



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))*

## **FEDERATED EXTERNAL CONTROL ARMS METHOD FOR PRIVACY-ENHANCED CAUSAL INFERENCE ON DISTRIBUTED DATA**

### FIELD OF INVENTION

**[0001]** This invention relates generally to machine learning and more particularly to using federated learning methods to adapt external control arms techniques for privacy enhanced causal inference on distributed data.

### BACKGROUND OF THE INVENTION

**[0002]** Correctly inferring causal relationships, such as treatment effects, has been an active research topic and remains an open problem in clinical research. The current gold standard to establish the effect of a new drug is to randomly assign patients to either a treated group (or arm) or to a control group. Treatment given to patients in the control arm can be placebo or the current standard of care (SOC), against which the new experimental drug will be compared. In this random setting, the populations of different treatment groups are directly comparable and the effect of the drug in terms of target outcomes can be assessed without biases. In healthcare, such procedures are called randomized controlled trials (RCT) and have enabled major progresses in drug discovery over the past decades. However, RCTs are expensive, lengthy and have high failure rates. Less expensive alternatives to RCT exist and leverage external data or real-world evidence (RWE) to form the control arm and perform the comparison, called external control arm (ECA). In this setting, because of the lack of randomization when building the arms, the treatment effect estimation is subject to biases and requires rigorous statistical methodology in order to perform unbiased estimation. An example of such an established methodology is inverse probability weighted treatment effect estimation (IPTW).

**[0003]** A major obstacle to the feasibility of ECA is data sharing. Due to its sensitivity, health data is strictly regulated by, e.g., the General Data Protection Regulation (GDPR) in the EU and the Health Insurance Portability and Accountability Act (HIPAA) in the US and cannot be shared without being anonymized. Even in cases where compliant sharing is technically possible, data can be considered as a strategic asset by the pharmaceutical companies and healthcare centers that could take part in the ECA, which also drastically limits the feasibility to pool data in a

single place. These data sharing obstacles prevent the wider use of ECA, as the associated methods require training models on both control and treated groups.

**[0004]** Federated Learning (FL) is a privacy-enhancing technology (PET) allowing to extract knowledge from multiple institutions without pooling data. This technology allows to perform standard data analyses while being compliant with regulations and protecting intellectual properties (IP). Over the last years, several proof-of-concepts have demonstrated the feasibility of FL-powered privacy-enhanced collaboration between several medical institutions or between pharmaceutical companies.

**[0005]** FL approaches enable the use of ECA in the settings where the external control arm and the experimental treatment arm are held in different institutions. Such settings have received little attention from the community, with no working methodology in the case of time-to-event outcomes that are central in oncology. Methodologies exist in simpler settings where the survival model training (Cox model) can be stratified per-center or for global survival model training without re-weighting but both are inadequate for this setting. When the two arms cannot be pooled, matching adjusted indirect comparison (MAIC) is applicable as it only assumes access to individual patient data from one arm and access to summary statistics from the other arm.

SUMMARY OF THE DESCRIPTION

**[0006]** A method and apparatus of a device that computes a federated IPTW to leverage multicentric data whether real-world data or historical clinical trials as a control group for a single-arm trial while accounting for sampling biases due to the lack of randomization is described. In an exemplary embodiment, the device trains a first model using a plurality of training centers, wherein each training center includes a private set of training data and a compute unit, where the first model estimates the probability of receiving the experimental treatment for each sample given the distributed private set of training data. Furthermore, the device may train a second model using the plurality of training centers, wherein the training of the second model includes computing weights for each sample using the estimated probabilities from the first model, and the second model then estimates the effect of the drug on time-to-event outcomes on weighted data. In addition, the device may use the second model in addition to a third distributed procedure to test whether the experimental drug has a significant effect on the survival outcome in a robust fashion.

**[0007]** Other methods and apparatuses are also described.

**BRIEF DESCRIPTION OF THE DRAWINGS**

**[0008]** The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicate similar elements.

**[0009]** Figure 1 is an illustration of one embodiment of a system for computing a causal effect using federated learning that allows for private data to remain on the premises of its owners while learning global insights. This enhances the privacy of the training and keeps participants in control of their data as opposed to pooling data in a central place and gives equivalent results.

**[0010]** Figure 2 is a flow diagram of one embodiment of a process to compute a causal effect using federated learning that allows for private data to remain on the premises of its owners while learning global insights. This enhances the privacy of the training and keeps participants in control of their data as opposed to pooling data in a central place and gives equivalent results.

**[0011]** Figure 3 is a flow diagram of one embodiment of a process to train a propensity score model.

**[0012]** Figure 4 is a flow diagram of one embodiment of a process to train a Cox model using federated IPTW.

**[0013]** Figure 5 is a flow diagram of one embodiment of a process to perform federated IPTW.

**[0014]** Figure 6 is a flow diagram of one embodiment of a process to compute gradient and hessian components at each center.

**[0015]** Figure 7 is a comparison of one embodiment between pooled IPTW and the method called federated external control arm (FedECA) or federated IPTW.

**[0016]** Figure 8 is a flow diagram of one embodiment of a process to test the Cox model.

**[0017]** Figure 9 illustrates one example of a typical computer system, which may be used in conjunction with the embodiments described herein.

DETAILED DESCRIPTION

**[0018]** A method and apparatus of a device that computes a federated IPTW to leverage multicentric real-world data or historical clinical trials as a control group for a single-arm trial while accounting for sampling biases due to the lack of randomization is described. The method will be denoted as “federated external control arm” (FedECA) or federated IPTW. In the following description, numerous specific details are set forth to provide thorough explanation of embodiments of the present invention. It will be apparent, however, to one skilled in the art, that embodiments of the present invention may be practiced without these specific details. In other instances, well-known components, structures, and techniques have not been shown in detail in order not to obscure the understanding of this description.

**[0019]** Reference in the specification to “one embodiment” or “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiment can be included in at least one embodiment of the invention. The appearances of the phrase “in one embodiment” in various places in the specification do not necessarily all refer to the same embodiment.

**[0020]** In the following description and claims, the terms “coupled” and “connected,” along with their derivatives, may be used. It should be understood that these terms are not intended as synonyms for each other. “Coupled” is used to indicate that two or more elements, which may or may not be in direct physical or electrical contact with each other, co-operate or interact with each other. “Connected” is used to indicate the establishment of communication between two or more elements that are coupled with each other. The term “pooled” will be used to indicate a setting with no privacy constraints where all data can be pooled in a third-party entity. This simpler “pooled” setting will act as a reference to validate the results of the method.

**[0021]** The processes depicted in the figures that follow, are performed by processing logic that comprises hardware (e.g., circuitry, dedicated logic, etc.), software (such as is run on a general-purpose computer system or a dedicated machine), or a combination of both. Although the processes are described below in terms of some sequential operations, it should be appreciated that some of the operations described may be performed in different order. Moreover, some operations may be performed in parallel rather than sequentially.

**[0022]** The terms “server,” “client,” and “device” are intended to refer generally to data processing systems rather than specifically to a particular form factor for the server, client, and/or device.

**[0023]** A method and apparatus of a device that computes a federated IPTW to leverage multicentric real-world data or historical clinical trials as a control group for a single-arm trial while accounting for sampling biases due to the lack of randomization is described. In one embodiment, external control arms (ECA) can inform the early clinical development of experimental drugs and, in specific cases, can be leveraged in the regulatory setting. ECA methods such as inverse probability of treatment weighting (IPTW) allow to leverage a multicentric real-world data or historical clinical trials as a control group for a single-arm trial while accounting for sampling biases due to the lack of randomization. Unfortunately, such multicentric data is often hard to gather in a single place due to stringent health data regulations and the incompatible data governance of pharmaceutical companies and healthcare institutes. In this embodiment, Federated Learning (FL) is used to circumvent this data bottleneck in a privacy-enhancing fashion. This can be described as federated IPTW for time-to-event outcomes and also is denoted as “federated external control arm” (FedECA). Using simulated data, it can be shown that FedECA is equivalent or very close to IPTW on pooled data and provides the same or similar statistical power and type I error for robust estimation. Furthermore, the addition of differential privacy to federated IPTW is studied. In addition, experiments are performed on simulated data using Substra, an open-source Federated Learning software with proven experience in privacy-sensitive contexts.

**[0024]** The drug development cycle suffers from a high attrition rate. In particular, the long and expensive phase III randomized trials that aim at establishing clinical efficacy fail approximately in one case out of two. An ECA that complements a single-arm phase II trial can provide earlier evaluation of efficacy which can inform the transition from a single-arm phase II to a phase III clinical trial. When the externally controlled phase II shows that there is no clinically significant treatment effect, the trial sponsor can decide not to start a phase III. Adoption of ECA to help the sponsor’s internal decision making has the potential to reduce phase III failure rate.

**[0025]** While externally controlled trials will not replace most randomized controlled trials (RCT), there are specific situations where an ECA can be considered by regulatory authorities or health technology assessment authorities because a RCT would be unfeasible. This is the case for rare diseases where there might be insufficient number of patients to run a study as well as in some cases in oncology.

**[0026]** The lack of randomization between the treated arm and the external control arm makes a naive comparison between the two susceptible to confounding bias. Statistical methods that correct for confounding are propensity-score matching, inverse probability of treatment weighting (IPTW), G-computation, and Doubly Debiased Machine Learning. The use of such a method is necessary to provide valid estimates of the treatment effect when using an ECA. Historical clinical trials, insurance claims, and EHR have been considered as the data source for building an ECA.

**[0027]** A major obstacle to the feasibility of ECA is data sharing. Due to its sensitivity, health data is strictly regulated by, e.g., the General Data Protection Regulation (GDPR) in the EU and the Health Insurance Portability and Accountability Act (HIPAA) in the US and cannot be shared without being anonymized. Even in cases where compliant sharing is technically possible, data can be considered as a strategic asset by the pharmaceutical companies and healthcare centers that could take part in the ECA, which also drastically limits the ability to pool data in a single place. These data sharing obstacles prevent the wider use of ECA, as the associated methods require training models on both control and treated groups.

**[0028]** Federated Learning (FL) is a privacy-enhancing technology (PET) allowing to extract knowledge from multiple institutions without pooling data. This technology allows to perform standard data analyses while being compliant with regulations and protecting intellectual properties (IP). Over the last years, several proofs of concepts have demonstrated the feasibility of FL-powered privacy-enhanced collaboration between several medical institutions or between pharmaceutical companies.

**[0029]** As described below, the use of FL approaches to enable ECA when the external control arm and the experimental treatment arm are held in different institutions is described. This problem has received little attention from the community, with no working methodology in the

case of time-to-event outcomes that are central in oncology. When the two arms cannot be pooled, MAIC is also applicable as it only assumes access to individual patient data from one arm and access to summary statistics from the other arm.

**[0030]** In one embodiment, an algorithm is presented to adapt the IPTW methodology to federated datasets, under varying privacy constraints. The proposed methods, FedECA (Federated External Control Arm) and DP-FedECA (Differentially Private FedECA), extend the framework of WebDISCO (Web Distributed Cox model training) to use propensity models in a federated setting with time-to-event outcomes to provide treatment effect and associated robust variance estimation. In one embodiment, Substra, an open-source FL software hosted by the Linux Foundation for AI that was previously used in similar privacy sensitive contexts in healthcare, can be used.

**[0031]** In one embodiment, FedECA is shown to enable ECA for time-to-event outcomes in distributed settings, reaching identical conclusions to IPTW on pooled data including robust estimation of variance. Furthermore, it is also demonstrated that DP-FedECA enables use-specific tradeoffs between accuracy of the method and formal privacy guarantees. In addition, in both cases, the associated runtimes in a deployed distributed system are measured, demonstrating the applicability of the method in real-world settings.

**[0032]** Figure 1 is an illustration of one embodiment of a system 100 for computing a causal effect using federated learning that allows for private data to remain on the premises of its owners while learning global insights and give equivalent results. This enhances the privacy of the training and keeps participants in control of their data as opposed to pooling data in a central place. In Figure 1, the system 100 includes a central aggregator 102 coupled to multiple training centers 104A-N and a “pharma” node 106, where the label “pharma” indicates the owner of the treated arm. The system 100 trains the machine learning model(s) to leverage multicentric real-world data or historical clinical trials coming from 104A-N as a control group for a single-arm trial of the new drug elaborated by the organization owning the data in the “pharma” node 106, while accounting for sampling biases due to the lack of randomization. In one embodiment, each of the central aggregator 102 and training centers 104A-N and 106 can include one of a personal computer, laptop, server (on premise or in the cloud), mobile device (e.g., smartphone, laptop,

personal digital assistant, music playing device, gaming device, etc.), and/or any device capable processing data. In addition, each of the aggregator 102 and training centers 104A-N and 106 can include either a virtual or a physical device. In one embodiment, a machine learning model (or simply a model) is a potentially large file containing the parameters of a trained model.

**[0033]** In one embodiment, each of the training centers 104A-N includes data from historical patients that can be used for training the model(s) using FedECA and training center 106 owns the data from the treated arm. In this embodiment, one or more of the training centers are used to construct the external control arm. In one embodiment, each of the training centers 104A-N from the control arm can be hospitals or medical institutions that collect the patient training data.

**[0034]** In one embodiment, FedECA uses two different models, a propensity score model and a Cox model. As both models are generalized linear models, a model refers to a set of weights that describes the model. A trained model is the result of training a model with a given set of training data. In one embodiment, the aggregator 102 manages the communication across centers and the updates of the model. In another embodiment, a third-party might be responsible for the orchestration of the distributed computation. In this embodiment, the central aggregator 102 receives the quantities computed in each center, aggregates them before redistributing the updated quantities, where the updated quantities now include collective insights from all the training centers 104A-N and 106 to each of the individual training centers 104A-N and 106.

**[0035]** In one embodiment, each of the training centers 104A-N and 106 are used to compute components used for training of the two models. In this embodiment, the training centers 104A-N and 106 each receive the same propensity model initialization which is a deterministic set of weights or a random set of weights that it will use for computing the components used to train the models, computes these components, and returns the components to the aggregator 102. For example, and in one embodiment, each of the training centers 104A-N and 106 receives a vector full of zeros as initialization for the propensity model and use it to compute a local gradient and Hessian of the current model using the training data that each of the training centers 104A-N and 106 holds. In one embodiment, the raw training data of each of the training centers 104A-N and 106 remains undisclosed to any of the other participants involved including the aggregator 102. In one embodiment, the aggregated quantities (gradients, Hessians and building blocks thereof)

are shared, whereas the private data for each of the training centers 104A-N remains private to that training center 104A-N. The aggregator averages the local Hessians and gradients and redistributes them back to the training centers 104A-N and 106. This procedure is repeated multiple times until convergence. With the trained propensity score model and the privately held data, each of the training centers 104A-N and 106 computes components (scores or weighting) that are used to train the Cox model in a similar fashion, although the components shared by the training centers are not exactly the same. While in one embodiment, there are two training centers 104A-N are illustrated, in other embodiments, there can be more or less training centers.

**[0036]** To illustrate the performance of the proposed FL implementation, simulations with synthetic data are relied upon. Covariates and related time-to-event outcomes respecting the proportional hazards (PH) assumption, with the baseline hazard function derived from a Weibull distribution are simulated. For simplicity, in this embodiment, a constant treatment allocation is assumed across the population within a training center but in another embodiment training centers could have patient data with both treated and untreated patients. The data generation process consists of several consecutive steps that are described below assuming our target is a dataset with  $p$  covariates and  $n$  samples. The synthetic data generation procedure of this embodiment is only used to illustrate the capabilities of the method, which should operate the same way on any synthetic or non-synthetic data distribution irrespective of the details of the generation.

**[0037]** First, a design matrix  $X \in R^{n \times p} \sim N(0, \Sigma)$  is drawn from a multivariate normal distribution to obtain (baseline) observations for  $n$  individuals described by  $p$  covariates. The covariance matrix  $\Sigma$  is taken to be a Toeplitz matrix, such that the correlation coefficients of the covariates decay geometrically. This implies that, for a fixed  $\rho > 0$ , the correlation between two variables  $X_i$  and  $X_j$  is given by  $\rho^{|i-j|}$ . Such a covariance matrix implies a locally and hierarchically grouped structure underlying the covariates, which we choose to mimic the potentially complex structure of real-world data. To reflect the varying correlations of the covariates with the outcome of interest, the coefficients  $\beta_i$  of the linear combination used to build the hazard ratio are drawn from a standard normal distribution.

$$\Sigma = \text{Toeplitz}(1, \rho, \rho^2, \dots, \rho^{p-1}),$$

$$\begin{aligned} X &\in R^{n \times p} \sim N(0, \Sigma), \\ \beta &\in R^p \sim N(0, 1). \end{aligned} \quad (1)$$

**[0038]** Since it is considered the case of external control arms that implies non-randomized treatment allocation, the treatment allocation variable  $A$  is simulated in such a way that it depends on the covariates  $X_1, \dots, X_p$ . More precisely, it is assumed that  $A$  follows a Bernoulli distribution, where the probability of being treated (the propensity score)  $q$  depends on a linear combination of the covariates, connected by a logit link function  $g$ . The coefficients  $\alpha_i$  of the linear combination are drawn from a uniform distribution, where the range  $k \geq 0$  is symmetric around 0 and is normalized by the number of covariates. The degree of influence of the covariates on  $A$  can be regulated by adjusting the value of  $k$ . The greater the value of  $k$ , the stronger the influence, and therefore the lower the degree of overlap between the distributions of propensity scores of the treated and (external) control groups. Conversely,  $k = 0$  removes the dependence, leading to a randomized treatment allocation.

$$\begin{aligned} \alpha_i &\in R^p \sim p^{-1/2} U(-k, k), \\ q_i &= g^{-1}(\alpha_i X_i) = (1 + e^{-\alpha_i X_i})^{-1}, \\ z_i | X_i &\sim \text{Bern}(q_i). \end{aligned} \quad (2)$$

**[0039]** Once drawn, the treatment allocation variable  $A_i$  is composed with the constant treatment effect, defined here as the hazard ratio  $\mu$ , to obtain the final hazard ratio  $h_i$  for each individual. The time-to-event  $T_i^*$  of each sample is then drawn from a Weibull distribution with shape  $v$  and the scale depending on  $h_i$  and  $v$ . Meanwhile, for all samples, a constant dropout (or censoring) rate  $d$  across time is assumed, resulting in a censoring time that follows an exponential distribution.

$$\begin{aligned} h_i(a_i) &= \mu^{a_i} \exp(\beta_i X_i), \\ T_i^* &\sim W(h_i(a_i)^{\frac{1}{v}}, v) \\ C_i &\sim E(d) \end{aligned} \quad (3)$$

**[0040]** Finally, from  $T_i^*$  and  $C_i$ , the status variable  $\delta_i$  can immediately be obtained:  $\delta_i = \mathbb{1}_{T_i^* \leq C_i}$ . Additionally, the possibility to specify a percentage of ties is implemented, e.g., the percentage of individuals who share a time- to-event. The observed outcome  $Y_i$  for the  $i^{\text{th}}$  individual is

defined as the couple  $Y_i = (T_i = \min(T_i^*, C_i), \delta_i)$ , i.e., it corresponds to the observed time and the information whether this corresponds to the time-to-event or the censoring time.

**[0041]** In one embodiment, FedECA comprises three major steps: (1) training a propensity score model; (2) training a federated weighted Cox model; and (3) using the quantities produced by the Cox model training in a third federated procedure in order to compute a robust p-value, which gives a significance score of the treatment effect. In this embodiment, each patient is represented by covariates  $X \in \mathbb{R}^p$ . It undergoes treatment  $A \in \{0, 1\}$ , corresponding either to the treated ( $A = 1$ ) or control ( $A = 0$ ) arm. Following treatment, the patient has an event of interest (e.g., death or disease relapse) at a random time  $T^*$ . The patient may leave the arm before the event of interest is actually observed, a phenomenon called censoring: the *observed* time  $T$  is denoted, whose realizations are denoted  $t_i$ . It is noted  $\delta_i = 1$  if this corresponds to a true event, respectively  $\delta_i = 0$  if censorship took place. Let  $n$  denote the total number of patients, indexed by  $i$ .

**[0042]** Let  $S$  denote the finite set of all potential times, e.g.,  $S = \{t_i\}^n$ . At a given time  $s$ , let  $D_s$  denote the set of patients with an event at this time, e.g.,

$$\forall s \in S, D_s = \{i | t_i = s, \delta_i = 1\} \quad (4)$$

and let  $R_s$  denote the set of patients *at risk* at this time (also called “risk set”), i.e.

$$\forall s \in S, R_s = \{i | t_i \geq s\}. \quad (5)$$

Further, let  $\dot{S}$  denote the set of times where at least one true event occurs, i.e.

$$\dot{S} = \{s \in S | D_s \neq \emptyset\} \quad (6)$$

**[0043]** Data is distributed among  $K$  different training centers, with  $n_k$  samples per center. It is denoted  $x_{i,k}$  as the  $i^{th}$  covariates from the  $k^{th}$  center; accordingly,  $a_{i,k}$  denotes the treatment allocation,  $t_{i,k}$  the observed time event, and  $\delta_{i,k}$  whether a true event took place. Similarly, for each time  $s$  and center  $k$ , the subset  $D_{s,k}$  and  $R_{s,k}$  are defined as the respective restrictions of  $D_s$  and  $R_s$  to center  $k$ .

**[0044]** Figure 2 is a flow diagram of one embodiment of a process 200 to compute a causal effect using federated learning that maintains privacy of the training data. In one embodiment, process 200 is performed by an aggregator of a federated learning system, such as the aggregator 102. Figure 2 begins by process 200 receiving input data at block 202. In one embodiment, process 200 receives configuration information that is used to perform the FedECA computation.

[0045] At block 204, process 200 trains a propensity score model. In one embodiment, due to the lack of randomization, for each sample, the probability of being assigned the treatment A depends on the covariates  $\mathbf{X}$ . a propensity score model  $p_\theta$  with parameters  $\theta$  is trained such that

$$p_\theta(\mathbf{x}) \approx P[A | \mathbf{X} = \mathbf{x}]. \quad (7)$$

[0046] In one embodiment, a logistic model is used for  $p_\theta$ , e.g.,

$$p_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)} \quad (8)$$

Propensity score model training is further described below with reference to Figure 3.

[0047] At block 206, process 200 trains a Cox model using federated IPTW to generate a final set of weights and a Hessian. In one embodiment, for each sample  $i$ , an IPTW weight is defined as  $w_i \in (0, +\infty)$  based on the propensity score model trained in the previous block as

$$w_i = \begin{cases} \frac{1}{p_\theta(x_i)} & \text{if } a_i = 1, \\ \frac{1}{1-p_\theta(x_i)} & \text{otherwise} \end{cases} \quad (9)$$

[0048] In order to avoid overflow errors, in one embodiment one might clip the weights  $w_i$  using 99% quantiles using federated quantile computations (federated averaging applied on the pinball loss). In this embodiment, weights are clipped at  $10^{-16}$  (or another clipping threshold) to avoid overflow errors.

[0049] A weighted Cox proportional hazards (CoxPH) model is trained with parameters  $\beta \in \mathbb{R}^q$ , related to patient-specific variables  $z_i \in \mathbb{R}^q$ . In one embodiment, the variables  $z_i$  are not the same as the covariates  $x_i$ . More precisely, for the vanilla IPTW method, the covariate used is the treatment allocation, e.g.,  $z_i = a_i$ . In another embodiment for the *Adjusted* IPTW (adjIPTW) method, additional covariates can be used, especially if they are known confounders. For simplicity, and in one embodiment, vector notations are used to support both cases.

[0050] The CoxPH model is fitted by maximizing a data-fidelity term consisting in the partial likelihood  $L(\beta)$  with Breslow approximation.

$$L(\beta) = \prod_{i: \delta_i=1} \left( \frac{e^{\beta^T z_i}}{\sum_{j: r_j \geq r_i} w_j e^{\beta^T z_j}} \right)^{w_i} = \prod_{s \in S} \prod_{i \in D_s} \left( \frac{e^{\beta^T z_i}}{\sum_{j \in R_s} w_j e^{\beta^T z_j}} \right)^{w_i} \quad (10)$$

where the second equation has been rewritten using the sets  $D_s$  and  $R_s$ . For numerical stability, the log-likelihood  $\ell(\beta) = \log L(\beta)$  is used, which reads

$$l(\beta) = \sum_{s \in \mathcal{S}} \sum_{i \in D_s} \left\{ w_i \beta^T z_i - w_i \log \log \left( \sum_{j \in R_s} w_j e^{\beta^T z_j} \right) \right\} \quad (11)$$

Minimizing this loss in the federated setting is further described below with reference to Figure 4. Process 200 tests the Cox model block 208. Testing the Cox model is further described in Figure 8 below.

**[0051]** In one embodiment, using the fitted weights  $\hat{\beta}$ , process 200, in block 208, estimates the variance matrix of  $\hat{\beta}$  using the Hessian of  $\ell$ ,  $\nabla^2 \ell(\hat{\beta})$ , as well as the quantities  $\zeta_s^0(\hat{\beta})$ ,  $\zeta_s^1(\hat{\beta})$  and  $\sum_{j \in D_{s'}} \hat{w}_j$ , by constructing a robust sandwich-type variance estimator of the form.

$$\widehat{Var}(\hat{\beta}) = H^{-1} Q (H^{-1})^T \quad (12)$$

where  $H = \nabla^2 \ell(\hat{\beta})$  and

$$Q = \sum_{i=1}^n \hat{\phi}_i(\hat{\beta}) \hat{\phi}_i(\hat{\beta})^T \quad (13)$$

and

$$\begin{aligned} \hat{\phi}_i(\hat{\beta}) = & \delta_i \hat{w}_i \left( z_i - \frac{\zeta_s^1(\hat{\beta})}{\zeta_s^0(\hat{\beta})} \right) - \hat{w}_i e^{\hat{\beta}^T z_j} z_i \sum_{s' \in \mathcal{S}} \sum_{j \in D_{s'}} \hat{w}_j \frac{1_{\{s' \leq s\}}}{\zeta_{s'}^0(\hat{\beta})} + \\ & \hat{w}_i e^{\hat{\beta}^T z_j} \sum_{s' \in \mathcal{S}} \sum_{j \in D_{s'}} \hat{w}_j \frac{1_{\{s' \leq s\}} \zeta_{s'}^1(\hat{\beta})}{\zeta_{s'}^0(\hat{\beta})^2} \end{aligned} \quad (14)$$

where we have  $t_i = s$ , with  $1_{\{s' \leq s\}}$  the indicator function that is equal to 1 only on times  $s'$  (with events) smaller or equal to  $s$ , and 0 otherwise. The definitions of  $\zeta_s^k(\beta)$ ,  $k=0, 1$  are given in equations 26 and 27.

Note that  $\zeta_s^k(\beta)$  intervene only for  $s$  in  $\hat{S}$ .

In addition, a Wald test might be performed on the entry of  $\beta$  corresponding to the treatment allocation, following a  $\chi^2$  distribution with 1 degree of freedom.

**[0052]** As described above, the propensity score model from 204 is used as input to the Cox model training. Figure 3 is a flow diagram of one embodiment of a process 300 to train a propensity score model. In one embodiment, process 300 fits a model to obtain the propensity score, Eq. (9,) based on distributed data  $\{(x_{i,k}, a_{i,k})\}_{i=1}^K$ . Let  $J$  denote the full negative log-likelihood of the model, and  $J_k$  the log-likelihood for each center is

$$J_k(\theta) = \sum_{i=1}^{n_k} \left\{ a_{i,k} \log \log p_\theta(x_{i,k}) + (1 - a_{i,k}) \log \log (1 - p_\theta(x_{i,k})) \right\} \quad (15)$$

Due to the separability of each loss term in per-sample terms, Eq. 15 can be written as

$$J(\theta) = \sum_{k=1}^K J_k(\theta) \quad (16)$$

**[0053]** Using the separability (16), it is straightforward to optimize  $J$  using a second-order method, since its gradient and hessian can be computed from the sum of local quantities, as described above. This strategy is described in Figure 3. In addition, Figure 3 includes a hyperparameter corresponding to the number of steps. While in one embodiment,  $E = 10$  is sufficient to obtain proper convergence, in another embodiment, another value can be used as a maximum number of steps. In one embodiment, this strategy computes full batch gradients and Hessians, in time  $O(n_k)$  on each center, and each communication with the aggregator requires the exchange of  $O(p^2)$  floating numbers. In the setting of ECAs, in this embodiment, we might have  $n_k \leq 10,000$  and  $p \leq 50$ , making such a second approach tractable.

**[0054]** Figure 3 begins by receiving the initial propensity score model at block 302. At block 304, process 300 initializes the propensity model parameters. In one embodiment, process 300 zeroes the parameters of the propensity model. Process 300 sets the initialized propensity model to the current model at block 306.

**[0055]** At block 308, process 300 sends the current model parameters to each of the training centers, where each training center computes a gradient and Hessian of the model using the current model parameters. In one embodiment, each  $k^{th}$  of the training centers computes the gradient using Eq. (17):

$$g_{c,k} = \nabla_\theta J_k(\theta_{c-1}) \quad (17)$$

and computing the Hessian using Eq. (18):

$$H_{c,k} = \nabla_\theta^2 J_k(\theta_{c-1}) \quad (18)$$

In one embodiment, the  $k^{th}$  training center computes  $J_k(\theta_{c-1})$  using the Eq. (15) and the historical data that is private to the  $k^{th}$  training center. With the gradient and Hessian computed, each of the training centers returns the gradient and Hessian to process 300. Alternatively, process 300 can compute a gradient without computing the Hessian by adding noise to the gradient computation. In one embodiment, process 300 modifies the gradient computation  $g_{c,k}$

described in (17) by stochastically sampling with replacement a batch of size  $B < \min(n_k)$  in each training center then restricting the computation of the gradient  $\nabla_{\theta} J_k(\theta_{c-1})$  to this batch, then clipping the norm of each individual sample's gradients to a value  $C$  and adding differential privacy noise on all individual gradients. At block 310, process 300 receives the gradients and Hessians computed by each training center. In another embodiment, if process 300 computes just the gradient by adding noise as described above, process 300 receives the gradients and not the Hessian.

**[0056]** Process 300 computes the global gradient and Hessian from the individual gradient and Hessian components received from the training centers at block 312. In one embodiment, process 300 computes an average of the gradient and Hessian components using Eq. (19) and (20):

$$g_c = \frac{1}{K} \sum_{k=1}^K g_{c,k} \quad (19)$$

$$H_c = \frac{1}{K} \sum_{k=1}^K H_{c,k} \quad (20).$$

In another embodiment Process 300 receives only the (noisy) gradients from each of the training centers, and process 300 just computes the global gradient using (19) above. At block 314, process 300 computes the updated propensity score model parameters. In one embodiment, process 300 computes the updated propensity score model parameters using the Eq. (20):

$$\theta_c = \theta_{c-1} - (H_c)^{-1} g_c \quad (21)$$

**[0057]** Alternatively, if process 300 just computes a global gradient, process 300 computes the updated propensity score model parameters using  $\theta_c = \theta_{c-1} - \lambda g_c$ . Where  $\lambda$  is a given value called the learning rate.

**[0058]** At block 316, process 300 determines if the model has converged. In one embodiment, process 300 determines that the model has converged if the number of loops has reached a max loop threshold. In another embodiment, process 300 can determine if the model has converged using a different scheme, such as inspecting the norm of the hessian, the difference between the current likelihood and the likelihood from the round just before, or another scheme to determine convergence. If the model has converged, process 300 outputs the updated propensity score

model at 318. If the model has not converged, process 300 sets the updated model to the current model at block 320. Execution proceeds to block 308 above.

**[0059]** In one embodiment, the propensity score model that is computed in Figure 3 is used for the training of the Cox model. Figure 4 is a flow diagram of one embodiment of a process 400 to train a Cox model using federated IPTW. As described above with reference to Eq. (11), the log-likelihood equation has cross-terms that make it difficult to spread the computation of the log-likelihood across different training centers. As described below, a method to minimize the regularized weighted CoxPH model in a federated fashion is described. Since the non-separability of the weighted CoxPH log-likelihood  $\ell(\beta)$  prevents the use of a straightforward FL algorithms, a modified Web Distributed Cox model (DISCO) approach is used to build a pooled-equivalent second-order method.

**[0060]** In one embodiment, compared to the logistic propensity score model, the main difficulty of federating Equation (11) stems from the non-separability of the log-likelihood component because of the cross-center terms. Indeed, for any time  $s$ , the risk set  $R_s$  is a union of per-center terms,

$$R_s = \bigcup_{k=1}^K R_{s,k} \quad (22)$$

Thus, the aggregated Eq. (11) can be rewritten as

$$\ell(\beta) = \sum_{k=1}^K \left\{ \sum_{s \in S} \sum_{i \in D_{s,k}} \left\{ w_i \beta^T z_{i,k} - w_i \log \log \left( \sum_{j \in R_{s,k}} w_j e^{\beta^T z_{j,k}} + \sum_{k' \neq k} \sum_{j \in R_{s,k'}} w_j e^{\beta^T z_{j,k'}} \right) \right\} \right\} \quad (23)$$

where the loss for each sample  $i$  of each training center  $k$  involves terms from other samples  $j$  in other centers  $k' \neq k$ . The non-separability of the CoxPH loss is a well-known issue in a federated setting and previous works have investigated reformulations to make it amenable to vanilla federated learning solvers. Here, a modified WebDISCO method to the weighted case in order to keep pooled-equivalent results and benefit from second-order acceleration is used.

**[0061]** In one embodiment, the modified WebDISCO includes performing an iterative server-level Newton-Raphson descent on  $L$ . The gradient  $\nabla_{\beta} l(\beta)$  and Hessian  $\nabla_{\beta}^2 l(\beta)$  thus needs to be computed in a federated fashion. These quantities can be computed in closed form as

$$\nabla_{\beta} l(\beta) = \sum_{s \in \mathcal{S}} \sum_{i \in D_s} w_i z_i - w_i \frac{\sum_{j \in R_s} w_j e^{\beta^T z_j z_j}}{\sum_{j \in R_s} w_j e^{\beta^T z_j}} \quad (24)$$

and

$$\nabla_{\beta}^2 l(\beta) = - \sum_{s \in \mathcal{S}} \sum_{i \in D_s} w_i \left\{ \frac{\sum_{j \in R_s} w_j e^{\beta^T z_j z_j} z_j z_j^T}{\sum_{j \in R_s} w_j e^{\beta^T z_j}} - \frac{\left( \sum_{j \in R_s} w_j e^{\beta^T z_j z_j} \right) \left( \sum_{j' \in R_s} w_{j'} e^{\beta^T z_{j'} z_{j'}^T} \right)}{\left( \sum_{j \in R_s} w_j e^{\beta^T z_j} \right)^2} \right\} \quad (25)$$

**[0062]** In one embodiment, the following quantities aggregated on risk sets, can be defined as:

$$\zeta_s^0(\beta) = \sum_{j \in R_s} w_j e^{\beta^T z_j} \quad (26)$$

$$\zeta_s^1(\beta) = \sum_{j \in R_s} w_j e^{\beta^T z_j} z_j \quad (27)$$

$$\zeta_s^2(\beta) = \sum_{j \in R_s} w_j e^{\beta^T z_j} z_j z_j^T \quad (28)$$

Each of these quantities has an associated local component that is computed on each of the training centers:

$$\zeta_{s,k}^0(\beta) = \sum_{j \in R_{s,k}} w_j e^{\beta^T z_j} \quad (29)$$

$$\zeta_{s,k}^1(\beta) = \sum_{j \in R_{s,k}} w_j e^{\beta^T z_j} z_j \quad (30)$$

$$\zeta_{s,k}^2(\beta) = \sum_{j \in R_{s,k}} w_j e^{\beta^T z_j} z_j z_j^T \quad (31)$$

In addition, define:

$$W_s = \sum_{i \in D_s} w_i \quad (32)$$

$$Z_s = \sum_{i \in D_s} w_i z_i \quad (33)$$

as well as  $W_{s,k}$  and  $Z_{s,k}$  where the sum is restricted to the set of  $z_i$  owned by training center  $k$  ( $i \in D_{s,k}$ ), where by convention, the sum is set to zero in the case of an empty set. Equations (23) and (24) can be re-written as:

$$\nabla_{\beta} l(\beta) = \sum_{s \in \dot{S}} Z_s - W_s \frac{\zeta_s^1(\beta)}{\zeta_s^0(\beta)} \quad (34)$$

and (25) as:

$$\nabla_{\beta}^2 l(\beta) = - \sum_{s \in \dot{S}} W_s \left\{ \frac{\zeta_s^2(\beta)}{\zeta_s^0(\beta)} - \frac{\zeta_s^1(\beta) \zeta_s^1(\beta)^T}{\zeta_s^0(\beta)^2} \right\} \quad (35)$$

In one embodiment, re-writing equations (34) and (35) for multiple training centers, the gradient  $\nabla_{\beta} l(\beta)$  and Hessian  $\nabla_{\beta}^2 l(\beta)$  can be expressed as

$$\nabla_{\beta} l(\beta) = \sum_{k=1}^K \left\{ \sum_{s \in \dot{S}} Z_{s,k} - W_{s,k} \frac{\sum_{k'} \zeta_{s,k'}^1(\beta)}{\sum_{k'} \zeta_{s,k'}^0(\beta)} \right\} \quad (36)$$

$$\nabla_{\beta}^2 l(\beta) = - \sum_{k=1}^K \sum_{s \in \dot{S}} W_{s,k} \left\{ \frac{\sum_{k'} \zeta_{s,k'}^2(\beta)}{\sum_{k'} \zeta_{s,k'}^0(\beta)} - \frac{\left( \sum_{k'} \zeta_{s,k'}^1(\beta) \right) \left( \sum_{k'} \zeta_{s,k'}^1(\beta) \right)^T}{\left( \sum_{k'} \zeta_{s,k'}^0(\beta) \right)^2} \right\} \quad (37)$$

Assuming the set of all true events  $\dot{S}$  is known to all clients, it is possible to reconstruct the full gradient  $\nabla_{\beta} l(\beta)$  and Hessian  $\nabla_{\beta}^2 l(\beta)$  based on the 5-uplet

$\left\{ (W_{s,k}), (Z_{s,k}), (\zeta_{s,k}^0(\beta)), (\zeta_{s,k}^1(\beta)), (\zeta_{s,k}^2(\beta)) \right\}_{s,k}$ . This algorithm is embodied in Figure 4 below.

**[0063]** Figure 4 begins by initializing the weights of the Cox model at block 402. In one embodiment, process 400 initializes the weights by setting the weights to zero. In another embodiment process 400 could initialize the weights to any other deterministic value or generate random weights. At block 404, process 400 sets the initial weights to be the current weights.

**[0064]** Process 400 computes an updated gradient and Hessian at block 406. In one embodiment, process 400 computes the updated gradient and Hessian as described in Figure 5 below. At block 408, process 400 computes the updated weights using the updated gradient and Hessian. In one embodiment, process 400 computes the updated weights using the equation:

$$\beta_e = \beta_{e-1} - \alpha_e \left( \nabla_{\beta}^2 L(\beta_{e-1}) \right)^{-1} \left( \nabla_{\beta} L(\beta_{e-1}) \right) \quad (38)$$

where  $\nabla_{\beta}^2 L(\beta_{e-1})$  and the  $\nabla_{\beta} L(\beta_{e-1})$  are the Hessian and gradients, respectively, that are computed using the following equations:

$$\nabla_{\beta} L(\beta_{e-1}) = \nabla_{\beta} l(\beta_{e-1}) + \gamma \nabla_{\beta} \psi(\beta) \quad (39)$$

$$\nabla_{\beta}^2 L(\beta_{e-1}) = \nabla_{\beta}^2 l(\beta_{e-1}) + \gamma \nabla_{\beta}^2 \psi(\beta) \quad (40)$$

and  $\nabla_{\beta} l(\beta_{e-1})$  and  $\nabla_{\beta}^2 l(\beta_{e-1})$  are the gradient and Hessian components that have been computed using a distributed procedure by the training centers illustrated in Fig. 5 described below.

**[0065]** At block 410, process 400 determines if the model has converged. In one embodiment, process 400 determines that the model has converged if the number of loops has reached a max loop threshold. In another embodiment, process 400 determines if the model has converged using a different scheme such as inspecting the norm of the global Hessian or computing the improvement over the current log-likelihood. If the model has converged, process 400 outputs the updated model weights and Hessian at block 412. If the model has not converged, process 400 sets the updated model to the current model at block 414. Execution proceeds to block 406 above.

**[0066]** As described above, a distributed procedure is used to compute the gradient and Hessian components. Illustrated below is the distributed procedure necessary to obtain global gradient and hessian used at block 408. Figure 5 is a flow diagram of one embodiment of a process 500 to compute gradient and Hessian components for a federated IPTW. In one embodiment, process 500 is performed by an aggregator to compute the gradient and Hessian components, such as the

aggregator 106 as described in Figure 1 above. In Figure 5, process 500 begins by receiving the current weights and the set  $S$  at block 502.

**[0067]** At block 504, process 500 requests a computation of  $W_{k,s}$ ,  $Z_{k,s}$ , and the aggregated quantities over the risk set components ( $\zeta_{s,k}^0(\beta)$ ,  $\zeta_{s,k}^1(\beta)$ ,  $\zeta_{s,k}^2(\beta)$ ) from the training centers. Computing  $W_{k,s}$ ,  $Z_{k,s}$ , and the risk set components ( $\zeta_{s,k}^0(\beta)$ ,  $\zeta_{s,k}^1(\beta)$ ,  $\zeta_{s,k}^2(\beta)$ ) is further described in Figure 6 below. Process 500 receives  $W_{k,s}$ ,  $Z_{k,s}$ , and the risk set components ( $\zeta_{s,k}^0(\beta)$ ,  $\zeta_{s,k}^1(\beta)$ ,  $\zeta_{s,k}^2(\beta)$ ) from the training centers at block 506. With these components, process 500 computes the gradient, Hessian, and risk set components at block 508. In one embodiment, process 500 uses Equation (36) and (37) to compute gradient and Hessian with  $W_{k,s}$ ,  $Z_{k,s}$ , and the risk set components ( $\zeta_{s,k}^0(\beta)$ ,  $\zeta_{s,k}^1(\beta)$ ,  $\zeta_{s,k}^2(\beta)$ ). At block 510, process 500 determines if the computation is completed. In one embodiment, the computation is completed based on achieving a maximum number of steps. In alternative embodiments, the computation converges based on a threshold. If the computation is converged, process 500 returns the Cox model, Hessian and risk set components at block 512. If the computation is not converged, execution proceeds to block 504 above.

**[0068]** As described above, each training center can compute intermediate quantities that are used by the aggregator to compute a Hessian and risk set components for the Cox model. Figure 6 is a flow diagram of one embodiment of a process 600 to compute Hessian and risk set components of the Cox model at each center. In one embodiment, Figure 6 begins by receiving the weights and the set  $\dot{S}$  that are used to compute the intermediate data at block 602. At block 604, process 600 computes  $W_{k,s}$ ,  $Z_{k,s}$ , for each item in its training data that is in set  $\dot{S}$ . In one embodiment, process 600 computes  $W_{k,s}$ ,  $Z_{k,s}$ , using Equations (32) and (33) above. Process 600, for each item in set  $\dot{S}$  such that  $W_{k,s} > 0$ , computes the risk set components. In one embodiment, process 600 computes ( $\zeta_{s,k}^0(\beta)$ ,  $\zeta_{s,k}^1(\beta)$ ,  $\zeta_{s,k}^2(\beta)$ ) using Equations (29), (30), and (31) respectively. Process 600 returns  $W_{k,s}$ ,  $Z_{k,s}$ , and the risk set components ( $\zeta_{s,k}^0(\beta)$ ,  $\zeta_{s,k}^1(\beta)$ ,  $\zeta_{s,k}^2(\beta)$ ) at block 608.

**[0069]** As described above in Fig. 2, block 208, the Cox model is tested. In one embodiment, the testing of the Cox model is distributed across the multiple training centers. Figure 8 is a flow diagram of one embodiment of a process 800 to test the Cox model. In one embodiment, process

800 begins by receiving the outputted Cox model weights, the Hessian and the risk set components at block 802. In this embodiment, process 800 receives the outputted Cox model weights, the Hessian and the risk set components as described in Figure 4, block 412 above. As part of this process,  $\beta$  is now fixed to the final value  $\hat{\beta}$ . At block 804, process 800 computes  $M_K$  defined in (43). In one embodiment, process 800 computes  $M_K$  by sending the Cox model weights  $\hat{\beta}$ , the Hessian, and the risk set components and requesting each of the training centers to compute a corresponding  $M_K$  using Cox model weights  $\hat{\beta}$ , the Hessian, and the risk set components. In this embodiment, each training center computes  $M_K$  using (43) below.

**[0070]** In one embodiment, by using (12), (13) and (14), a robust sandwich estimator of the variance can be derived:

$$\begin{aligned} \text{Var}(\hat{\beta}) &= H^{-1} Q (H^{-1})^T \\ Q &= \sum_{i=1}^n \phi_i(\hat{\beta}) \phi_i(\hat{\beta})^T \end{aligned}$$

This computation can be factorized per center in the following fashion, we have:

$$\text{Var}(\beta) = \sum_{i=1}^n H^{-1} \phi_i(\hat{\beta}) ((H^{-1}) \phi_i(\hat{\beta}))^T \quad (41)$$

Thus, we can write:

$$\text{Var}(\beta) = \sum_k M_k \quad (42)$$

with:

$$M_k = \sum_{i=1}^{n_k} H^{-1} \phi_i(\hat{\beta}) ((H^{-1}) \phi_i(\hat{\beta}))^T \quad (43)$$

Where the last sum is on all  $n_k$  indices belonging to client k. In this embodiment, the training centers compute their  $M_k$  by means of  $\zeta_s^0(\beta)$ ,  $\zeta_s^1(\beta)$ , H and with their own private data and send it to the aggregator 102 for it to reconstruct M by summing all  $M_k$ , which then allows it to compute  $\text{Var}(\hat{\beta})$  and to perform a Wald test. At block 806, process 800 receives the  $M_k$  computed using the  $\zeta_s^0(\beta)$ ,  $\zeta_s^1(\beta)$ , H as in block 804. In one embodiment, each of the training centers computes  $M_k$  using (43) and returns the corresponding  $M_k$  to process 800. Process 800

computes  $Var(\hat{\beta})$  with the gathered  $M_k$  at block 808 using (42). At block 810, process 800 performs the Wald test using the computed  $Var(\hat{\beta})$ .

**[0071]** Figure 7 is an illustration of a plot 700 of one embodiment between pooled IPTW and FedECA. In one embodiment, the error between the pooled IPTW and FedECA is small, on the order of  $10^{-5}$  for propensity scores,  $10^{-5}$  (or smaller) for hazard ratio,  $10^{-4}$  for p-values,  $10^{-5}$  for propensity scores, and  $10^{-6}$  for partial likelihood. In this example, the errors between pooled IPTW and FedECA are small.

**[0072]** Figure 9 shows one example of a data processing system 900, which may be used with one embodiment of the present invention. For example, the system 900 may be implemented by any of the training centers 104A-N, 106 and aggregator 102 as shown in Figure 1 above. Note that while Figure 9 illustrates various components of a computer system, it is not intended to represent any particular architecture or manner of interconnecting the components as such details are not germane to the present invention. It will also be appreciated that network computers and other data processing systems or other consumer electronic devices, which have fewer components or perhaps more components, may also be used with the present invention.

**[0073]** As shown in Figure 9, the computer system 900, which is a form of a data processing system, includes a bus 903 which is coupled to a microprocessor(s) 905 and a ROM (Read Only Memory) 907 and volatile RAM 909 and a non-volatile memory 911. The microprocessor 905 may include one or more CPU(s), GPU(s), a specialized processor, and/or a combination thereof. The microprocessor 905 may retrieve the instructions from the memories 907, 909, 911 and execute the instructions to perform operations described above. The bus 903 interconnects these various components together and also interconnects these components 905, 907, 909, and 911 to a display controller and display device 917 and to peripheral devices 915 such as input/output (I/O) devices which may be mice, keyboards, modems, network interfaces, printers and other devices which are well known in the art. Typically, the input/output devices 915 are coupled to the system through input/output controllers 913. The volatile RAM (Random Access Memory) 909 is typically implemented as dynamic RAM (DRAM), which requires power continually in order to refresh or maintain the data in the memory.

**[0074]** The mass storage 911 is typically a magnetic hard drive or a magnetic optical drive or an optical drive or a DVD RAM or a flash memory or other types of memory systems, which maintain data (e.g., large amounts of data) even after power is removed from the system. Typically, the mass storage 911 will also be a random-access memory although this is not required. While Figure 9 shows that the mass storage 911 is a local device coupled directly to the rest of the components in the data processing system, it will be appreciated that the present invention may utilize a non-volatile memory which is remote from the system, such as a network storage device which is coupled to the data processing system through a network interface such as a modem, an Ethernet interface or a wireless network. The bus 903 may include one or more buses connected to each other through various bridges, controllers and/or adapters as is well known in the art.

**[0075]** Portions of what was described above may be implemented with logic circuitry such as a dedicated logic circuit or with a microcontroller or other form of processing core that executes program code instructions. Thus, processes taught by the discussion above may be performed with program code such as machine-executable instructions that cause a machine that executes these instructions to perform certain functions. In this context, a “machine” may be a machine that converts intermediate form (or “abstract”) instructions into processor specific instructions (e.g., an abstract execution environment such as a “virtual machine” (e.g., a Java Virtual Machine), an interpreter, a Common Language Runtime, a high-level language virtual machine, etc.), and/or, electronic circuitry disposed on a semiconductor chip (e.g., “logic circuitry” implemented with transistors) designed to execute instructions such as a general-purpose processor and/or a special-purpose processor. Processes taught by the discussion above may also be performed by (in the alternative to a machine or in combination with a machine) electronic circuitry designed to perform the processes (or a portion thereof) without the execution of program code.

**[0076]** The present invention also relates to an apparatus for performing the operations described herein. This apparatus may be specially constructed for the required purpose, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-

ROMs, and magnetic-optical disks, read-only memories (ROMs), RAMs, EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

**[0077]** A machine-readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine-readable medium includes read only memory (“ROM”); random access memory (“RAM”); magnetic disk storage media; optical storage media; flash memory devices; etc.

**[0078]** An article of manufacture may be used to store program code. An article of manufacture that stores program code may be embodied as, but is not limited to, one or more memories (e.g., one or more flash memories, random access memories (static, dynamic or other)), optical disks, CD-ROMs, DVD ROMs, EPROMs, EEPROMs, magnetic or optical cards or other type of machine-readable media suitable for storing electronic instructions. Program code may also be downloaded from a remote computer (e.g., a server) to a requesting computer (e.g., a client) by way of data signals embodied in a propagation medium (e.g., via a communication link (e.g., a network connection)).

**[0079]** The preceding detailed descriptions are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the tools used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of operations leading to a desired result. The operations are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

**[0080]** It should be kept in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the above discussion, it is appreciated that throughout the description, discussions utilizing terms such as “receiving,”

“computing,” “exchanging,” “processing,” “configuring,” “augmenting,” “sending,” “assembling,” “monitoring,” “gathering,” “updating,” “pushing,” “training,” “testing,” “communicating,” or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

**[0081]** The processes and displays presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct a more specialized apparatus to perform the operations described. The required structure for a variety of these systems will be evident from the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein.

**[0082]** The foregoing discussion merely describes some exemplary embodiments of the present invention. One skilled in the art will readily recognize from such discussion, the accompanying drawings and the claims that various modifications can be made without departing from the spirit and scope of the invention.

CLAIMS

What is claimed is:

1. A method comprising:
  - training a first model using a plurality of training centers, wherein each training center includes a private set of training data and a compute unit and the first model estimates the probability of receiving the experimental treatment for each sample given the plurality of private sets of training data;
  - training a second model using the plurality of training centers, wherein the training includes computing weights for each sample using estimated probabilities from the first model and the second model estimates an effect of a drug on time-to-event outcomes on weighted data; and
  - testing the second model to estimate an effect of a treatment variable on a time-to-event outcome, wherein the testing uses a distributed robust variance estimation for the effect estimate.
2. The method of claim 1, wherein the training of the second model comprises:
  - training the second model using a federated optimization scheme to fit a weighted Cox model using the plurality of training centers.
3. The method of claim 2, wherein the training of the second model comprises:
  - sending a request to each of the plurality of training centers to compute local quantities on each risk set that are sent to an aggregator node to compute a global gradient and a global Hessian.
4. The method of claim 3, wherein the local quantities computed by the local centers are weighted sums of  $\zeta_{\square}^0$ ,  $\zeta_{\square}^1$ ,  $\zeta_{\square}^2$  over each risk set, weighted sums of covariates used for IPTW and the sum of weights on distinct event times.
5. The method of claim 4 wherein the testing on the second model for an effect comprises:
  - computing a robust variance estimator using the aggregated Hessian and the local quantities.

6. The method of claim 3, wherein the aggregation can be performed on a trusted execution environment or using secure aggregation methods.
7. The method of claim 3, wherein the training of the second model comprises:
  - receiving the current coefficients and the risk set of counterparts from each of the plurality of training centers; and
  - computing the aggregated gradient and the aggregated Hessian of the second model from the plurality of coefficients and the plurality of risk set of counterparts.
8. The method of claim 3, wherein the training of the second model comprises:
  - computing an updated set of coefficients for the second model using the aggregated gradient and the aggregated Hessian.
9. The method of claim 1, wherein the first model is a logistic regression model.
10. The method of claim 1, wherein the second model is a weighted Cox model.
11. The method of claim 1, wherein the training of the first model comprises:
  - sending a first set of parameters of the first model to each of the plurality of training centers, wherein each of the plurality of training centers computes a gradient and a Hessian of the first model using the first set of parameters;
  - receiving a plurality of gradients and a plurality of Hessians of the first model from the plurality of training centers;
  - computing an aggregated gradient and an aggregated Hessian of the first model from the plurality of gradients and the plurality of Hessians; and
  - computing an updated set of parameters of the first model using the aggregated gradient and the aggregated Hessian.
12. The method of claim 1, wherein the training of the first model is performed with a privacy enhancing layer by applying a distributed differential privacy gradient descent on the plurality of training centers.

13. The method of claim 10, wherein each of the plurality of training centers computes a log-likelihood for each set of parameters.
14. The method of claim 13, wherein each of the plurality of training centers computes a gradient and a Hessian of the first model using the log-likelihood of the first set of parameters.
15. The method of claim 1, wherein the testing of the second model comprises:  
computing the distributed robust variance estimation using the plurality of training centers, wherein each of the plurality of training centers computes  $M_K$  using the private set of training data corresponding to that training center.
16. The method of claim 1, wherein the testing of the second model is a Wald test using the distributed robust variance estimation.
17. A machine-readable medium having executable instructions to cause one or more processing units to perform the method of any preceding claim.

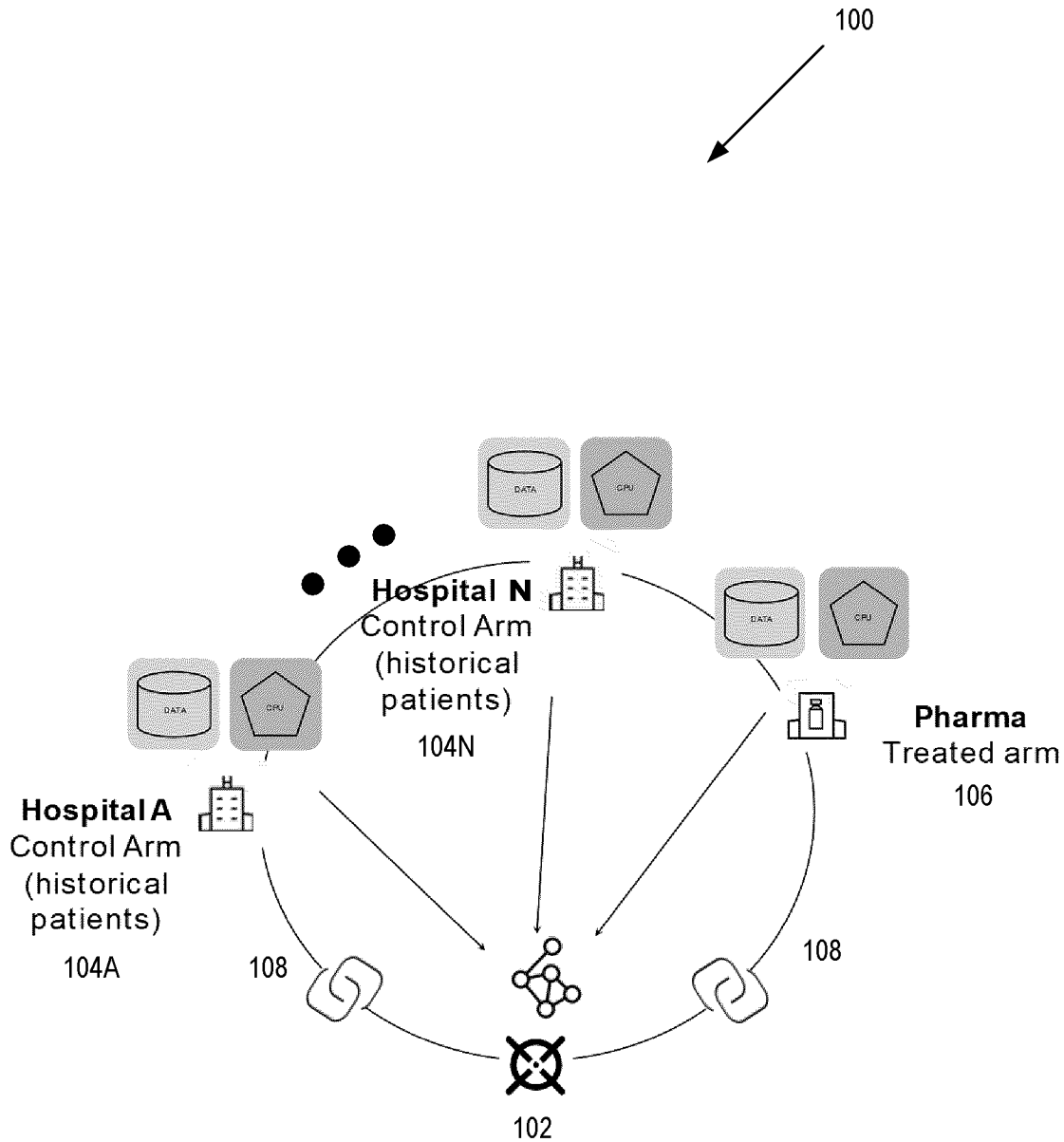


FIG. 1

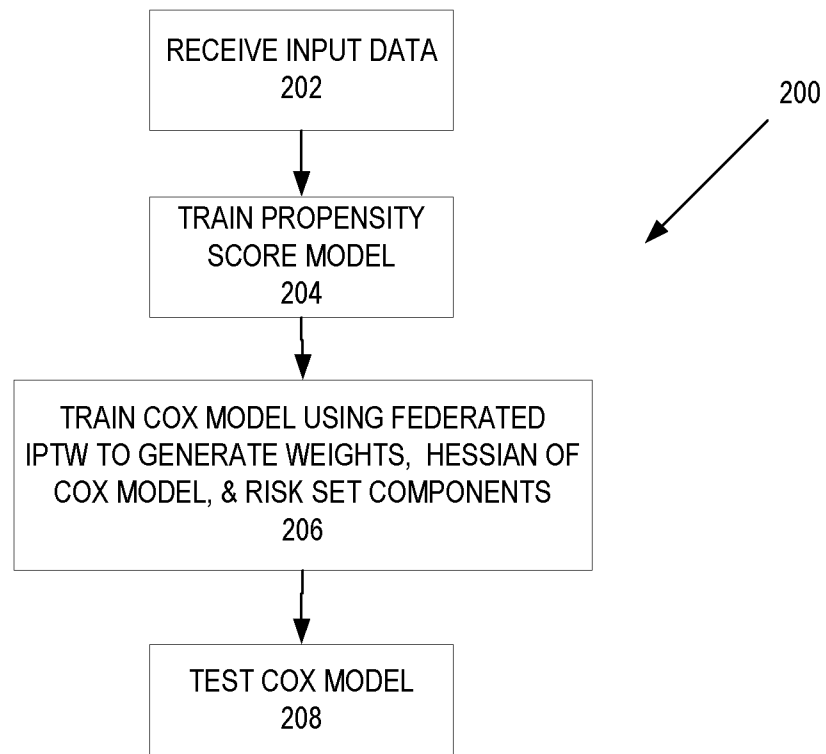


FIG. 2

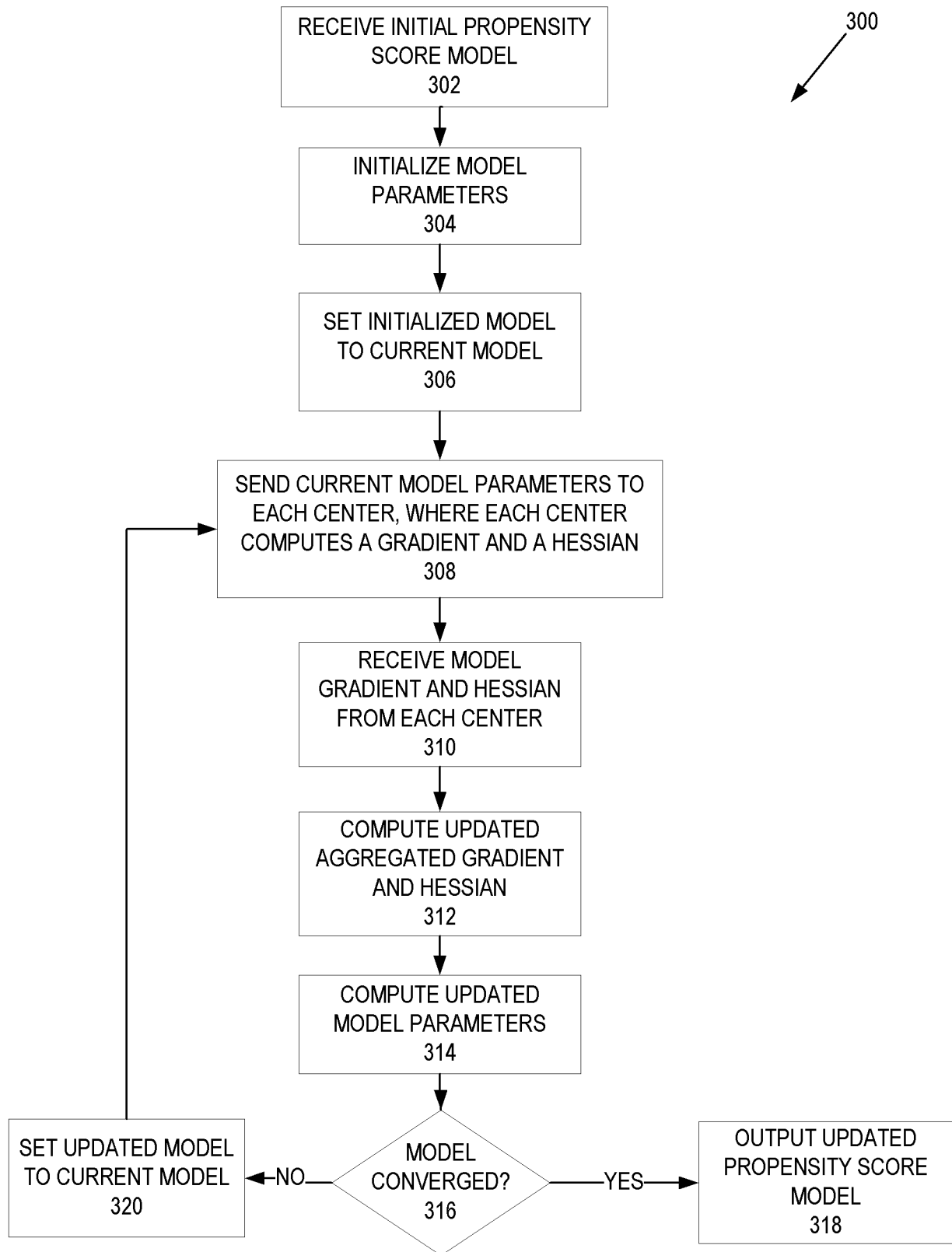


FIG. 3

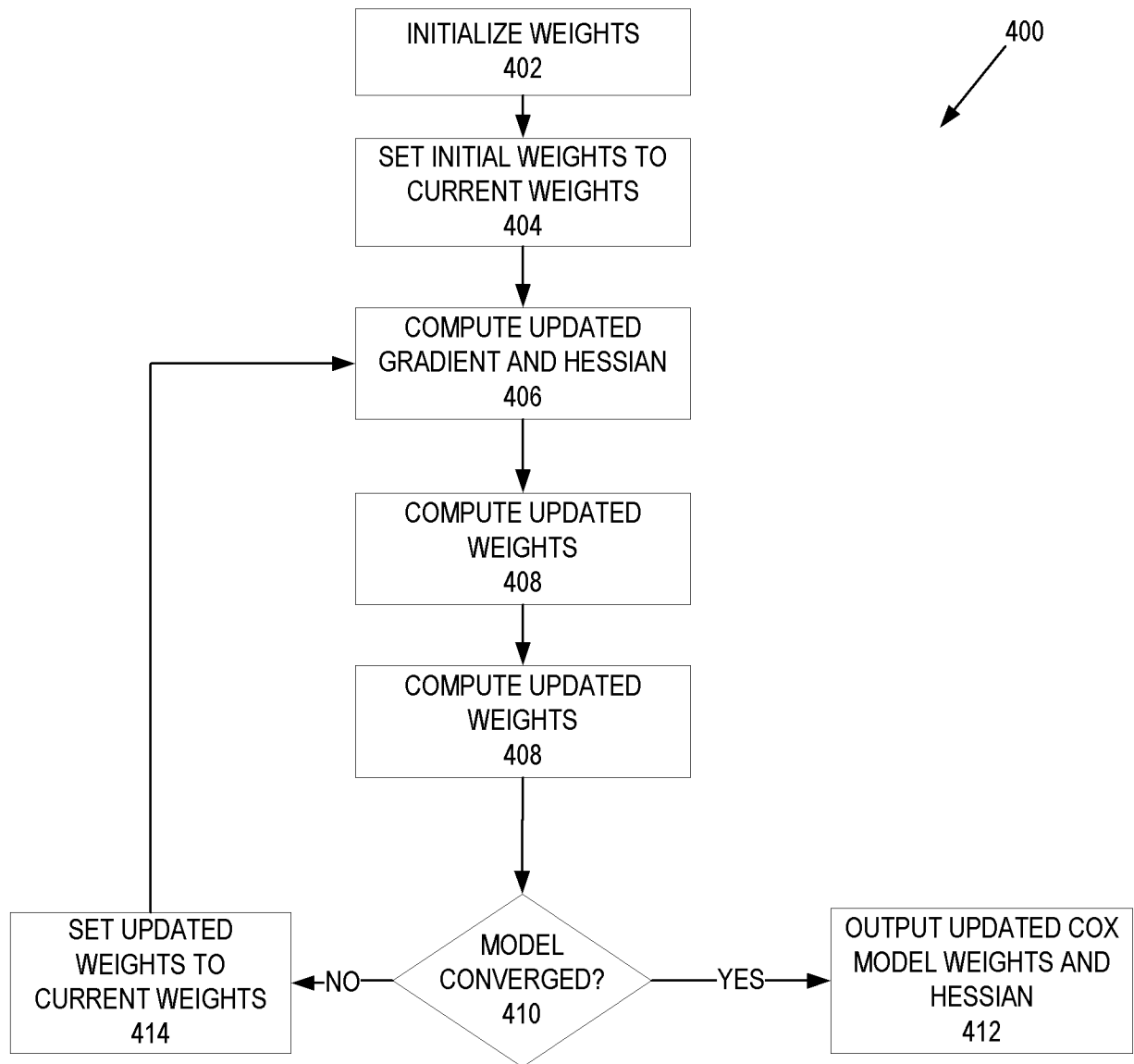


FIG. 4

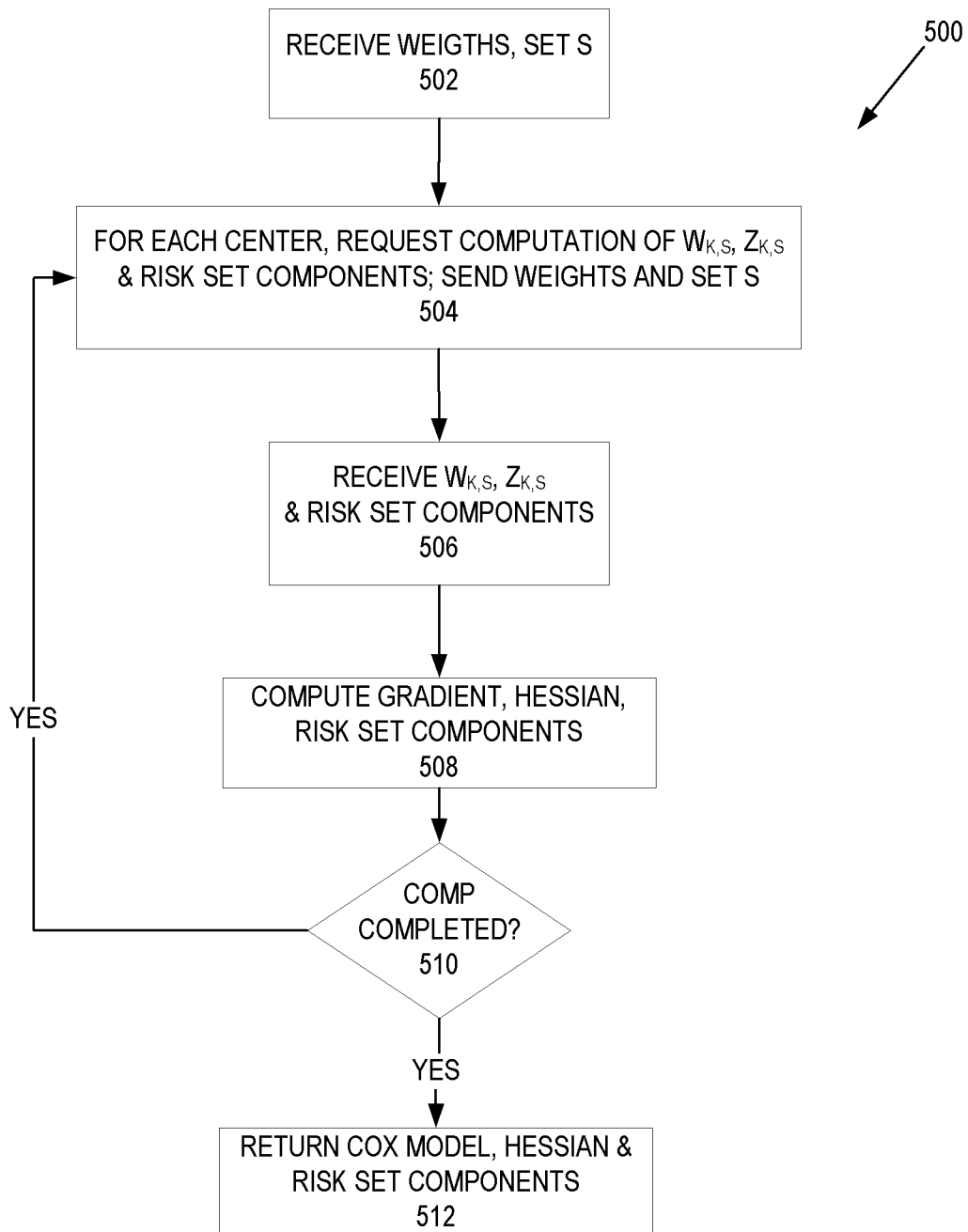


FIG. 5

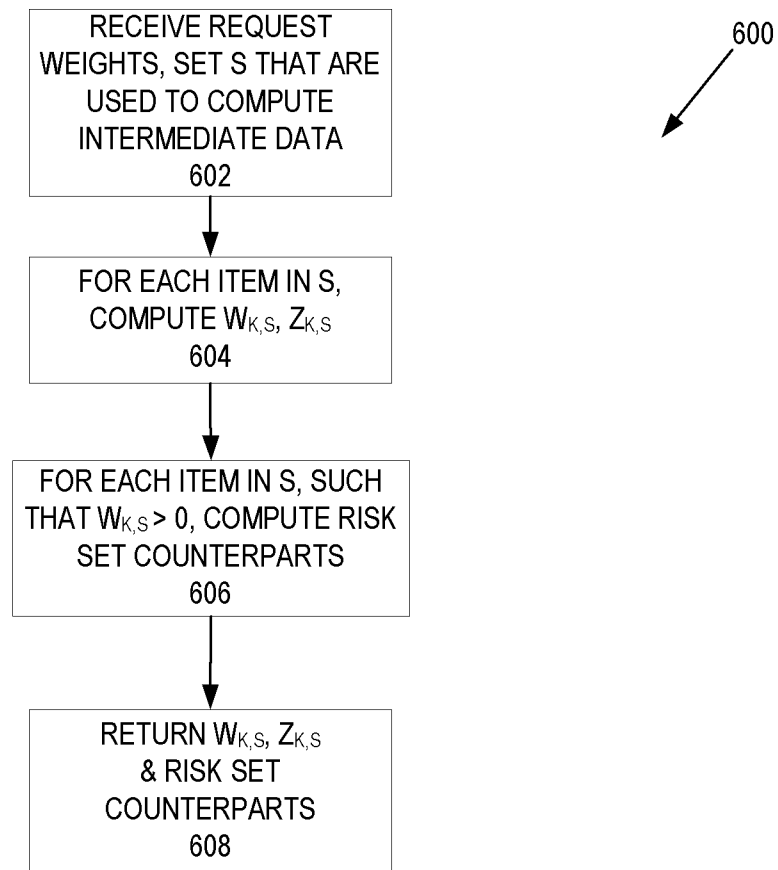


FIG. 6

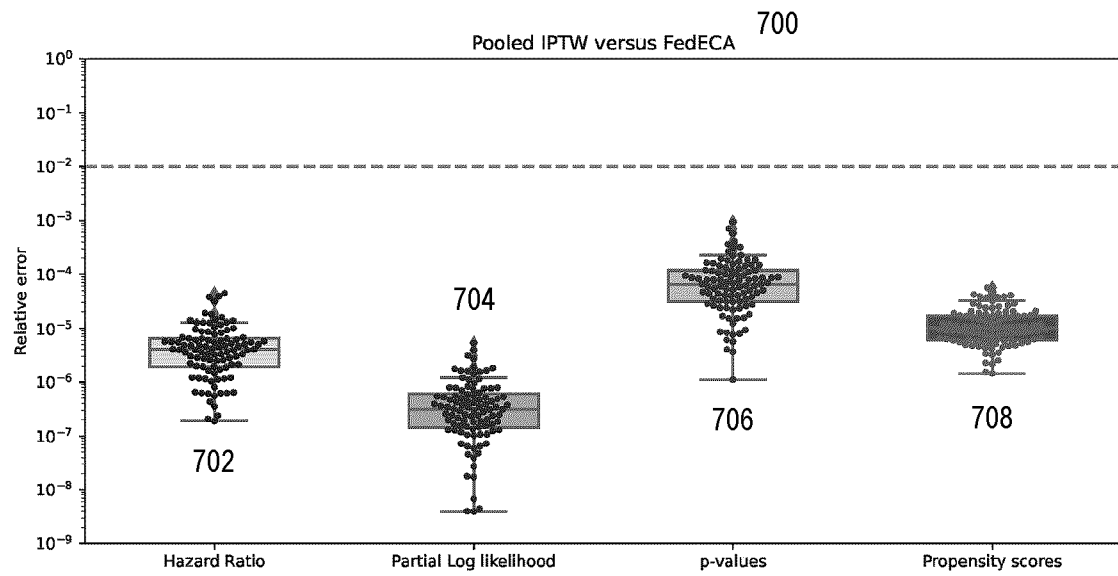


FIG. 7

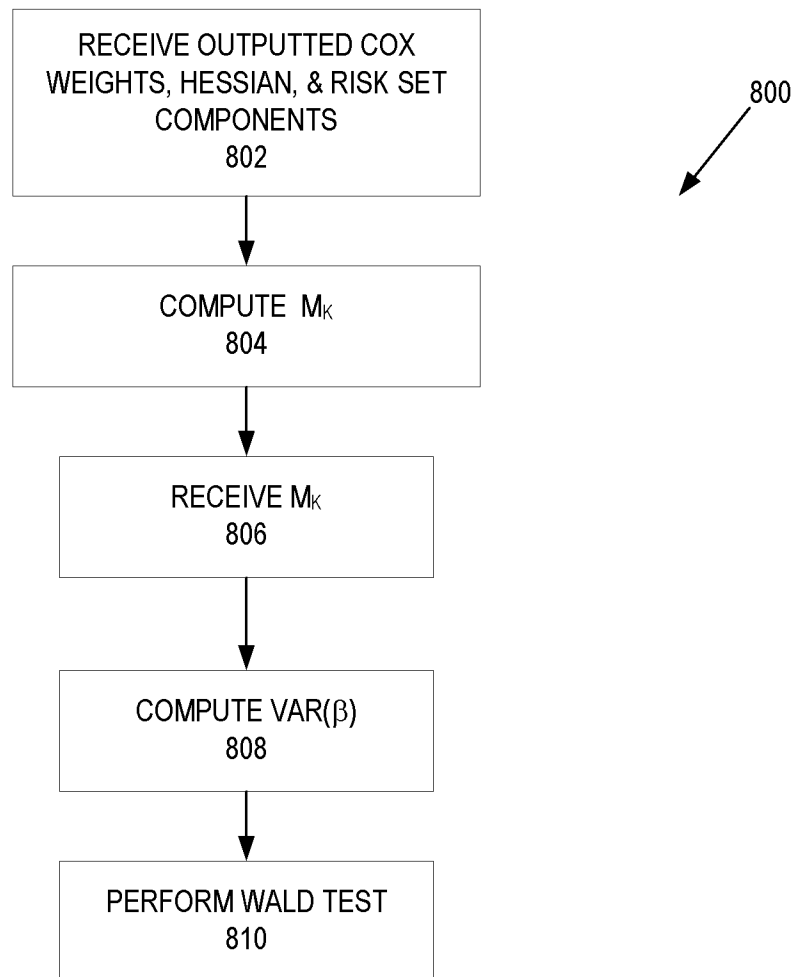
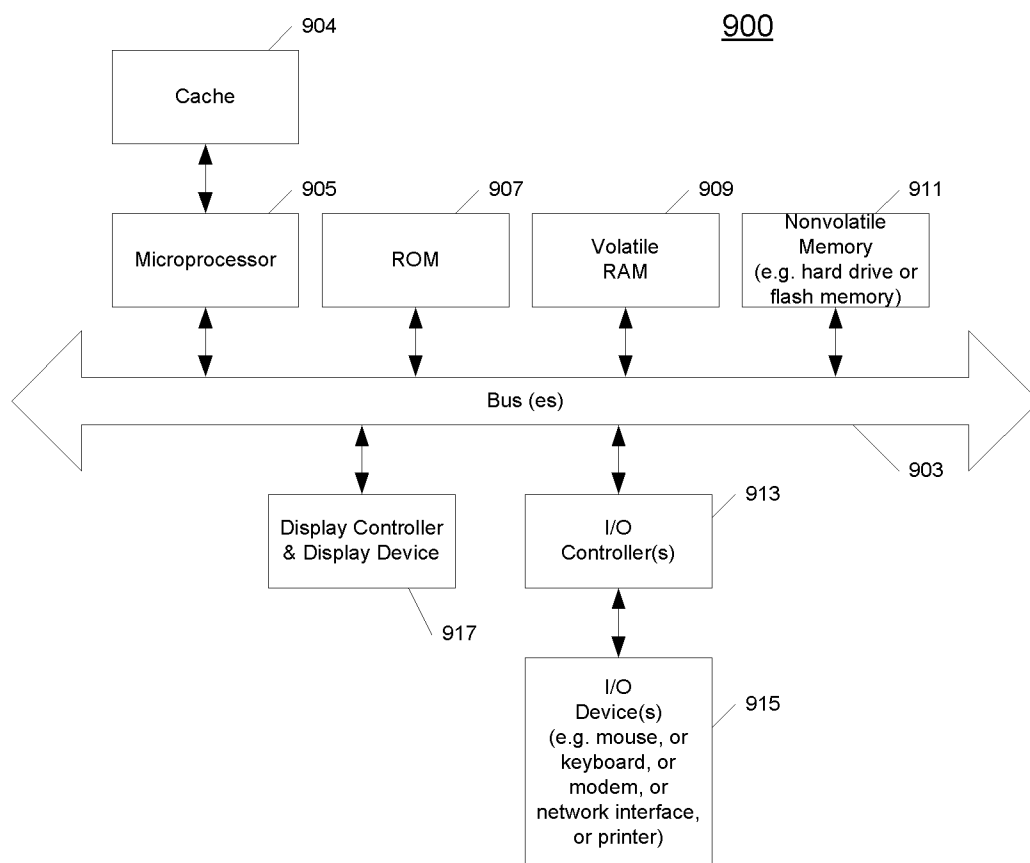


FIG. 8

**Fig. 9**

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2024/051768

**A. CLASSIFICATION OF SUBJECT MATTER**IPC: **G16H 20/00** (2024.01); **G16H 10/60** (2024.01); **G16H 50/00** (2024.01); G16H 20/10 (2024.01); G16H 50/30 (2024.01); G16H 50/70 (2024.01)CPC: **G16H20/00**; **G16H10/60**; **G16H50/00**; G16H20/10; G16H50/30; G16H50/70

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. |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| A         | US 2019/0371468 A1 (INFORM GENOMICS, INC.) 05 December 2019 (05.12.2019)<br>Paragraphs [0024, 0029, 0032-0034, 0036, 0040-0041, 0049, 0071-0072] | 1-17                  |
| A         | US 2021/0142910 A1 (TEMPUS LABS, INC.) 13 May 2021 (13.05.2021)<br>Paragraphs [0033, 0113, 0149-0151]                                            | 1-17                  |
| A         | US 2020/0411199 A1 (CANCER COMMONS) 31 December 2020 (31.12.2020)<br>Paragraphs [0194-0195]                                                      | 1-17                  |



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

17 December 2024 (17.12.2024)

Date of mailing of the international search report

14 January 2025 (14.01.2025)

Name and mailing address of the ISA/US

COMMISSIONER FOR PATENTS  
MAIL STOP PCT, ATTN: ISA/US  
P.O. Box 1450  
Alexandria, VA 22313-1450  
UNITED STATES OF AMERICA

Authorized officer

SHANE THOMAS

Facsimile No. 571-273-8300

Telephone No. PCT Helpdesk: (571) 272-4300