(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 July 2022 (14.07.2022)

WIPO | PCT

(10) International Publication Number
WO 2022/150062 A1

(51) International Patent Classification:
G06F 17/18 (2006.01)

(21) International Application Number:
PCT/US2021/043490

(22) International Filing Date:
28 July 2021 (28.07.2021)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

63/135,314	08 January 2021 (08.01.2021)	US
21187821.0	26 July 2021 (26.07.2021)	EP
117365	26 July 2021 (26.07.2021)	PT
17/386,291	27 July 2021 (27.07.2021)	US

(71) Applicant: **FEEDZAI - CONSULTADORIA E INOVACAO TECNOLOGICA, S.A.** [PT/US]; Rua Pedro Nunes - IPN, 3030-199 Coimbra (PT).

(72) Inventors: **SAMPAIO, Marco, Oliveira, Pena**; Rua Pedro Nunes - IPN, 3030-199 Coimbra (PT). **SILVA, Pedro, Cardoso, Lessa E**; Rua Pedro Nunes - IPN, 3030-199 Coimbra (PT). **AZEVEDO, Joao, Dias, Conde**; Rua Pedro Nunes - IPN, 3030-199 Coimbra (PT). **MOREIRA, Ricardo, Miguel De Oliveira**; Rua Pedro Nunes - IPN, 3030-199 Coimbra (PT). **ASCENSAO, Joao, Tiago, Barriga, Negra**; Rua Pedro Nunes - IPN, 3030-199 Coimbra (PT). **BIZARRO, Pedro, Gustavo, Santos, Rodrigues**; Rua Pedro Nunes - IPN, 3030-199 Coimbra (PT). **GOMES, Ana, Sofia, Leal**; Rua Pedro Nunes - IPN, 3030-199 Coimbra (PT). **OLIVEIRINHA, Joao, Miguel, Forte**; Rua Pedro Nunes - IPN, 3030-199 Coimbra (PT).

(74) Agent: **HU, Zhong-Min**; Van Pelt, Yi & James, LLP, 10050 N. Foothill Blvd., Suite 200, Cupertino, CA 95014 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, IT, JO, JP, KE, KG, KH, KN,

(54) Title: AUTOMATED FEATURE MONITORING FOR DATA STREAMS

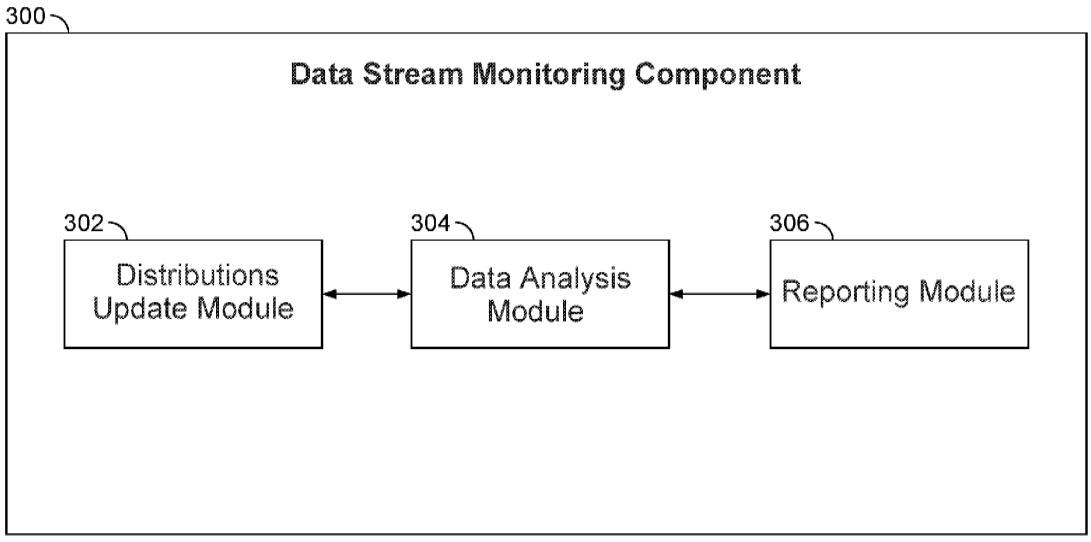

FIG. 3A

(57) Abstract: One or more events of a data stream are received. For each feature of a set of features, the one or more events are used to update a corresponding distribution of data from the data stream. For each feature of the set of features, the corresponding updated distribution and a corresponding reference distribution are used to determine a corresponding divergence value. For each feature of the set of features, the corresponding determined divergence value and a corresponding distribution of divergences are used to determine a corresponding statistical value. Using the statistical values each corresponding to a different feature of the set of features, a statistical analysis is performed to determine a result associated with a likelihood of data drift detection.

WO 2022/150062 A1

[Continued on next page]

KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

AUTOMATED FEATURE MONITORING FOR DATA STREAMS

CROSS REFERENCE TO OTHER APPLICATIONS

[0001] This application claims priority to U.S. Provisional Patent Application No. 63/135,314 entitled AUTOMATED FEATURE MONITORING FOR DATA STREAMS filed January 08, 2021 which is incorporated herein by reference for all purposes.

[0002] This application claims priority to Portugal Provisional Patent Application No. 117365 entitled AUTOMATED FEATURE MONITORING FOR DATA STREAMS filed July 26, 2021, which is incorporated herein by reference for all purposes.

[0003] This application claims priority to European Patent Application No. 21187821.0 entitled AUTOMATED FEATURE MONITORING FOR DATA STREAMS filed July 26, 2021, which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

[0004] Data analysis is a process for obtaining raw data and converting it into information useful for informing conclusions or supporting decision-making. Typical data analysis steps include collecting data, organizing data, manipulating data, and summarizing data. Oftentimes, data analysis is performed automatically by computer systems on datasets that are too large and complex for analysis by a human. In many scenarios, a goal of automated data analysis is to determine whether a given collection of data is substantially similar, in a specified and quantifiable sense, to another collection of data. Accomplishing this goal requires comparing relevant data patterns and structures, which can be challenging to do efficiently, particularly in the context of automated data analysis of large amounts of data. Thus, it would be beneficial to develop techniques directed toward efficient and robust characterization of similarity between datasets.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

[0006] Figure 1 is a high-level block diagram illustrating an embodiment of a framework for performing feature monitoring.

[0007] Figure 2A is a block diagram illustrating an embodiment of a build reference component.

[0008] Figure 2B is a diagram illustrating example data patterns generated by a build reference component.

[0009] Figure 2C is a diagram illustrating example data outputs of a build reference component.

[0010] Figure 3A is a block diagram illustrating an embodiment of a data stream monitoring component.

[0011] Figure 3B is a diagram illustrating examples of data analysis associated with a data stream monitoring component.

[0012] Figure 4 is a flow diagram illustrating an embodiment of a process for generating divergence distributions for automated data analysis.

[0013] Figure 5 is a flow diagram illustrating an embodiment of a process for performing automated feature monitoring of a data stream.

[0014] Figure 6 is a flow diagram illustrating an embodiment of a process for performing statistical analysis to determine whether to generate an alarm associated with feature monitoring.

[0015] Figure 7 is a functional diagram illustrating a programmed computer system.

DETAILED DESCRIPTION

[0016] The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to

perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

[0017] A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

[0018] Automated feature monitoring for data streams is disclosed. One or more events of a data stream are received. For each feature of a set of features, the one or more events are used to update a corresponding distribution of data from the data stream. For each feature of the set of features, the corresponding updated distribution and a corresponding reference distribution are used to determine a corresponding divergence value. For each feature of the set of features, the corresponding determined divergence value and a corresponding distribution of divergences are used to determine a corresponding statistical value. Using the statistical values each corresponding to a different feature of the set of features, a statistical analysis is performed to determine a result associated with a likelihood of data drift detection.

[0019] As used herein, a data stream refers to a sequence of digitally encoded coherent signals used to transmit or receive information. A data stream can be comprised of logically separate data elements referred to as events. Stated alternatively, events are distinct data elements (e.g., arriving in the data stream at distinct moments in time) utilized as fundamental units for automated data analysis. Each data element can be comprised of several data items that can be characterized to determine properties (also referred to herein as features) useful for data analysis. Stream processing systems (also referred to as data stream processing systems) are computer systems that analyze data streams. An example of a stream processing system is a fraud detection system that receives information associated with user transactions and determines whether identity theft (e.g., wrongful use of personal information of another) has likely occurred. Many stream processing systems (e.g., fraud detection systems) are real-time stream processing systems. Real-

time processing refers to processing that occurs at the time data is generated or published or processing that occurs as frequently as necessary for a particular use case.

[0020] The initial configuration of real-time stream processing systems typically assumes that future data flowing through the systems approximately follows the same patterns as previously received data. Real-time stream processing systems typically require reconfiguration over time because even though the systems may initially perform well, over time, due to data drift, a static configuration may result in performance deterioration. Data drift refers to changes in data patterns and/or structure (e.g., a change in the distribution of data) over time. Data drift can occur due to expected or unexpected factors. With respect to fraud detection, an example of expected data drift is seasonal changes in user purchasing patterns, and an example of unexpected data drift is a change in data patterns as a result of technical issues that corrupt or otherwise alter the observed data. Data drift can be gradual or sudden and may occur in specific features of data or collectively for all or almost all the features. Data drift can be a significant problem. For example, machine learning (ML) models responsible for predictive tasks might make inaccurate decisions because of extrapolations from short-sighted training set observations (that do not account for future data pattern changes).

[0021] Accurate and timely detection of data drift can allow for measures to be taken to mitigate the negative effects of data drift. This is a challenging task that is made easier in systems where a clear performance target and immediate labels allow performance evaluation in real-time. However, this is rarely the case, and in many domains, such as the detection of fraud, labels cannot be obtained in real-time. Several weeks or even months may pass until labels are available. In this scenario, unsupervised methods are required to detect data drift timely and accelerate corrective action. Furthermore, in many streaming data scenarios, a very large volume of data is generated every day at very high rates. Thus, if real-time monitoring is required, lightweight solutions are needed to avoid high additional computational and storage costs.

[0022] Techniques to automate the detection and alarming of data drifts based on monitoring the distribution of data features, referred to herein as feature monitoring, are disclosed herein. Technological advantages (e.g., that improve the functioning of computer systems) of the solutions disclosed herein include lower memory footprint (e.g., by using histograms to summarize the data distributions) and reduced computational cost (e.g., by using estimation methods that support recursive updates, resulting in low real-time latencies). In various embodiments, a multivariate data-driven statistical test using a reference data period is utilized. In various embodiments, histograms are utilized to summarize data distributions (also referred to herein

simply as distributions). However, it is not a requirement that data distributions be represented by histograms. Other data distribution representations, such as parameterized functions (functions that have parameters fixed to data), may also be utilized. It is understood that, as used herein, a reference to histograms also contemplates other data distribution representations. Furthermore, several types of histograms are possible. As used herein, data distribution representations may be referred to as distribution representations or simply distributions.

[0023] A framework for feature monitoring is disclosed herein. The framework includes a build reference component to build reference distributions for the data features to be monitored as well as a distribution of their reference statistical fluctuations (e.g., measured by a divergence between distributions) based on a reference dataset or portion thereof of the reference dataset. In some embodiments, the framework includes a pre-processing module to extract a set of features to monitor. The features can be original features (identity mapping) or a new set of decorrelated or independent features to be monitored. In various embodiments, for each feature, the build reference component builds a reference distribution from the reference dataset, samples the reference dataset to generate corresponding subsets, builds distributions for the subsets, determines divergences between the subset distributions and the reference distribution, and builds a distribution of the divergences. In some embodiments, the build reference component outputs a triplet of distribution representations per feature, e.g.: 1) a reference histogram built from the reference dataset, 2) a histogram built from a sample of the reference dataset, and 3) a histogram of divergence values.

[0024] The framework also includes a data stream monitoring component to process a series of time-ordered events and monitor a set of features against corresponding input reference distributions (e.g., provided by the build reference component). The data stream monitoring component generates alarms (with corresponding reports). In various embodiments, for each feature of a set of features, the data stream monitoring component processes streaming data (e.g., event by event, in batches of events, etc.) to build a target distribution that corresponds to an input reference distribution. In some embodiments, for each feature, the input reference distribution is received by the data stream monitoring component from the build reference component. In various embodiments, for each feature, the data stream monitoring component computes a divergence between the target distribution and the input reference distribution and computes a statistical value (e.g., a p-value). This results in a set of statistical values that can be ranked, scaled, and utilized in a multivariate statistical test with a suitable threshold to identify features for which to raise an alarm. Alarmed features can be mapped to a set of interpretable features, ranked by importance to explain the alarm, and reported (along with other information, such as when the alarm occurred).

[0025] Technological advantages of the techniques disclosed herein include: 1) lightweight computing in terms of both computational complexity as well as space complexity by using a constant memory reference state and a constant memory streaming state updated via a recursive technique and 2) computational savings from reduced retraining of models (e.g., machine learning models) by using data drift detection to determine the most suitable time to retrain models and avoiding unnecessary retraining actions.

[0026] Figure 1 is a high-level block diagram illustrating an embodiment of a framework for performing feature monitoring. Framework 100 includes build reference component 104, data stream monitoring component 108, and various data that is received or generated. Framework 100 illustrates data flow associated with build reference component 104 and data stream monitoring component 108. Reference data 102 is received by build reference component 104. Build reference component 104 generates distribution data 106, which is transmitted to data stream monitoring component 108. Data stream monitoring component 108 receives streaming data 110. Based at least in part on distribution data 106 and streaming data 110, data stream monitoring component 108 generates monitoring results data 112.

[0027] Framework 100 involves monitoring distributions of features of a dataset in a streaming production environment (feature monitoring). The features that are monitored may comprise several fields (e.g., categorical or numerical), which may be provided directly or derived/calculated using other collected fields for incoming events. Framework 100 does not rely on any assumptions on what the features are used for in the production environment. In some applications, the features are used by a pre-trained ML model for predictive tasks. For example, with respect to credit card (or other) fraud detection, the ML model may be a model that determines whether a transaction (e.g., a transaction made via the Internet) corresponding to an incoming event is legitimate or not.

[0028] In various embodiments, build reference component 104 is responsible for producing data-driven references for distributions to be monitored, along with estimates of expected levels of their statistical fluctuations. In the example illustrated, build reference component 104 receives reference data 102, which is a data source with a plurality of events used to define a reference distribution of its data. Reference data 102 may correspond to a fixed period before deployment (e.g., a training period of an ML model for a predictive task). Reference data 102 can also correspond to any past period, including after deployment (e.g., if the feature monitoring framework raises an alarm on a feature that changed permanently, a system

administrator may want to refresh reference data 102 with the latest data). In various embodiments, reference data 102 corresponds to an extended period, e.g., several weeks or months of data.

[0029] In some embodiments, build reference component 104 performs feature space decorrelation/reduction. In some applications, a set of features may be highly correlated and have strong dependencies. Hence, the effective dimensionality of the monitored feature space may be much smaller. In those scenarios, it may be desirable to work with a different set of features that are less correlated or even independent (e.g., if build reference component 104 results are used in a data stream monitoring component that assumes feature independence). Therefore, in those cases, build reference component 104 may utilize a sub-component to apply a decorrelation or an independent features extraction technique to reduce the dimensionality of the feature space.

[0030] In various embodiments, for each feature to be monitored, build reference component 104 produces a representation of a reference distribution of the data of reference data 102 (e.g., encoded in a histogram). In various embodiments, for each feature to be monitored, several reference distributions (e.g., histograms), corresponding to different time windows within the time period spanned by reference data 102, are computed. Stated alternatively, reference data 102 can be sampled. This can be used to characterize the distribution of the data fluctuations in shorter periods within reference data 102. For example, suppose reference data 102 spans six months. In that case, each distribution corresponding to a sample (sampled distribution) can be a distribution of a feature in a one-week, one-month, etc. sub-period. The sub-periods may be sampled randomly. In various embodiments, for each feature, build reference component 104 computes a measure of divergence between each sampled distribution and the corresponding reference distribution, resulting in an output that is used to produce a representation of the distribution of divergence values of the feature (e.g., a histogram of divergence values for the feature). In various embodiments, distribution data 106 includes this representation of the distribution of divergence values for each feature, as well as the corresponding reference distribution representation for each feature. In addition, for each feature, an example sampled distribution representation may be included. A detailed example of build reference component 104 is described with respect to Figures 2A – 2C.

[0031] Framework 100 illustrates data stream monitoring component 108 processing streaming data 110. In various embodiments, this processing occurs in real-time. For each arriving event or arriving batch of events, data stream monitoring component 108 computes new estimates for distributions of monitored feature values. In various embodiments, the new estimates are compared with data received from build reference component 104, in the form of distribution data

106, to determine if a data drift has likely occurred and thus an alarm should be triggered. In various embodiments, distribution data 106 includes a set of distributions (e.g., represented by histograms), one per feature, to serve as an initial state, a corresponding set of reference distributions, and a corresponding set of divergence distributions (e.g., also represented by histograms) to characterize distributions of divergence values for each feature. Specifically, in various embodiments, data stream monitoring component 108 performs the following for each monitored feature: 1) updates a distribution representation taking into account one or more new values, 2) computes a divergence measure between the updated distribution representation and a reference distribution representation, and 3) computes a statistical value (e.g., a p-value) by comparing the computed divergence measure with a reference distribution of divergence values. In various embodiments, data stream monitoring component 108 then performs a multivariate statistical test that combines various statistical values (e.g., p-values) corresponding to different features to determine if the computed set of divergence measures are within expected statistical fluctuations. If the divergence measures exceed an expected level of fluctuations, an alarm is raised. In various embodiments, the alarm includes explanatory information that summarizes information associated with the computed statistical values (e.g., the p-values). The explanatory information may also include ranking of features by importance to explain the alarm, which may involve transformations back to an original feature set if a decorrelation or independent components extraction or feature space dimensionality reduction procedure was applied. In the example illustrated, data stream monitoring component 108 outputs monitoring results 112, which can include the alarm explanatory information. Additional actions can also occur when an alarm is triggered. For example, ML model retraining, build reference component 104 retraining to suppress continuing alarms (e.g., if a permanent drift occurs), threshold adjustments, or system administrator notifications can occur. For efficiency reasons, the computation of divergence values may not occur for every event. In this scenario, feature distribution representations can still be updated for every event of streaming data 110, but the computation of divergences between updated distributions and reference distributions would only occur after a specified number of events have passed. A detailed example of data stream monitoring component 108 is described with respect to Figures 3A – 3B.

[0032] Figure 2A is a block diagram illustrating an embodiment of a build reference component. In some embodiments, build reference component 200 is build reference component 104 of framework 100 of Figure 1. In some embodiments, build reference component 200 (including its sub-components) is comprised of computer program instructions that are executed on a general-purpose processor, e.g., a central processing unit (CPU), of a programmed computer system. Figure 7 illustrates an example of a programmed computer system. It is also possible for

the logic of build reference component 200 to be executed on other hardware, e.g., executed using an application-specific integrated circuit (ASIC) or a field programmable gate array (FPGA).

[0033] In the example shown, build reference component 200 includes feature space reduction module 202 and feature handling component 204, which is comprised of reference distribution build module 206 and sampling and divergence distribution build module 208. The components and organization shown in Figure 2A is merely illustrative and not restrictive. Other configurations are also possible. For example, in embodiments without decorrelation or independent components extraction or feature space dimensionality reduction, feature space reduction module 202 is not included. In the example shown, portions of the communication path between the components are shown. Other communication paths may exist, and the example of Figure 2A has been simplified to illustrate the example clearly. Although single instances of components have been shown to simplify the diagram, additional instances of any of the components shown in Figure 2A may exist. Components not shown in Figure 2A may also exist. Build reference component 200 estimates reference distributions for each of a set of features analyzed. In various embodiments, this estimation is based on a fixed data source comprising several events. In some embodiments, the data source is reference data 102 of Figure 1.

[0034] In some embodiments, data is first processed by feature space reduction module 202. Oftentimes, in ML applications, features available to describe underlying statistical processes have strong correlations. Features may originate from separate systems where they are collected or computed from related transformations on a set of raw fields to capture different views of underlying data processes. For example, in fraud detection, three different profile features such as a sum of transaction amounts in the last 5 minutes (feature 1), a number of transactions in the same period (feature 2), and a sum of transaction amounts over 1 minute (feature 3) can encode overlapping information. In this example, the distinction between them is either the type of aggregation (sum vs. count) or the window size (1 minute vs. 5 minutes). All of these three features originate from the same raw field, and thus they are not independent of each other. In applications where the output of build reference component 200 is used for monitoring in a data stream monitoring component (e.g., data stream monitoring component 300 of Figure 3A), these observations show that monitoring the full set of features may result in entangled signals or redundancies in the alarm information. This may mask subdominant components or introduce additional noise. Furthermore, in the data stream monitoring component, some multivariate tests may assume independence between the various hypotheses being tested to achieve a higher statistical power. This assumption motivates transforming the feature space to a new space of

independent components, which may result in reducing the number of components being characterized by build reference component 200. Feature space reduction can be generically defined as a transformation taking as input a set of initial features X_1, ..., X_K and mapping them to a set of final features to be monitored F_1, ..., F_N, with N less than or equal to K. Example options for performing feature space reduction include, but are not limited to, the techniques described below.

[0035] Principal component analysis (PCA) reduction may be utilized to perform feature space reduction. Given a set of features, PCA assumes that the data is distributed as a multivariate Gaussian distribution. PCA re-centers and rotates the coordinate axes so that the origin becomes the center of mass of the estimated Gaussian, and the axes become aligned with its covariance matrix, resulting in new features that are linearly uncorrelated. A dimensionality reduction technique of selecting the features that explain most of the variance can be applied. This can be achieved by ranking features by descending variance and selecting the top features that (cumulatively) explain a specified fraction (e.g., 99%) of the total variance. The remaining features can be assumed to be noise and dropped.

[0036] Pairwise correlations reduction may also be utilized. With this approach, the first step is to use PCA to estimate the dimensionality of the feature space. PCA is applied to transform from the original features to a set of principal components and to select the highest variance principal components that cumulatively explain a specified fraction of the variance. This provides an estimate of the effective dimensionality of the feature space. If the total explained variance threshold is large (e.g., 99% of explained variance), the remaining principal components to be dropped correspond to very low variance directions in the feature space. Each of these directions often results from pairs of very correlated features, which PCA transforms into a large variance component and a small variance component. Motivated by the latter, the next step comprises an iterative procedure of pairwise correlation reduction on the original features until the target dimensionality indicated by the PCA reduction is attained. Starting with the most correlated pair, one of the features is removed and this continues iteratively for all other possible pairs until a specified number of features remains.

[0037] Both PCA and pairwise correlation reduction can be extended to any measure of correlation, linear or non-linear. Spearman correlation measures some types of non-linear bivariate correlations by transforming each feature to a rank value. Thus, by first transforming the features to a space of ranks and applying PCA afterwards, some types of non-linear correlations can be removed. An assumption is made that the rank features are Gaussian distributed. Various non-linear methods can also be used to decorrelate features. Furthermore, approaches such as independent

components analysis (ICA) may be used to identify independent signals responsible for generating the data, rendering the independence assumption of the set of components to be monitored more valid.

[0038] In various embodiments, once the set of features to analyze is fixed (e.g., after processing by feature space reduction module 202), each feature is handled by feature handling component 204. In the descriptions below with respect to Figures 2A – 2C (and also Figures 3A – 3B), in order to illustrate specific examples, histograms are the data distribution representations described in detail as utilized by the various components. This is merely illustrative and not restrictive. Other distribution representations, such as parameterized functions, may also be used. In various embodiments, reference distribution build module 206 builds a histogram $H_{R,f}$ for each feature f = 1, ..., N to characterize a training data distribution, X_T , in a reference period of the training data (e.g., time period spanned by the training data). The obtained histograms represent distributions in the whole reference period. Various options are possible regarding the data used, such as using all of X_T or randomly sampling from it. As for building the histogram configuration of bins (or, in general, the representation of the distribution), various approaches may be utilized, as long as the configuration obtained for a specific feature f in this step is fixed across all feature f distributions used in build reference and data stream monitoring phases of a single framework / system. This configuration does not need to be the same for all features.

[0039] In some embodiments, only numerical features are included. In other embodiments, both numerical and non-numerical features are utilized. A set of equal height (or equal-frequency) reference histograms $H_{R,f}$ may be built for each feature f. This is illustrated in Figure 2B as generated reference histogram 212 from all reference data 210. Figure 2B schematically illustrates data patterns that are relevant on a per feature basis. In the example shown in Figure 2B, reference histogram 212 is an equal-height histogram (bin widths vary but bin heights do not). Equal-height histograms result in a larger number of bins covering the densest regions of the distribution. An example bin approach is as follows: utilize b + 3 bins where b bins are used to cover all existing values in the distribution of the reference data and add 3 additional bins, these additional bins being a leftmost bin covering negative infinity to the first bin of the b bins, a next to rightmost bin covering bin b of the b bins to infinity, and a rightmost bin (NaN bin). The semi-infinite bins cover regions not observed in the reference dataset. In contrast, the NaN bin is necessary for instances with invalid feature values (e.g., an empty value due to a feature collection problem). These three specific additional bins ensure that the histograms always have full support for any possible value.

[0040] In various embodiments, sampling and divergence distribution build module 208 performs sampling of reference data for each feature (e.g., sampling of all reference data 210). Sampling is of interest in part because monitoring of feature values during a data stream monitoring phase covers timescales that are usually considerably shorter than the reference period. For example, assume that all reference data 210 corresponds to six months where an ML model was trained. It may be desired to monitor the behavior of each feature in one-week periods after the model is deployed in a streaming environment. Hence, at each time step, it would be desirable to monitor the distribution of the values for each feature in the last week compared to its distribution in the original reference period of six months.

[0041] In various embodiments, a measure of divergence to compare two histograms (computation of divergence is described in further detail herein) is utilized. A primary assumption is that the reference period defines the expected data distribution. It is possible to estimate a histogram H_f of a feature f at various time steps within the reference period and compare each estimate with the full reference period $H_{R,f}$. An estimate of the distribution of expected fluctuations of H_f can be a new histogram of divergence values $H_{D,f}$. Samples of all reference data 210 (corresponding to $H_{R,f}$), e.g., at various time steps (corresponding to H_f), are illustrated in Figure 2B as samples 214 (Sample 1), 218 (Sample 2), and 222 (Sample M). Histograms for those samples are illustrated as sample histograms 216, 220, and 224. In Figure 2B, divergence histogram 226 (corresponding to $H_{D,f}$) shows a distribution of divergence values computed by comparing each sample histogram (including sample histograms 216, 220, and 224) with reference histogram 212. In many datasets, fluctuations of H_f are often associated with seasonal patterns. For example, variability of patterns during a reference period can be particularly prevalent for purchasing behaviors that have a seasonality associated with them (e.g., festivities with gift-exchanges, product launches/promotions, or other daily, weekly, and monthly routines).

[0042] The histogram of divergence values $H_{D,f}$ can, in principle, be obtained by computing the divergence of all possible H_f histograms (e.g., for all time steps) within the reference period. However, realistic datasets oftentimes have millions of events per day, requiring computing billions of histograms and their divergences, which becomes very computationally costly. To solve this problem, in various embodiments, random sampling is utilized to obtain a smaller representative number of time steps at which to compute the divergence values.

[0043] Data stream monitoring focuses on monitoring extreme events when the divergence values become large (in the higher percentiles of $H_{D,f}$). This means that a good estimate of the upper tails of distributions is needed. In various embodiments, the approach employed is to perform

a statistical test under a (multiple) null hypothesis that the divergence values observed for all the features during a data stream monitoring phase follow the same distributions as the corresponding $H_{D,f}$ during a build reference phase. The multiple hypothesis test requires setting a global significance level (or family-wise error rate) for its composite null hypothesis, which corresponds to the p-value of rejecting it due to random fluctuations. However, in multiple hypothesis testing, this typically results in a much stricter significance level being applied to at least some of the individual hypotheses because the probability that at least one of the tests fails by chance grows with the number of tests.

[0044] To obtain a conservative upper bound on the critical level for any feature, the Bonferroni correction, which is valid even if the hypotheses are dependent, can be utilized. Therefore, if any of N individual hypotheses (corresponding to N features) fails at a level $\alpha = \overline{\alpha}/N$, then the multiple hypothesis fails at a level $\overline{\alpha}$. In various embodiments, it is desired to ensure that divergence histograms have enough points to estimate the α-upper-tail appropriately based on this conservative bound. If the number of samples produced to represent $H_{D,f}$ is M, then the probability p_0 that none of those samples falls on the tail (assuming independent samples) is: $p_0 = (1-\alpha)^M$ (Equation 1). Furthermore, because N histograms are built, the probability that any histograms are missing samples in the tail of the distribution needs to be limited. The probability γ that one or more histograms are missing samples on the tail is related to the probability that none of them are missing samples on the tail: $\gamma = 1-(1-p_0)^N = 1-(1-(1-\alpha)^M)^N$ (Equation 2). This limits the probability of having one or more "tail"-incomplete histograms. Inverting this formula and replacing α by $\overline{\alpha}$ indicates that the minimum number of samples is: $M = \frac{\log[1-(1-\gamma)^{\frac{1}{N}}]}{\log(1-\frac{\overline{\alpha}}{N})} \simeq \frac{N\log(\frac{N}{\gamma})}{\overline{\alpha}}$ (Equation 3), where $\overline{\alpha} \ll 1$ and $\gamma \ll 1$ are used in the last step. With these numbers, the expected number μ of samples and its standard deviation σ in the tail region (the α-tail) can be estimated using the binomial distribution: $\mu = \alpha \times M$ and $\sigma = \sqrt{\mu(1-\alpha)}$. Thus, for example, with a family-wise error rate of $\overline{\alpha} = 0.01$, N = 100 features and $\gamma = 0.01$, the result is 9.2 ± 3.0 samples in the tail region of each histogram.

[0045] Sampling time steps when the sampled period is considerably smaller than the full reference period should capture well the fluctuations due to the time variability of the process and its statistical variability. However, in a scenario where the sampled/monitored timescale is comparable or even equal to the reference period, there will be few or only one possible sample. In that case, despite being impossible to estimate the fluctuations related to time variability, it is still possible to estimate the expected statistical variability in the histograms by making bootstrap

samples of the reference data to evaluate the expected distribution of divergences. In principle, this procedure can be combined with the sampling of time steps, also when the monitored timescale is short, to obtain an estimate of the distribution of divergences that better accounts for the expected statistical fluctuations in the reference data itself. When there is substantial seasonality, the main source of variability should be temporal, so sampling time steps is focused on herein.

[0046] In various embodiments, moving histograms are utilized. The computation of a moving histogram H_f to estimate the distribution of a feature in a given period may be performed in various ways. A sliding window (e.g., one week) may be used and the histogram may be computed using all the window events. However, during a data stream monitoring phase, this requires keeping track and storing all the events in the window, which can be computationally costly, especially for long windows and/or use cases with considerable event rates. A solution to this problem is to utilize a technique to estimate the distribution of features using either an Unbiased Exponential Moving Histogram (UEMH) or its time-based version—an Unbiased Time-Exponential Moving Histogram (UTEMH).

[0047] For a UEMH, an Unbiased Exponential Moving Average (UEMA) count is kept for each bin of a histogram. When a new instance arrives in the stream, UEMH uses a suppression factor to discount all histogram bin counters and then finds the bin associated with the new instance and increments its value by 1. The bin intervals are defined independently from the UEMH. A UEMA is a modification of an Exponential Moving Average (EMA) to correct for a bias towards the first value of the stream. An EMA makes use of a weighting method that discounts the previous event relative to the current one, always using the same suppression factor. Therefore, the EMA aggregations can be calculated as geometric progressions, which are recursive by definition. The EMA is calculated at each step considering only the current data instance value arriving at the stream and the aggregation value at the previous step. Therefore, EMAs do not require storing any events, only the current value of the aggregations. UEMAs have the same computational complexity advantages as the original EMAs. UTEMHs are similar to UEMHs but use a time-based discount factor instead, similar to Unbiased Time-Exponential Moving Averages (UTEMAs). A UTEMA is a version of EMAs in which instead of considering constant suppression steps between events, the time differences between instances are used to increase (longer time steps) or decrease (shorter time steps) the suppression factor. This approach is particularly suited for irregularly spaced time-series.

[0048] For UEMH and UTEMH, no events need to be stored, only the estimated histogram itself at each time step. The histogram is updated on each incoming event via a recursion formula.

Thus, the time and memory complexities of this method are O(nb) with n being the number of features and b the number of histogram bins. Since these two quantities are constant and small, the complexity of the update operation is constant both in time and memory with respect to the number of events contributing to the histogram. With this approach, all past events contribute to the histogram H_f, but with an exponentially decaying weight (older events are, in effect, forgotten). A half-life parameter controls the timescale corresponding to the histogram. For example, if the goal is to monitor a timescale of about one-week, a half-life corresponding to a few days would be appropriate to suppress events beyond a week. In various embodiments, matching format (same parameters) UEMHs or UTEMHs are used in build reference and data stream monitoring phases.

[0049] In various embodiments, sampling and divergence distribution build module 208 computes a representation of the distribution of divergence values $H_{D,f}$ for each feature by computing a divergence between the reference histogram $H_{R,f}$ and each sample H_f for each feature. Each divergence value contributes to a given bin of $H_{D,f}$. The particular binning strategy for $H_{D,f}$ is not critical but should have enough resolution (enough bins) in the tail spanning from $\bar{\alpha}$ to α to be able to estimate the most extreme percentiles reliably to determine the corresponding p-value of observing such a divergence. If the number of samples M is not very large, the simplest option is to save all divergence values for maximal resolution. Various types of divergence measures may be utilized to compare pairs of histograms and the divergence measure need not be the same for all features. Examples of divergence measures include Kolmogorov-Smirnov, Kuiper, and Anderson-Darling test statistics, various information theory divergences such as Kullback-Leibler (KL) divergence and Jensen-Shannon divergence (JSD), and Wasserstein distances.

[0050] In various embodiments, for each feature, build reference component 200 outputs a final state that can be used by a data stream monitoring component. In some embodiments, this final state is included in distribution data 106 of Figure 1. An example output of build reference component 200 is illustrated schematically in Figure 2C. In Figure 2C, for each feature f in the diagram (features 1, 2, ... N), outputs include a reference histogram $H_{R,f}$, a histogram of divergence values $H_{D,f}$, and an initial configuration for H_f. For feature 1 in Figure 2C, these are reference histogram 228, divergence histogram 230, and sample histogram 232, respectively. For feature 2 in Figure 2C, these are reference histogram 234, divergence histogram 236, and sample histogram 238, respectively. For feature N in Figure 2C, these are reference histogram 240, divergence histogram 242, and sample histogram 244, respectively. In some embodiments, each sample histogram is the last sample (e.g., last time step) sampled from the reference data. The sample histogram is typically chosen such that the data stream monitoring component starts in a

configuration that represents the reference period to avoid artificial alarms in the first iterations while processing the first events in the data stream monitoring phase.

[0051] A build reference component that outputs data to be provided to a corresponding data stream monitoring component is described in substantial detail herein. It is also possible to use the build reference component for other purposes. For example, the build reference component may be utilized to select features that have less divergence. The build reference component may also be utilized to summarize a representation of state. The build reference component provides a representation of some properties of a data distribution in a reference period (e.g., a pattern of fluctuations observed in a distribution). This can be viewed as a set of meta-features describing a dataset that can be used as an input to a meta-machine learning process. An example of a meta-machine learning task is to predict the accuracy of a model for a given period based on meta-features that represent the dataset.

[0052] The build reference component can be utilized to automatically compare various types of datasets (e.g., by computing divergence between datasets), which is useful in many applications. Many trained machine learning models are oftentimes applied to new datasets on which the machine learning models have not been trained. To ensure that the machine learning models can perform accurately on the new datasets, oftentimes, time-consuming manual labeling of new data is performed to evaluate the ML models that have been trained on a dataset, on new datasets. Using the build reference component to compare the datasets eliminates or reduces the need for manual labeling of data. Thus, the build reference component can be useful for any application for which determining that one large dataset is similar to another is beneficial.

[0053] Figure 2B is a diagram illustrating example data patterns generated by a build reference component. Figure 2B is described above with respect to Figure 2A.

[0054] Figure 2C is a diagram illustrating example data outputs of a build reference component. Figure 2C is described above with respect to Figure 2A.

[0055] Figure 3A is a block diagram illustrating an embodiment of a data stream monitoring component. In some embodiments, data stream monitoring component 300 is data stream monitoring component 108 of framework 100 of Figure 1. In some embodiments, data stream monitoring component 300 (including its sub-components) is comprised of computer program instructions that are executed on a general-purpose processor, e.g., a CPU, of a programmed computer system. Figure 7 illustrates an example of a programmed computer system.

It is also possible for the logic of data stream monitoring component 300 to be executed on other hardware, e.g., executed using an ASIC or an FPGA.

[0056] In the example shown, data stream monitoring component 300 includes distributions update module 302, data analysis module 304, and reporting module 306. The components and organization shown in Figure 3A is merely illustrative and not restrictive. Other configurations are also possible. In the example shown, portions of the communication path between the components are shown. Other communication paths may exist, and the example of Figure 3A has been simplified to illustrate the example clearly. Although single instances of components have been shown to simplify the diagram, additional instances of any of the components shown in Figure 3A may exist. Components not shown in Figure 3A may also exist.

[0057] Data stream monitoring component 300 continuously or continually monitors features. The monitoring is done on an ordered set of data instances that is processed in a streaming fashion (e.g., a streaming system collecting data in real time), producing different outputs as a result. These outputs can include alarms (e.g., to be handled by system administrators), automated actions (e.g., model retraining, rule threshold adjustment, explanation computation), and/or signal visualizations. In many real-time applications, the monitoring runs over an unbounded stream of data. Stated alternatively, the streaming computation is intended to run indefinitely. Data stream monitoring component 300 receives streaming data (e.g., streaming data 110 of Figure 1). In some embodiments, data stream monitoring component 300 processes each incoming event, wherein an event is a set of data fields, one by one. It is also possible for data stream monitoring component 300 to process events in batches. In various embodiments, each event that is processed comprises a subset of fields that is equivalent to a subset of fields needed by a build reference component to build reference data representations of the features to be monitored. In various embodiments, the output of a previous build reference computation provides the input to data stream monitoring component 300. In various embodiments, the input to data stream monitoring component 300 includes, for each feature to be monitored, a reference histogram $H_{R,f}$ and a divergence histogram $H_{D,f}$. In some embodiments, another input is a configuration parameter specifying frequency (time or event based) at which hypothesis testing (to raise an alarm) occurs.

[0058] In various embodiments, for each feature, distributions update module 302 updates a target histogram using a same update method that is used to build sample histograms by a build reference component. In some embodiments, the target histogram is initialized to be a received sample histogram generated and outputted by the build reference component. For EMA based histograms, when the latest event arrives, all bin counts in the target histogram are suppressed by a

common factor. This factor can be a constant. It can also be an exponential of the time difference since the last event. This suppression mechanism can be viewed as a soft expiration mechanism for older events, whose contribution is progressively reduced towards zero (instead of a hard expiration, e.g., in a sliding window). The bin corresponding to the feature value for the incoming event is identified and its count is increased by one. This is illustrated schematically in Figure 3B as bin increment 310 of target histogram 312 for feature 1, bin increment 314 of target histogram 316 for feature 2, and bin increment 318 of target histogram 320 for feature N. The histogram update operation can be a computationally demanding component, particularly when it is performed for each event. In various embodiments, UEMH and UTEMH techniques (as described above) are utilized to reduce the time and space complexity of histogram updates to a constant factor that depends only on the number of features under monitoring and the number of histogram bins used. In addition, the computational demands can be reduced by performing batch processing of events.

[0059] In various embodiments, for each feature, data analysis module 304 performs data analysis on an updated target histogram using a corresponding reference histogram and divergence histogram. This analysis does not need to occur for every incoming event. Rather, it may be performed for every batch of events. In various embodiments, a divergence between a current target histogram Hr and a corresponding reference histogram $H_{R,f}$ is computed. This value is then located within a corresponding divergence histogram $H_{D,f}$. This is illustrated in Figure 3B for feature 1, feature 2, and feature N. For feature 1, the divergence between target histogram 312 and reference histogram 322 is computed, and the computed divergence value 324 is located on divergence histogram 326. For feature 2, the divergence between target histogram 316 and reference histogram 328 is computed, and the computed divergence value 330 is located on divergence histogram 332. For feature N, the divergence between target histogram 320 and reference histogram 334 is computed, and the computed divergence value 336 is located on divergence histogram 338. As shown in Figure 3B, the location where the computed divergence value lies on the corresponding divergence histogram varies. For feature 1, computed divergence value 324 lies in the middle of divergence histogram 326, indicating computed divergence value 324 is not an outlier in divergence histogram 326. In contrast, for feature 2, computed divergence value 330 lies at the right hand tail end of divergence histogram 332, indicating computed divergence value 330 is an outlier in divergence histogram 332. In various embodiments, for each feature, data analysis module 304 also computes a statistical value for each computed divergence value to characterize whether the computed divergence value is significantly larger than typically observed divergence values (e.g., larger than divergence values determined during a build reference phase). In some embodiments, the statistical value computed is a p-value. In some embodiments, the p-value is

computed by estimating a corresponding cumulative distribution function (CDF) value and subtracting it from 1. Each of these p-values is schematically represented in Figure 3B as the area under the histogram to the right of each computed divergence value.

[0060] In various embodiments, for each feature, the data analysis module applies a multivariate hypothesis test after statistical values (e.g., p-values) are computed. In the example shown in Figure 3B, p-value 1, p-value 2, ..., and p-value N would be the inputs to the multivariate hypothesis test. Stated alternatively, a statistical test that aggregates and/or transforms the individual p-values is applied. The aggregation may include, for example, re-ordering and scaling of the p-values to correct for the fact that the probability of observing a large statistical fluctuation in one of the p-values grows with the number of p-values tested. In some embodiments, a Holm-Bonferroni correction is applied, but other tests are possible (e.g., Hochberg, Hommel, and Rom multiple testing procedures). In some embodiments, the p-values are first ordered by ascending value $p_1, ..., p_N$. Each p-value p_i is associated with a feature f_i. Then each p-value p_i can be scaled to produce a signal s_i defined as $s_i = p_i \times (N + 1 - i), \ with \ i = 1, \dots, N$ (Equation 4). The null hypothesis is rejected if, for any (or several) of the features f_i, $s_i < \bar{\alpha}$, and an alarm is then raised. For this test, $\bar{\alpha}$ serves as the threshold.

[0061] In various embodiments, in response to an alarm being raised, reporting module 306 outputs an alarm explanation to transmit to a user. In some embodiments, the alarm explanation is included in monitoring results data 112 of Figure 1. The alarm explanation indicates why an alarm has been raised and may help the user to quickly identify a root cause of a data drift. In various embodiments, the alarm explanation is conveyed in an alarm report and is critical for user investigation. The alarm report is also of interest to a user that is interested in regularly examining the state of data stream monitoring, regardless of any recent alarm.

[0062] An element of the multivariate test that can be used to explain the alarm is the set of signals, one per feature, and their respective ranking. This ranking automatically provides a measure of each feature's importance to explain the fluctuation (explain which features deviate the most from their reference distribution). In the particular case of the Holm-Bonferroni test, this ranking already considers that several features are tested simultaneously.

[0063] Although the signals produced directly from the monitored features may, in some cases, directly correspond to interpretable features, in the case where an initial transformation has been applied, e.g., to decorrelate features or to remove spurious ones, the same may not hold. For example, suppose a feature such as transaction amount is responsible for the alarm. In that case,

this means that its distribution is unusual, warranting investigation. However, consider an alarm in a derived feature such as 0.3 × amount + 0.7 × "number of transactions per credit card in the last hour", or in a component of a non-linear transformation of the original features using a neural network. In that case, it is not clear what happened or what the next steps in an investigation should be. In this case, transforming the signals to the original interpretable space of features via a linear or non-linear transformation is desirable. For example, if the de-correlation transformation was performed via PCA, the squared entries of the rows of an inverse projection matrix r^2_{xf} (used to reconstruct the feature set x from the monitored features f) may be used to assign a weight to each of the features of the original feature set x through a weighted sum $W_x = \frac{\sum_i r^2_{xf_i} s_i}{\sum_i r^2_{xf_i}}$ (Equation 5).

Other strategies of this type could be applied in the case of non-linear transformations. For example, a matrix of correlations (linear or non-linear correlations) could be computed between the interpretable features and the monitored features and then a similar procedure as in Equation 5 can be applied. After weights for interpretable features are obtained, interpretable features may be sorted by ascending weight value. This sorting provides a measure of feature importance for the alarm.

[0064] In various embodiments, reporting module 306 of data stream monitoring component 300 outputs a report of the alarm whenever one is produced and the report is appended to a list of historical alarms. A notification may be sent to one or more users, including complete information on when the alarm occurred, which features alarmed, the state of the computed signals, and the weights of the interpretable features. In various embodiments, even in the absence of an alarm, the information on the signals computed and the features' ranking and weights is saved by data stream monitoring component 300 for continuous system monitoring. In various embodiments, after the alarm is reported, data stream monitoring component 300 waits for a next event or group of events to process.

[0065] Figure 3B is a diagram illustrating examples of data analysis associated with a data stream monitoring component. Figure 3B is described above with respect to Figure 3A.

[0066] Figure 4 is a flow diagram illustrating an embodiment of a process for generating divergence distributions for automated data analysis. In some embodiments, the process of Figure 4 is performed by build reference component 104 of framework 100 of Figure 1 and/or build reference component 200 of Figure 2A.

[0067] At 402, a set of data elements is received. In some embodiments, the set of data elements is included in reference data 102 of Figure 1. In various embodiments, the data elements correspond to logically separate events, such as events, from a data stream, that occur at distinct moments in time. In various embodiments, each data element is comprised of sub-elements or items of data that correspond to features that are utilized to characterize that data element. An example of a data element is transaction information associated with a single purchase by a particular user. The transaction information can be comprised of various data items, such as transaction amount, transaction time and date, transaction history of the user, etc. In various embodiments, the data elements are collected from many users, oftentimes geographically spread across the world.

[0068] At 404, for each feature of a set of features, a corresponding reference distribution is determined for the set of data elements. With respect to the example of fraud detection, examples of features include amount of a purchase, time and date of the purchase, location of the purchase, purchaser identity, number of purchases made by the purchaser within a specified recent period of time, total amount of purchases made by the purchaser within the specified period of time, etc. As the above examples illustrate, features may be numerical features or non-numerical. Some features may be derived from other features. Values associated with each numerical feature vary and have a distribution. In some embodiments, this distribution, the reference distribution, is represented by a histogram (e.g., reference histogram 212 of Figure 2B), though other distribution representations (e.g., parameterized functions) are also possible.

[0069] At 406, for each feature of the set of features, one or more corresponding subset distributions are determined for one or more subsets sampled from the set of data elements. In some embodiments, the subsets sampled correspond to specified time steps within a reference time period associated with the set of data elements. For example, the reference time period for the set of data elements may be 3 months, 6 months, etc. Stated alternatively, all of the reference data may come from a time period of 3 months, 6 months, etc. The subsets may correspond to data from the set of data elements for each one-week sub-period within the 3-month, 6-month, etc. reference time period. Data values from each subset also have a distribution (similar to the reference distribution). In some embodiments, subset distributions are represented as histograms (e.g., sample histograms 216, 220, and 224 of Figure 2B).

[0070] At 408, for each feature of the set of features, the corresponding reference distribution is compared with each of the one or more corresponding subset distributions to determine a corresponding distribution of divergences. In various embodiments, a divergence measure between the reference distribution and each subset distribution is computed. For example,

a Jensen-Shannon divergence may be computed between a histogram representing the reference distribution and another histogram representing a subset distribution. This divergence quantitatively characterizes how different the two distributions are. Computing divergence measures between the reference distribution and each subset distribution results in multiple divergence values that themselves vary (according to a distribution). In some embodiments, this distribution of the divergence values is represented as a histogram (e.g., divergence histogram 226 of Figure 2B).

[0071] At 410, at least the determined distributions of divergences for the set of features are provided for use in automated data analysis. In some embodiments, the reference distributions are also provided. In some embodiments, for each feature, at least one subset distribution is also provided (e.g., corresponding to a most recent time step sample). In various embodiments, the various distributions are provided as histograms. In some embodiments, the distributions are included in distribution data 106 of Figure 1 and provided to data stream monitoring component 108 of Figure 1.

[0072] Figure 5 is a flow diagram illustrating an embodiment of a process for performing automated feature monitoring of a data stream. In some embodiments, the process of Figure 5 is performed by data stream monitoring component 108 of framework 100 of Figure 1 and/or data stream monitoring component 300 of Figure 3A.

[0073] At 502, one or more events in a data stream are received. In some embodiments, the data stream is comprised of streaming data 110 of Figure 1. The one or more events may be processed one by one (event by event) or in groups of events (batch processing). With respect to the example of fraud detection, each event may correspond to a purchase transaction by a user. In various embodiments, each event comprises various data items corresponding to features associated with the event. With respect to the example of fraud detection, examples of features include amount of a purchase, time and date of the purchase, location of the purchase, purchaser identity, number of purchases made by the purchaser within a specified recent period of time, total amount of purchases made by the purchaser within the specified period of time, etc. As the above examples illustrate, features may be numerical or non-numerical. Some features may be derived from other features.

[0074] At 504, for each feature of a set of features, the one or more events are used to update a corresponding distribution of data from the data stream. For event-by-event processing, updating occurs after each event is received. For batch processing, updating occurs after multiple events have been received. In some embodiments, the distribution of data is represented as a histogram (e.g., target histograms 312, 316, and 320 of Figure 3B corresponding to different

features). Updating of the histogram can be performed by incrementing bins of the histogram. For example, for a histogram of amount purchase values, histogram bins may be arranged in a specified manner to cover purchases from zero dollars to infinity dollars. If a first event is associated with a purchase amount of $100, a histogram bin including $100 would be incremented. Then, if a second event is associated with a purchase amount of $1000, most likely another histogram bin would be incremented (unless a bin covers a range that includes both $100 and $1000). In various embodiments, as more and more events of the data stream are processed, older events are expired. As used herein, expiration refers to deemphasizing or eliminating contributions. For example, events older than one week, two weeks, etc. may be completely eliminated from the histogram (e.g., using a sliding window technique) so that their corresponding data values are not counted in any of the bins of the histogram. Thus, in some scenarios, only a subset of the data stream affects the updated distribution of data.

[0075] Older events may also be deemphasized (e.g., by applying a scaling factor less than one) so that their corresponding data values are not given as much weight in the counts of the histogram. In some embodiments, a UEMH or similar type of histogram is utilized. With the UEMH, a UEMA count is kept for each bin of the histogram. The UEMA, similar to an EMA, makes use of a weighting method that discounts the previous event relative to the current one, using the same suppression factor. Therefore, the UEMA (and similar recursive measures) can be calculated as geometric progressions and can be calculated at each step considering only the current data instance value arriving at the data stream and the aggregation value at the previous step. Therefore, UEMA (and similar recursive measures) do not require storing any events, only the current value of the aggregations and consequently have computational complexity advantages (e.g., lower memory usage).

[0076] At 506, for each feature of the set of features, the corresponding updated distribution and a corresponding reference distribution are used to determine a corresponding divergence value. In some embodiments, the updated distribution and the reference distribution are represented as histograms. In some embodiments, the reference distribution is received from build reference component 104 of framework 100 of Figure 1 and/or build reference component 200 of Figure 2A. Examples of reference distribution histograms include reference histograms 322, 328, and 334 of Figure 3B. In various embodiments, the reference distribution is derived from a larger dataset than what is used to generate the updated distribution. For example, the reference distribution may be derived from 6 months of training data, whereas the updated distribution depends only on a week, two weeks, etc. of data. The divergence value is a measure of how similar the updated distribution

is to the reference distribution. In some embodiments, the divergence value is a Jensen-Shannon divergence. Various other types of divergence measures, e.g., Kolmogorov-Smirnov, Kuiper, and Anderson-Darling test statistics, Kullback-Leibler divergence, and Wasserstein distances, may also be used.

[0077] At 508, for each feature of the set of features, the corresponding determined divergence value and a corresponding distribution of divergences are used to determine a corresponding statistical value. In some embodiments, the statistical value is a p-value. In some embodiments, the distribution of divergences is represented as a histogram (e.g., divergence histograms 326, 332, and 338 of Figure 3B). In some embodiments, the distribution of divergences is received from build reference component 104 of framework 100 of Figure 1 and/or build reference component 200 of Figure 2A. In some embodiments, determining the statistical value includes determining a p-value by estimating a corresponding cumulative distribution function value and subtracting it from 1. The p-value is a measure of how much of an outlier the determined divergence value is in the distribution of divergences.

[0078] At 510, using the statistical values each corresponding to a different feature of the set of features, a statistical analysis is performed to determine a result associated with a likelihood of data drift detection. In some embodiments, the statistical analysis includes performing a multivariate hypothesis test that aggregates and/or transforms the statistical values (e.g., the p-values). The aggregation may include, for example, re-ordering and scaling of p-values to correct for the fact that the probability of observing a large statistical fluctuation in one of the p-values grows with the number of p-values tested. In some embodiments, a Holm-Bonferroni correction is applied, but other tests are possible (e.g., Hochberg, Hommel, and Rom multiple testing procedures). In various embodiments, the statistical analysis outputs a groupwise error rate value, which if in excess of a specified threshold (e.g., 0.05, 0.01, etc.) triggers an alarm because being in excess of the threshold indicates a threshold likelihood of data drift (e.g., an abnormal divergence between the updated distribution and the reference distribution) that warrants an alarm / report being made.

[0079] Figure 6 is a flow diagram illustrating an embodiment of a process for performing statistical analysis to determine whether to generate an alarm associated with feature monitoring. In some embodiments, the process of Figure 6 is performed by data stream monitoring component 108 of framework 100 of Figure 1 and/or data stream monitoring component 300 of Figure 3A. In some embodiments, at least a portion of the process of Figure 6 is performed in 510 of Figure 5.

[0080] At 602, statistical values are received. In some embodiments, each statistical value is a p-value associated with a divergence of a distribution with respect to a reference distribution. In various embodiments, each statistical value is associated with a specific feature in a set of features derived from data of a data stream.

[0081] At 604, a multivariate statistical test is performed using the statistical values. In various embodiments, the multivariate statistical test includes a correction, such as a Holm-Bonferroni correction, to account for how many statistical values are used in the multivariate statistical test. In various embodiments, the multivariate statistical test outputs a single numerical value.

[0082] At 606, it is determined whether an alarm condition has been met. In various embodiments, the alarm condition is met if the numerical value output of the multivariate statistical test is below a specified threshold. In various embodiments, this is indicative of a higher than specified threshold likelihood that data from the data stream diverge (e.g., are not sufficiently similar) to reference data. This is a cause for alarm in many applications (e.g., when an ML model is being utilized on data from the data stream but was trained on the reference data and the ML model may not operate adequately when the data from the data stream and the reference data are not sufficiently similar).

[0083] If at 606 it is determined that the alarm condition has not been met, no further action is taken. If at 606 it is determined that the alarm condition has been met, at 608, an alarm is generated. In some embodiments, information associated with the generated alarm is included in monitoring results data 112 of Figure 1. In various embodiments, the generated alarm includes an alarm explanation that indicates why an alarm was raised and may help a user to quickly identify a cause of data drift.

[0084] Figure 7 is a functional diagram illustrating a programmed computer system. In some embodiments, processes associated with build reference component 104 of framework 100 of Figure 1, build reference component 200 of Figure 2A, data stream monitoring component 108 of framework 100 of Figure 1, and/or data stream monitoring component 300 of Figure 3A are executed by computer system 700.

[0085] In the example shown, computer system 700 includes various subsystems as described below. Computer system 700 includes at least one microprocessor subsystem (also referred to as a processor or a central processing unit (CPU)) 702. For example, processor 702 can

be implemented by a single-chip processor or by multiple processors. In some embodiments, processor 702 is a general-purpose digital processor that controls the operation of computer system 700. Using instructions retrieved from memory 710, processor 702 controls the reception and manipulation of input data, and the output and display of data on output devices (e.g., display 718).

[0086] Processor 702 is coupled bi-directionally with memory 710, which can include a first primary storage, typically a random-access memory (RAM), and a second primary storage area, typically a read-only memory (ROM). As is well known in the art, primary storage can be used as a general storage area and as scratch-pad memory, and can also be used to store input data and processed data. Primary storage can also store programming instructions and data, in the form of data objects and text objects, in addition to other data and instructions for processes operating on processor 702. Also, as is well known in the art, primary storage typically includes basic operating instructions, program code, data, and objects used by the processor 702 to perform its functions (e.g., programmed instructions). For example, memory 710 can include any suitable computer-readable storage media, described below, depending on whether, for example, data access needs to be bi-directional or uni-directional. For example, processor 702 can also directly and very rapidly retrieve and store frequently needed data in a cache memory (not shown).

[0087] Persistent memory 712 (e.g., a removable mass storage device) provides additional data storage capacity for computer system 700, and is coupled either bi-directionally (read/write) or uni-directionally (read only) to processor 702. For example, persistent memory 712 can also include computer-readable media such as magnetic tape, flash memory, PC-CARDS, portable mass storage devices, holographic storage devices, and other storage devices. A fixed mass storage 720 can also, for example, provide additional data storage capacity. The most common example of fixed mass storage 720 is a hard disk drive. Persistent memory 712 and fixed mass storage 720 generally store additional programming instructions, data, and the like that typically are not in active use by the processor 702. It will be appreciated that the information retained within persistent memory 712 and fixed mass storages 720 can be incorporated, if needed, in standard fashion as part of memory 710 (e.g., RAM) as virtual memory.

[0088] In addition to providing processor 702 access to storage subsystems, bus 714 can also be used to provide access to other subsystems and devices. As shown, these can include a display monitor 718, a network interface 716, a keyboard 704, and a pointing device 706, as well as an auxiliary input/output device interface, a sound card, speakers, and other subsystems as needed. For example, pointing device 706 can be a mouse, stylus, track ball, or tablet, and is useful for interacting with a graphical user interface.

[0089] Network interface 716 allows processor 702 to be coupled to another computer, computer network, or telecommunications network using a network connection as shown. For example, through network interface 716, processor 702 can receive information (e.g., data objects or program instructions) from another network or output information to another network in the course of performing method/process steps. Information, often represented as a sequence of instructions to be executed on a processor, can be received from and outputted to another network. An interface card or similar device and appropriate software implemented by (e.g., executed/performed on) processor 702 can be used to connect computer system 700 to an external network and transfer data according to standard protocols. Processes can be executed on processor 702, or can be performed across a network such as the Internet, intranet networks, or local area networks, in conjunction with a remote processor that shares a portion of the processing. Additional mass storage devices (not shown) can also be connected to processor 702 through network interface 716.

[0090] An auxiliary I/O device interface (not shown) can be used in conjunction with computer system 700. The auxiliary I/O device interface can include general and customized interfaces that allow processor 702 to send and, more typically, receive data from other devices such as microphones, touch-sensitive displays, transducer card readers, tape readers, voice or handwriting recognizers, biometrics readers, cameras, portable mass storage devices, and other computers.

[0091] In addition, various embodiments disclosed herein further relate to computer storage products with a computer readable medium that includes program code for performing various computer-implemented operations. The computer-readable medium is any data storage device that can store data which can thereafter be read by a computer system. Examples of computer-readable media include, but are not limited to, all the media mentioned above: magnetic media such as hard disks, floppy disks, and magnetic tape; optical media such as CD-ROM disks; magneto-optical media such as optical disks; and specially configured hardware devices such as application-specific integrated circuits (ASICs), programmable logic devices (PLDs), and ROM and RAM devices. Examples of program code include both machine code, as produced, for example, by a compiler, or files containing higher level code (e.g., script) that can be executed using an interpreter.

[0092] The computer system shown in Figure 7 is but an example of a computer system suitable for use with the various embodiments disclosed herein. Other computer systems suitable for such use can include additional or fewer subsystems. In addition, bus 714 is illustrative of any

interconnection scheme serving to link the subsystems. Other computer architectures having different configurations of subsystems can also be utilized.

[0093] Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

CLAIMS

1. A method, comprising:

receiving one or more events of a data stream;

for each feature of a set of features, using the one or more events to update a corresponding distribution of data from the data stream;

for each feature of the set of features, using the corresponding updated distribution and a corresponding reference distribution to determine a corresponding divergence value;

for each feature of the set of features, using the corresponding determined divergence value and a corresponding distribution of divergences to determine a corresponding statistical value; and

using the statistical values each corresponding to a different feature of the set of features, performing a statistical analysis to determine a result associated with a likelihood of data drift detection.

2. The method according to any of the previous claims, wherein at least a portion of the one or more events have occurred at distinct points in time.

3. The method according to any of the previous claims, wherein the one or more events correspond to information associated with transactions being analyzed to detect fraud.

4. The method according to any of the previous claims, wherein one or more features of the set of features are associated with a numerical measurement of data.

5. The method according to any of the previous claims, wherein one or more features of the set of features are utilized by a machine learning model for predictive tasks.

6. The method according to any of the previous claims, wherein using the one or more events to update the corresponding distribution of data from the data stream includes assigning each of the one or more events to a category among a plurality of categories associated with the corresponding distribution of data and correspondingly incrementing counts of events in categories of the plurality of categories.

7. The method according to any of the previous claims, wherein the corresponding distribution of data from the data stream is represented as a histogram.

8. The method of claim 7, wherein the histogram is generated including by applying an exponential moving average suppression of older events.

9. The method according to any of the previous claims, wherein the corresponding statistical value is a p-value.

10. The method according to any of the previous claims, further comprising receiving, for each feature of the set of features, the corresponding reference distribution and the corresponding distribution of divergences.

11. The method according to any of the previous claims, wherein performing the statistical analysis includes performing a multivariate hypothesis test.

12. The method of claim 11, wherein performing the multivariate hypothesis test includes scaling the statistical values.

13. The method according to any of the previous claims, wherein the statistical analysis is performed each time a batch of events is received.

14. The method according to any of the previous claims, further comprising analyzing the result to determine whether a specified condition has been satisfied.

15. The method of claim 14, further comprising, in response to a determination that the specified condition has been satisfied, providing an alarm.

16. The method of claim 15, wherein the alarm causes a generation of an alarm report that includes a ranking of features of the set of features according to how much each feature of the set of features contributed to the alarm.

17. The method of claim 15, wherein the alarm causes retraining of a machine learning model.

18. The method according to any of claims 14-17, wherein the specified condition is associated with one or more comparisons to a threshold value.

19. A system, comprising:

one or more processors configured to:

receive one or more events in a data stream;

for each feature of a set of features, use the one or more events to update a corresponding distribution of data from the data stream;

for each feature of the set of features, use the corresponding updated distribution and a corresponding reference distribution to determine a corresponding divergence value;

for each feature of the set of features, use the corresponding determined divergence value and a corresponding distribution of divergences to determine a corresponding statistical value; and

using the statistical values each corresponding to a different feature of the set of features, perform a statistical analysis to determine a result associated with a likelihood of data drift detection; and

a memory coupled to at least one of the one or more processors and configured to provide at least one of the one or more processors with instructions.

20. A computer program product embodied in a non-transitory computer readable medium and comprising computer instructions for:

receiving one or more events in a data stream;

for each feature of a set of features, using the one or more events to update a corresponding distribution of data from the data stream;

for each feature of the set of features, using the corresponding updated distribution and a corresponding reference distribution to determine a corresponding divergence value;

for each feature of the set of features, using the corresponding determined divergence value and a corresponding distribution of divergences to determine a corresponding statistical value; and

using the statistical values each corresponding to a different feature of the set of features, performing a statistical analysis to determine a result associated with a likelihood of data drift detection.

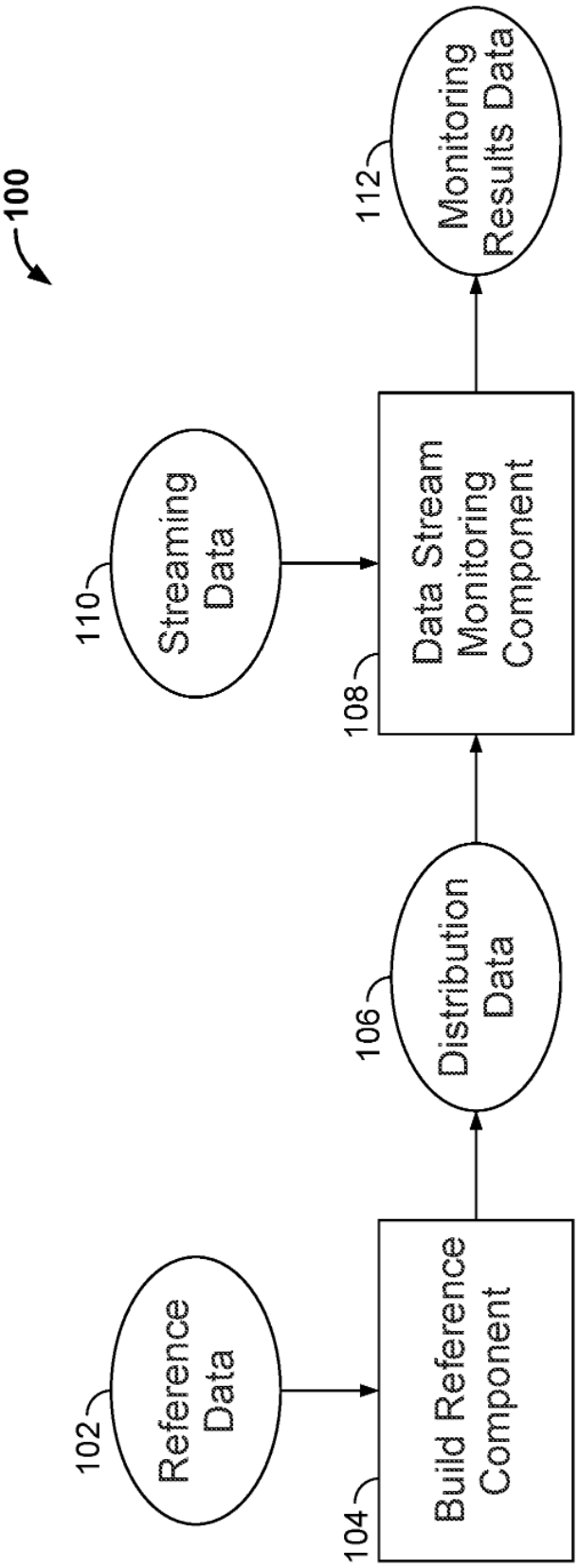
100
102
Reference Data
104
Build Reference Component
106
Distribution Data
108
Data Stream Monitoring Component
110
Streaming Data
112
Monitoring Results Data

FIG. 1

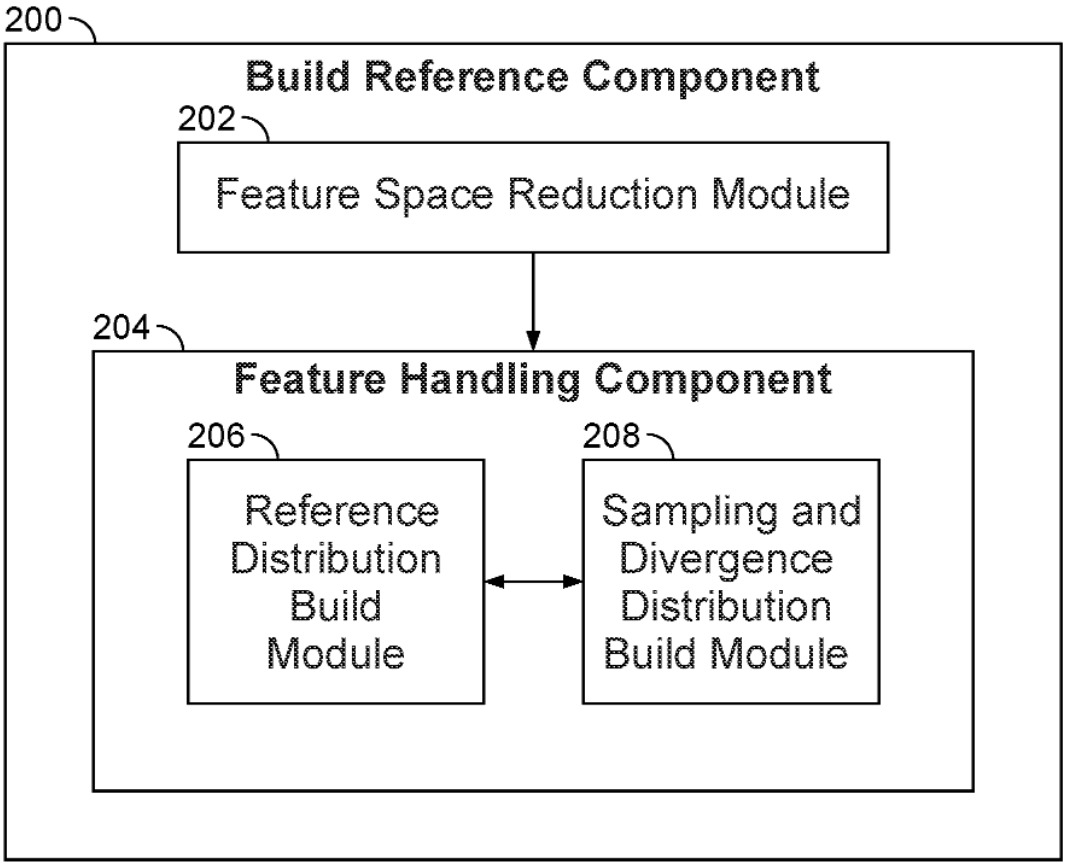
200
Build Reference Component
202
Feature Space Reduction Module
204
Feature Handling Component
206
Reference Distribution Build Module
208
Sampling and Divergence Distribution Build Module

FIG. 2A

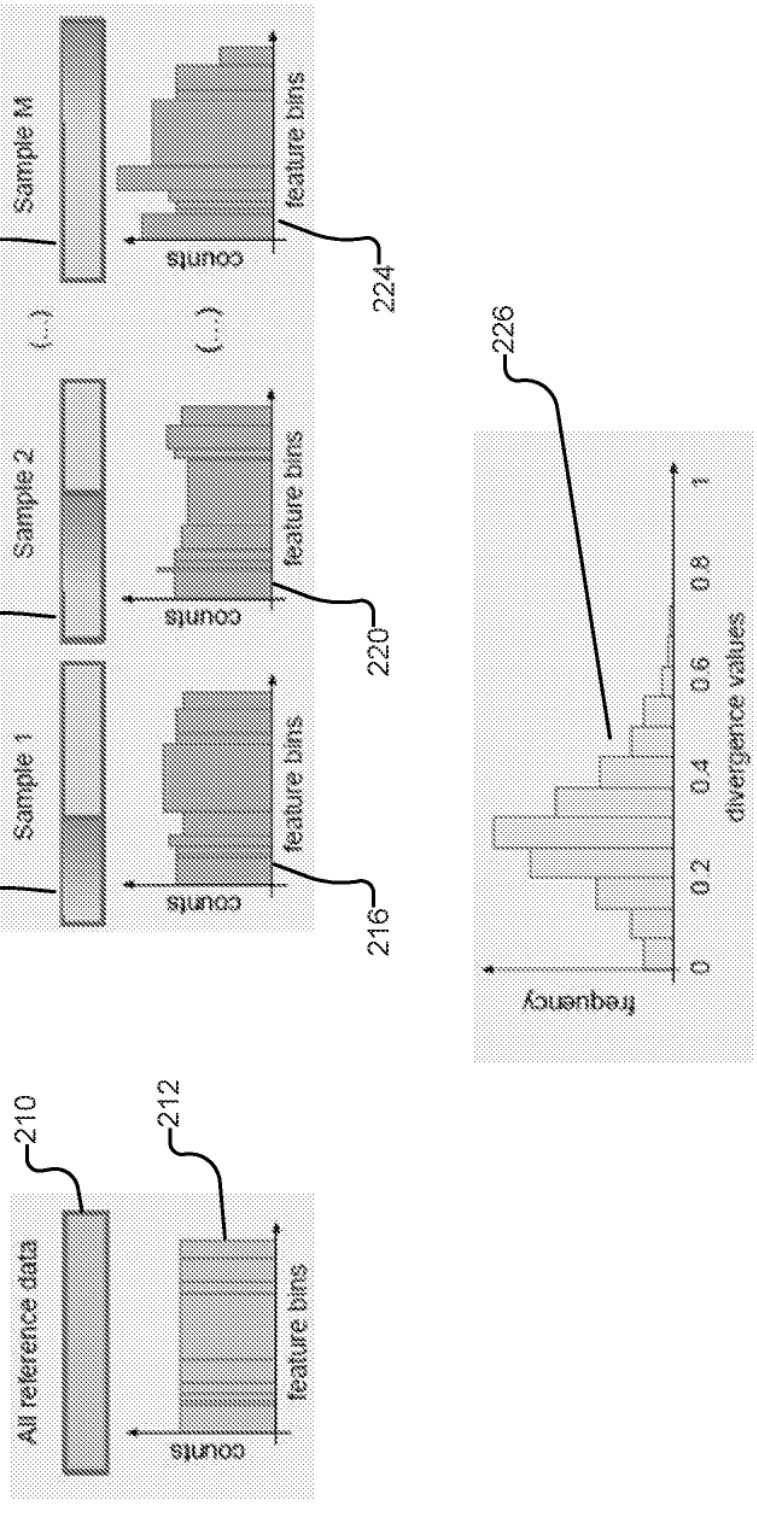
All reference data
counts
feature bins
210
212
Sample 1
Sample 2
Sample M
(...)
(...)
214
216
218
220
222
224
frequency
divergence values
0
0.2
0.4
0.6
0.8
1
226

FIG. 2B

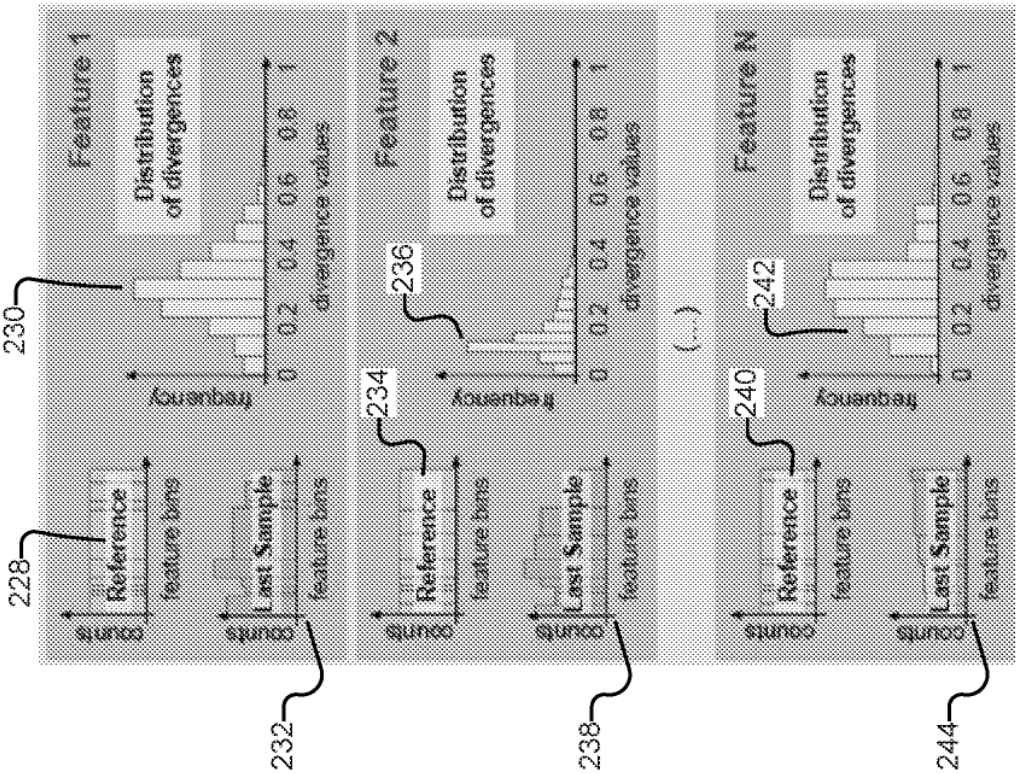
Feature 1
Distribution of divergences
Reference
feature bins
counts
Last Sample
feature bins
counts
frequency
0 0.2 0.4 0.6 0.8 1
divergence values
228
230
232
Feature 2
Distribution of divergences
234
236
238
(...)
Feature N
Distribution of divergences
240
242
244

FIG. 2C

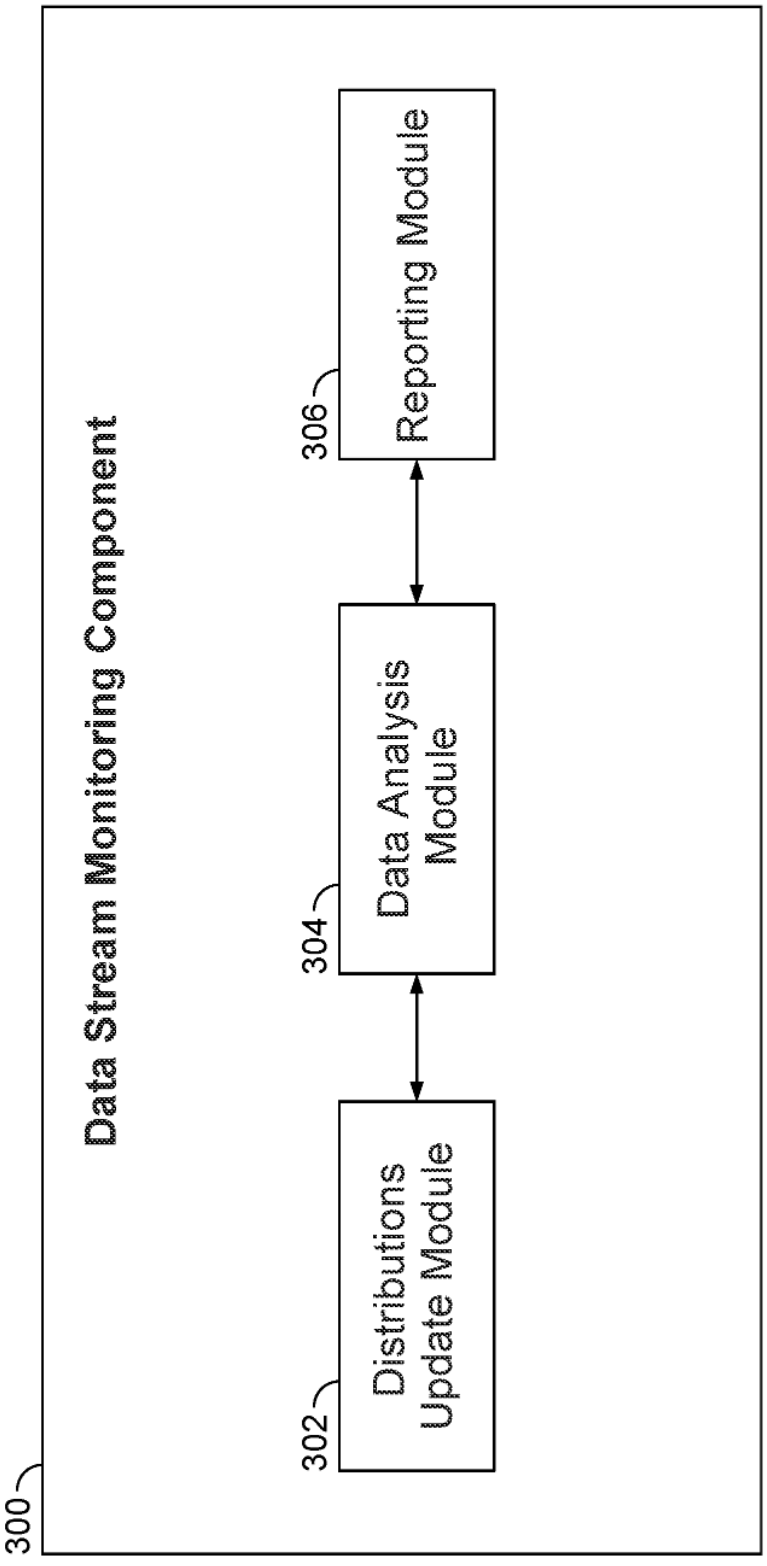
Data Stream Monitoring Component
300
Distributions Update Module
302
Data Analysis Module
304
Reporting Module
306

FIG. 3A

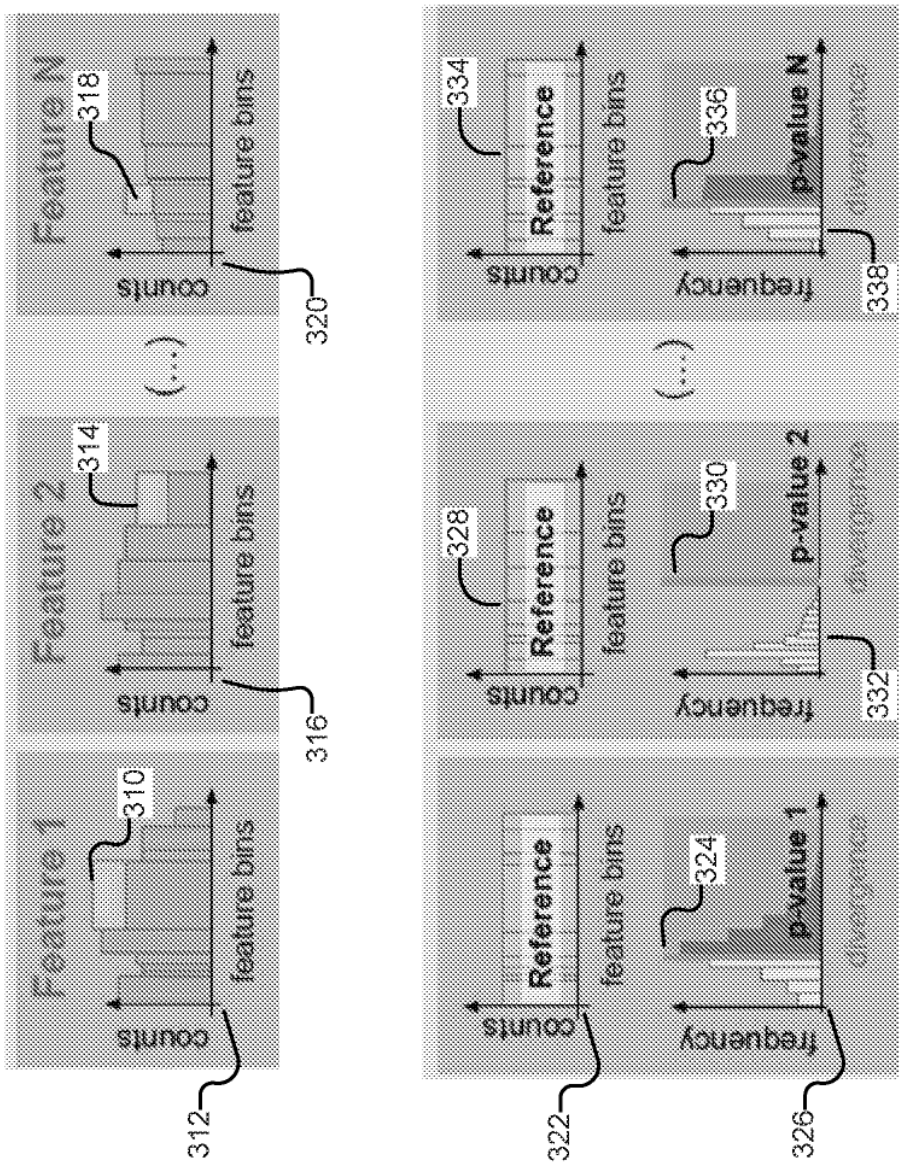
Feature 1
Feature 2
Feature N
counts
feature bins
Reference
frequency
p-value 1
p-value 2
p-value N
(...)
310
312
314
316
318
320
322
324
326
328
330
332
334
336
338

FIG. 3B

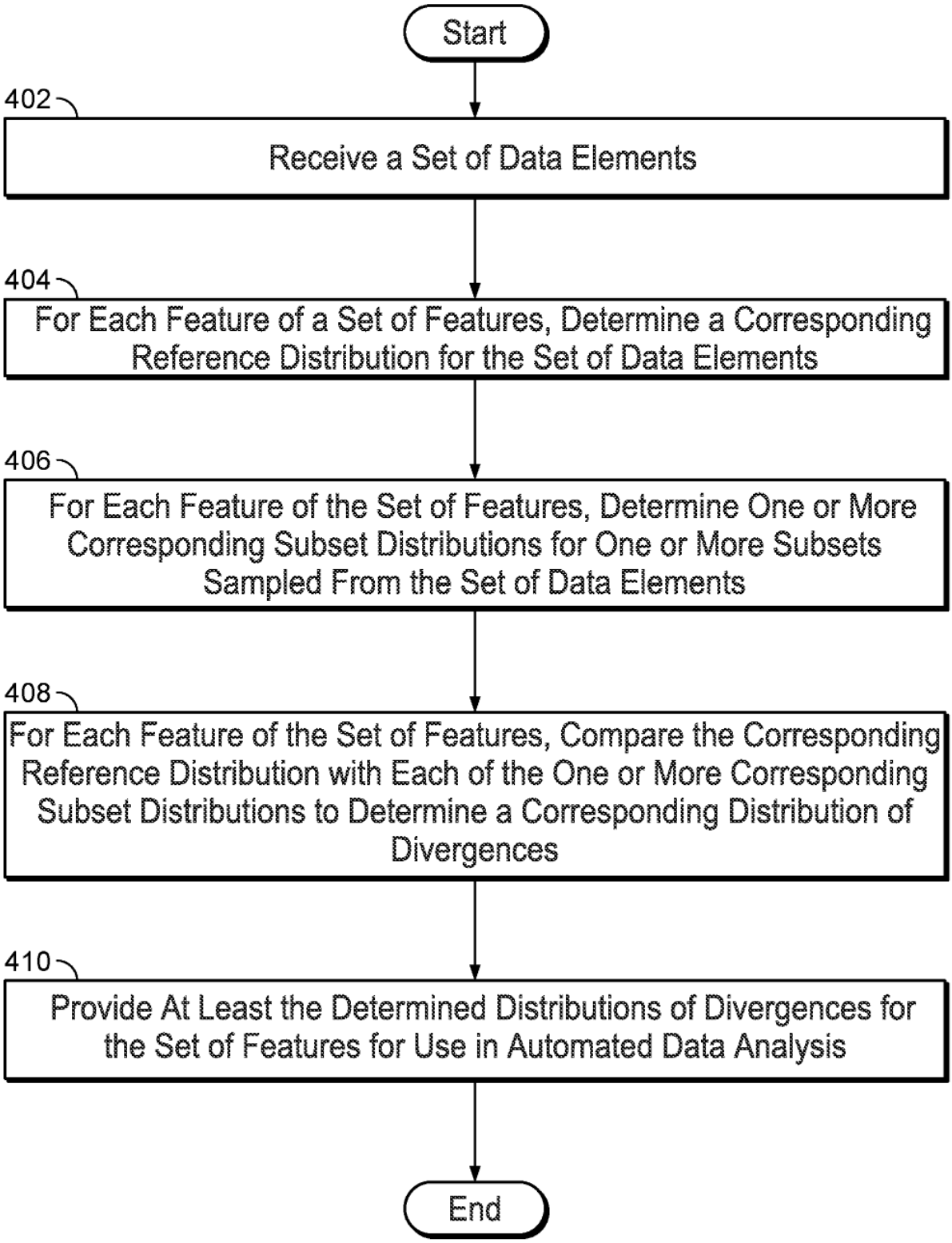
Start
402
Receive a Set of Data Elements
404
For Each Feature of a Set of Features, Determine a Corresponding Reference Distribution for the Set of Data Elements
406
For Each Feature of the Set of Features, Determine One or More Corresponding Subset Distributions for One or More Subsets Sampled From the Set of Data Elements
408
For Each Feature of the Set of Features, Compare the Corresponding Reference Distribution with Each of the One or More Corresponding Subset Distributions to Determine a Corresponding Distribution of Divergences
410
Provide At Least the Determined Distributions of Divergences for the Set of Features for Use in Automated Data Analysis
End

FIG. 4

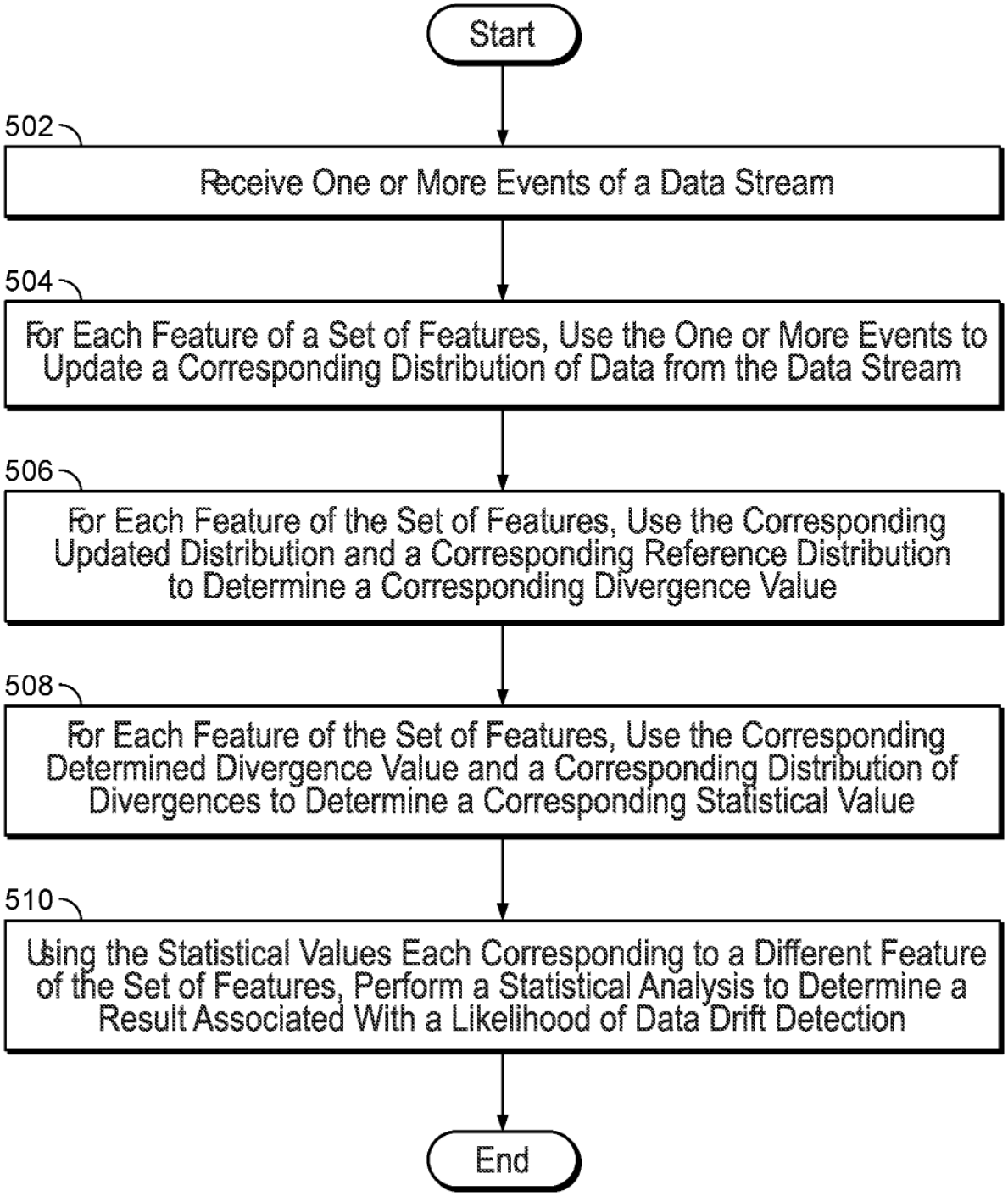
Start
502
Receive One or More Events of a Data Stream
504
For Each Feature of a Set of Features, Use the One or More Events to Update a Corresponding Distribution of Data from the Data Stream
506
For Each Feature of the Set of Features, Use the Corresponding Updated Distribution and a Corresponding Reference Distribution to Determine a Corresponding Divergence Value
508
For Each Feature of the Set of Features, Use the Corresponding Determined Divergence Value and a Corresponding Distribution of Divergences to Determine a Corresponding Statistical Value
510
Using the Statistical Values Each Corresponding to a Different Feature of the Set of Features, Perform a Statistical Analysis to Determine a Result Associated With a Likelihood of Data Drift Detection
End

FIG. 5

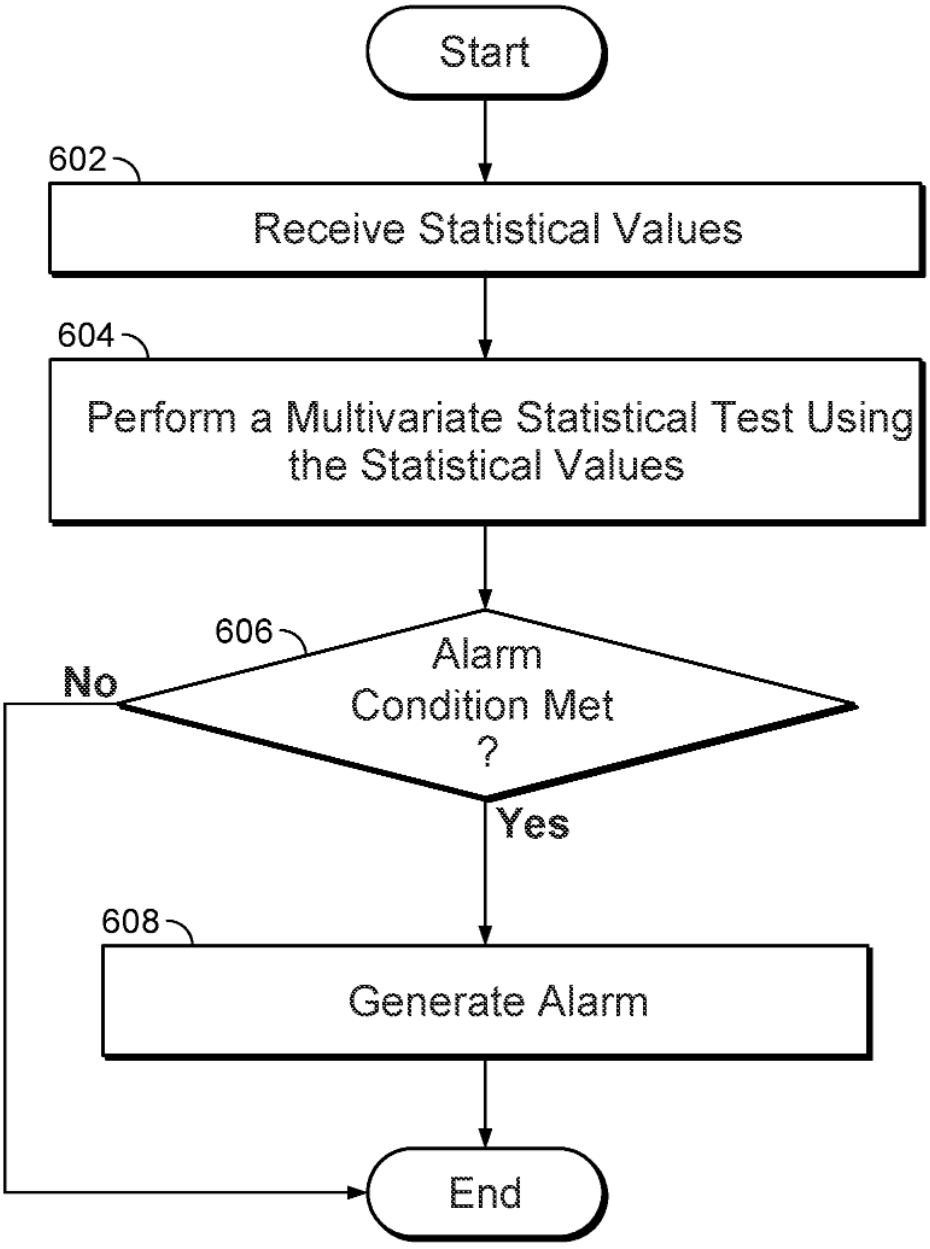
Start
602
Receive Statistical Values
604
Perform a Multivariate Statistical Test Using the Statistical Values
606
Alarm Condition Met ?
No
Yes
608
Generate Alarm
End

FIG. 6

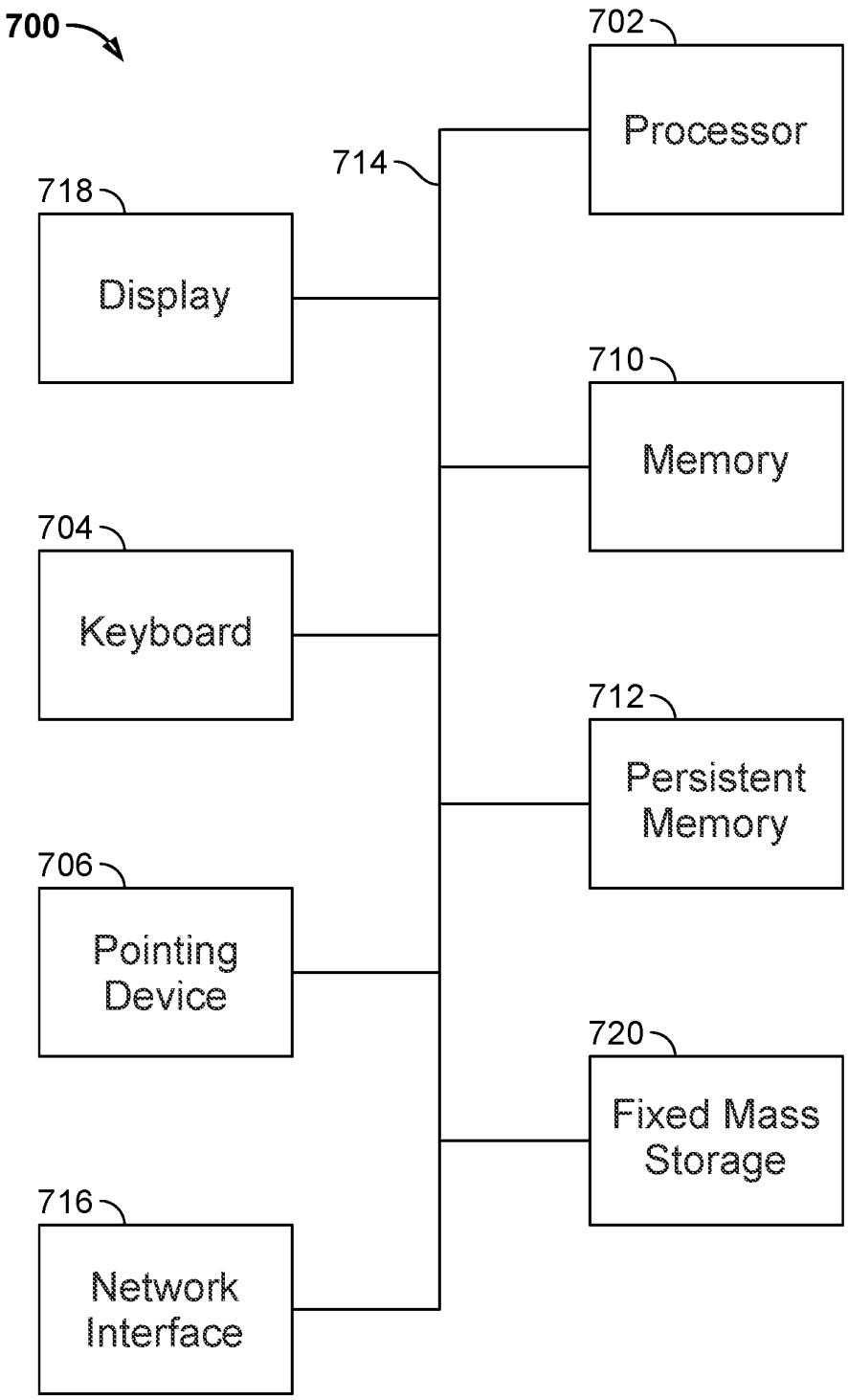
700
702
Processor
714
718
Display
710
Memory
704
Keyboard
712
Persistent Memory
706
Pointing Device
720
Fixed Mass Storage
716
Network Interface

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 21/43490

A. CLASSIFICATION OF SUBJECT MATTER

IPC - G06F 17/18 (2021.01)

CPC - G06F 17/18, G05B 23/024, G05B 23/02, G06Q 10/06, G05B 9/02, G06F 17/18, G07C 3/14, G06T 5/00, G06T 5/40, G06T 15/00, G06N 99/005, H01L 22/20

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2018/0018585 A1 (MICROSOFT TECHNOLOGY LICENSING, LLC) 18 January 2018 (18.01.2018), entire document, (especially abstract, para [0007], [0023], [0027], [0030], [0034], [0037], [0043], [0053], [0062], [0095]).	1-2, 19-20
A	US 2020/0410403 A1 (ROYAL BANK OF CANADA) 31 December 2020 (31.12.2020), entire document, especially abstract.	1-2, 19-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 September 2021 (30.09.2021)	NOV 01 2021
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Kari Rodriquez Telephone No. PCT Helpdesk: 571-272-4300

Form PCT/ISA/210 (second sheet) (July 2019)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 21/43490

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☒ Claims Nos.: 3-18
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.