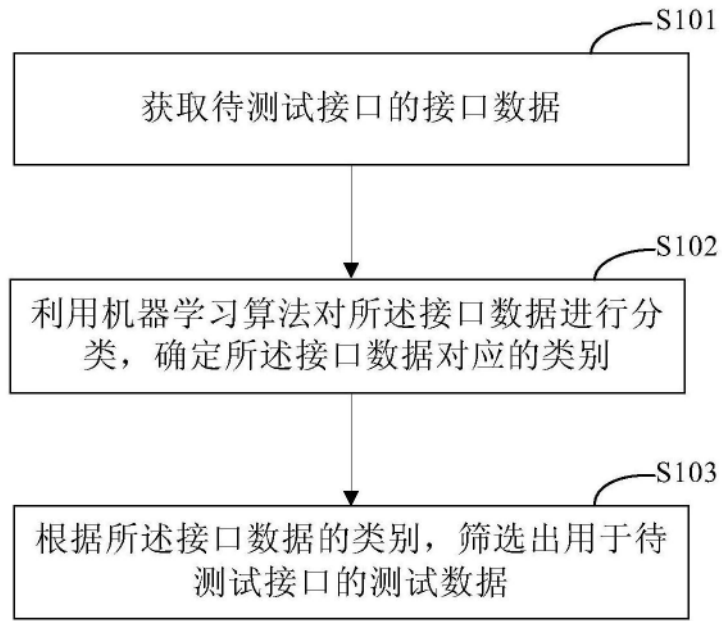


图1

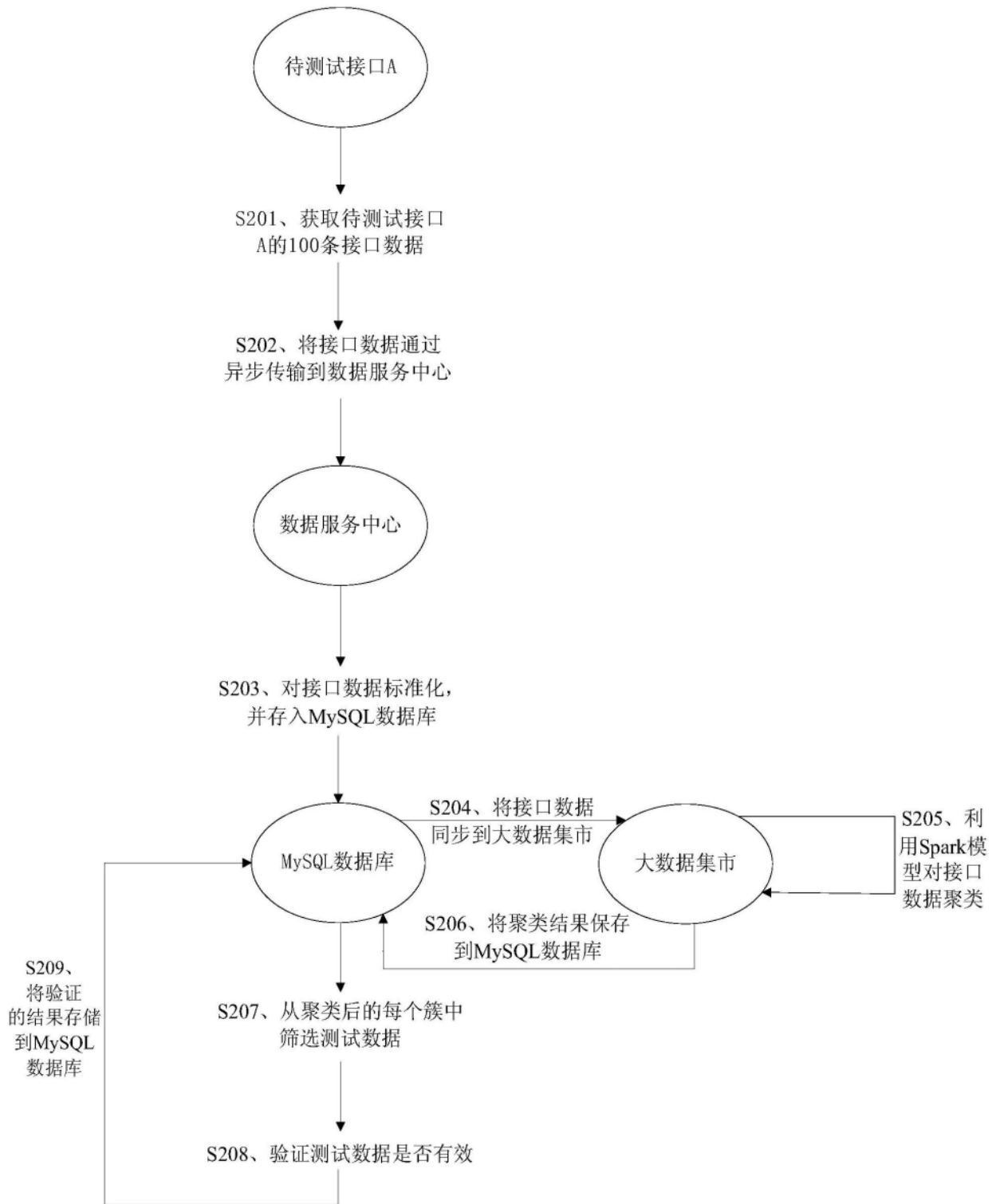


图2

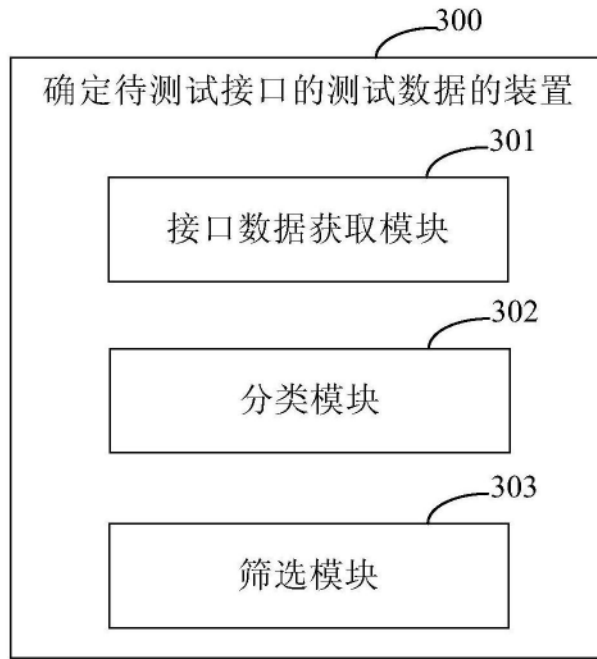


图3

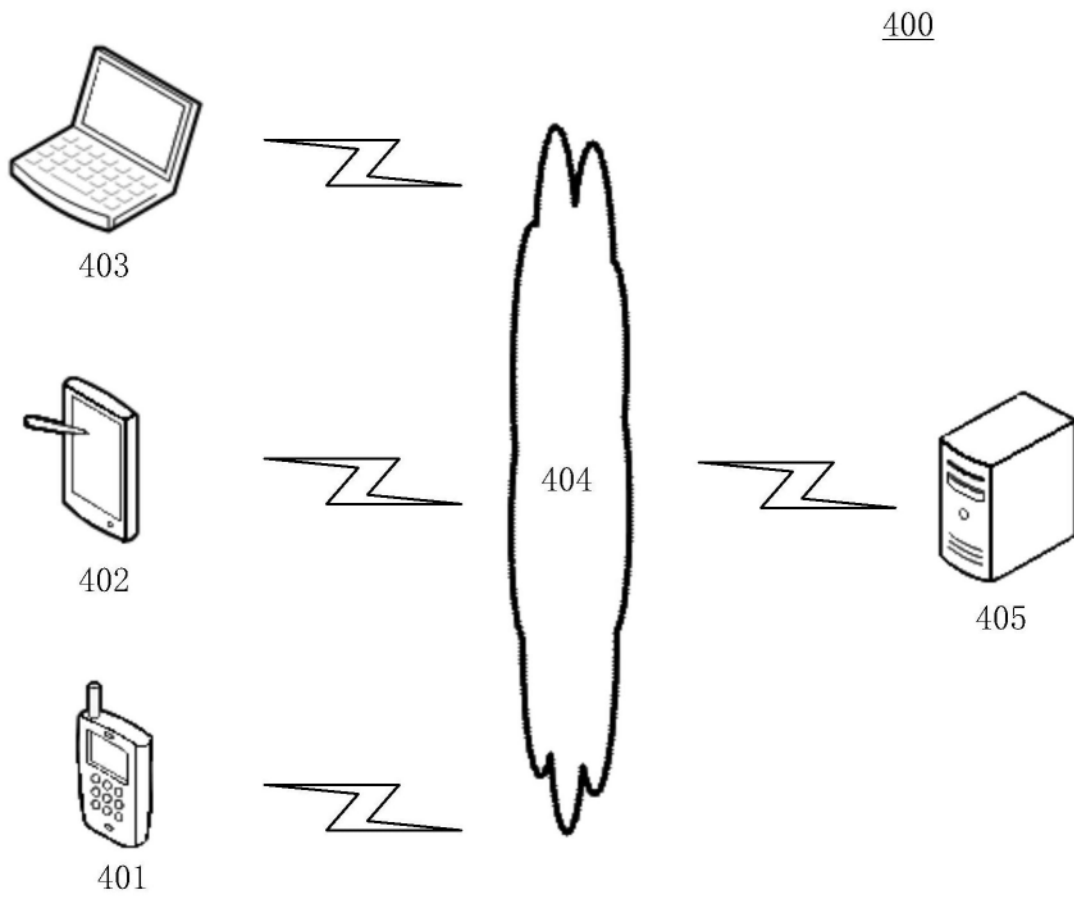


图4

500

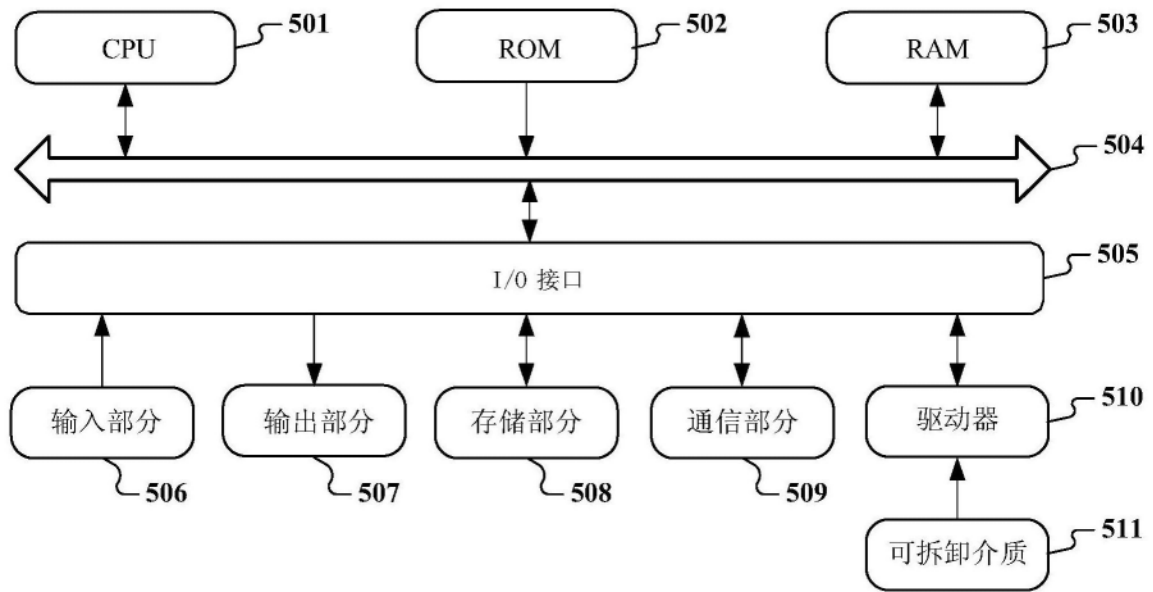


图5