

UNIFIED INTELLIGENCE SYSTEM AND METHOD FOR ADAPTIVE COLLABORATIVE LEARNING

FIELD OF TECHNOLOGY

[001] Certain embodiments of the disclosure relate to Machine Learning (ML) and Artificial Intelligence (AI). More specifically, certain embodiments of the disclosure relate to a unified intelligence (UI) System and Method thereof for adaptive collaborative learning.

BACKGROUND

[002] Recent trends showing phenomenal growth of data is encouraging the adoption of new-age data analytical technologies (such as ML, AI, deep learning, simulation, and automated systems) into businesses and enterprises with each passing day. Such technologies aim at understanding and analyzing raw datasets, deriving a conclusion regarding the information they hold, facilitating data driven informed decisions, and determining how to gain value and relevant business insights from it. Data analytical methods help businesses answer essential questions and, in turn, perform better. From providing financial insights to helping to understand customer behaviour, robust data analyses can drive a business forward and help to identify and mitigate risks.

[003] In certain scenarios, existing data analytical methods may provide the users with insights into the reliability of data, but compromise on data privacy safety and legal compliance as well. Such data analytical methods depend on the collection and storage of data in a central location. The collected data is then studied and analyzed for patterns and insights to train AI and ML models to provide insights, and predictions of probable future behaviour and trends. While this approach is robust, it is dependent on owners of the data granting permission for their data to be stored and used. However, with the increasing incidents of data thefts and misuse, there are data privacy issues. In another scenario, there are geographical areas with a portion of population- that have limited or no access to financial services – the underbanked and unbanked. For example, in Singapore, about 2% of the population do not have access to financial services. In the other Association of Southeast

Asian Nations (ASEAN) countries, the ranges between 15 and 69%. This results in the lack of data available to various services, aggravating the situation where services such as financial access, are not availed to them. Therefore, there requires a necessity for developing platforms through which financial and other services can be availed without causing any risks to personal data.

[004] Further limitations and disadvantages of conventional approaches will become apparent to one skilled in the art, through comparison of such systems with some aspects of the present disclosure as set forth in the remainder of the present application with reference to the drawings.

BRIEF SUMMARY OF THE DISCLOSURE

[005] Unified intelligence (UI) System and Method for adaptive collaborative learning, substantially as shown in and/or described in connection with at least one of the figures, is disclosed as set forth more completely in the claims.

[006] These and other advantages, aspects and novel features of the present disclosure, as well as details of an illustrated embodiment thereof, will be more fully understood from the following description and drawings.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[007] FIG. 1 is a block diagram that illustrates an exemplary UI System for Adaptive Collaborative Learning (ACL), in accordance with an exemplary embodiment of the disclosure;

[008] FIG. 2A illustrates a schematic diagram of a Central Server, that may serve Federated Learning (FL) and generating the Optimal Global Model, in accordance with various embodiments of the disclosure;

[009] FIG. 2B illustrates a schematic diagram of a Computing Device from the plurality of Computing Devices, in accordance with various embodiments of the disclosure;

[010] FIG. 3 illustrates a FL sequence depicting a sequence flow between various participants/actors, as disclosed in FIGs. 1, 2A, and 2B, in accordance with an embodiment of the disclosure;

[011] FIG. 4A illustrates a flowchart that depicts a first method performed at the Central Server for ACL, in accordance with an embodiment of the disclosure; and

[012] FIG. 4B illustrates a flowchart that depicts a second method performed at a Computing Device for ACL, in accordance with an embodiment of the disclosure.

DETAILED DESCRIPTION OF THE DISCLOSURE

[013] Various embodiments of the disclosure provide the UI System and Method for Adaptive Collaborative Learning. Digital footprints are revolutionizing the financial industry in digital transformation due to the rapid proliferation of Internet technologies in everyday life, making them ubiquitous. FL, also known as collaborative learning, is evolving as a distributed ML technique that realizes distributed modelling and improves effects on AI models. FL has widespread applications covering but not limited to telecommunications, IoT, pharmaceuticals, and financial product and services. As explicitly disclosed herein, FL based assessment of individual behaviour and character reliability may provide a comprehensive set of system design for efficient and effective services, such as credit risk assessment, credit scoring, and the like.

[014] In accordance with various embodiments of the present disclosure, the UI System may be employed to solve classification problems. Classification problems belong to a family of data analytics tasks that produce models to classify an object into particular categories. The classification outcome is estimated based on the characteristics of the object, so called features. If each feature is represented by a numerical or even categorical value, for example, 'x', an object may be therefore characterized by a list of values, called feature vector: $X=\{x_1, x_2, \dots\}$. The label associated with each object may be denoted as 'y', known as a target variable, dependent variable or outcome. 'y', can be a binomial i.e. either 1 or 0. In some embodiments, the 'y' can be multinomial i.e. discrete outcome

depending on the context. The UI System may be configured to train a model from such labeled objects and predict the outcome label of objects whose outcome is not yet known.

[015] In accordance with various embodiments of the present disclosure, the proposed UI System may be implemented in a variety of applications, such as, but not limited to healthcare, insurance, autonomous vehicles, tracking manufacture, or any online services which involves exchange of data and heterogeneity.

[016] In accordance with an aspect of the present disclosure, the UI System is disclosed that comprises a decentralized plurality of Computing Devices and a Central Server. Each of the plurality of Computing Devices may be configured to receive at least a current model and a model checkpoint from an FL Portal. Each of these Computing Devices train the received current model using local data comprising alternate data from the corresponding Computing Device associated with a user. The local data may correspond to objects with outcome labels. The availability of features of the objects with the outcome labels is in accordance with a defined configuration. The local data of each of the plurality of Computing Devices is encrypted such that the personal identifiable information of each Computing Device is protected from other Computing Devices. This protects the data privacy of the user associated with the corresponding Computing Device. Each of the plurality of Computing Devices may be further configured to generate an Updated Model, which is cryptographically masked, using the trained model for FL. The Central Server comprises the FL Portal, a Persistent Storage Device and an FL Coordinator. The FL Portal may comprise application program interfaces (APIs) configured to distribute the current model to the plurality of Computing Devices in accordance with a training plan. The Persistent Storage Device may be configured to store the current model, which is retrieved by the FL Coordinator. The FL Coordinator may be configured to transmit the current model, retrieved from the Persistent Storage Device, to the FL Portal, generate a Global Model based on model aggregation using Stochastic Gradient Descent (SGD) method on the Updated Models. The Updated Models are received from the plurality of Computing Devices for aggregation, in accordance with the defined configuration for adaptive learning. The Global Model may be re-distributed amongst the plurality of Computing Devices for the next training

round until convergence of the Global Model is achieved. The FL Coordinator may be further configured to estimate, using an Optimal Global Model corresponding to a converged Global Model, an outcome label classification of a new unlabeled object characterized by corresponding features. An estimated outcome label classification of the new unlabeled object may be used to determine, by an essential service provider, new data associated with the user.

[017] In accordance with an embodiment, the Central Server comprises a plurality of features and the outcome labels in accordance with a first configuration that corresponds to the defined configuration for adaptive learning. In accordance with such embodiment, the Central Server may be configured to transmit a contribution of the features to an outcome function, and weights in combination with outcome labels to each of the plurality of Computing Devices. A weight is a learnable parameter of a model. The Central Server may be further configured to receive the gradient of the weights together with the contribution of device-related features to the outcome function from each of the plurality of Computing Devices. The contribution of device-related features to the outcome function is computed by each of the plurality of Computing Devices. The gradient of the weights may be derived by each of the plurality of Computing Devices.

[018] In accordance with an embodiment, the Central Server comprises only the outcome labels in accordance with a second configuration that corresponds to the defined configuration for adaptive learning. In accordance with such embodiment, the Central Server may be configured to transmit the outcome label to each of the plurality of Computing Devices and receive gradient of the weights together with the contribution of the device-related features to an outcome function from the plurality of Computing Devices. The contribution of device-related features to the outcome function is computed by each of the plurality of Computing Devices. The gradient of the weights may be derived by each of the plurality of Computing Devices. The gradient of the weights derived by each of the plurality of Computing Devices may be sent to the next Computing Devices together with the outcome label.

[019] In accordance with an embodiment, the Updated Model may correspond to information unrelated to the features in accordance with a privacy preserving learning process. The information may comprise an updated set of aggregated values of weights multiplied by the features of the labeled objects, the weights and corresponding gradients, and outcome labels.

[020] In accordance with an embodiment, the outcome label classification may correspond to a binary classification of the new unlabeled object in accordance with a custom-built logical regression technique that uses a sigmoid function. The Optimal Global Model may be a distributed and adaptive learning model.

[021] In accordance with an embodiment, data exchanged between the FL Portal and the FL Coordinator may comprise data corresponding to the plurality of Computing Devices, the training plan, the current model, and a response generated by each of the plurality of Computing Devices.

[022] In accordance with an embodiment, each of the plurality of Computing Devices may be further configured to fine tune the current model received from the FL Portal.

[023] In accordance with an embodiment, the FL Coordinator may be configured to read the current model and the model checkpoint from the Persistent Storage Device to transmit to the plurality of Computing Devices and receive Updated Models from the plurality of Computing Devices for aggregation using Secure Multiparty Computation (SMPC) protocol. The FL Coordinator may be further configured to store a global checkpoint in the Persistent Storage Device.

[024] In accordance with an embodiment, the alternate data may comprise at least social, financial, and device data stored in corresponding Computing Device associated with the user. The social data may correspond to individual behaviour and character reliability.

[025] FIG. 1 is a block diagram that illustrates an exemplary UI System (100) for ACL, in accordance with an exemplary embodiment of the disclosure. Referring to FIG. 1, the UI System (100) may comprise various network entities, such as a plurality of Computing Devices (102) communicatively coupled to a Central Server (104). Each of the plurality of

Computing Devices (102) may be associated with a user from a plurality of users (106). There is further shown an Optimal Global Model (110A) created by the Central Server (104) based on aggregation of Updated Models (weights), such as Updated Models (114), obtained by training received models, such as a Model (112) trained in an iterative process at the plurality of Computing Devices (102) using Local Data (108). The Optimal Global Model (110A) may be obtained when convergence on a Global Model (110) is achieved.

[026] The plurality of Computing Devices (102) comprises a first set of Computing Devices (102A) and a second set of Computing Devices (102B). The first set of Computing Devices (102A) and the second set of Computing Devices (102B) may be decentralized Computing Devices that may hold local training data samples, i.e. the Local Data (108), without any exchange of Personally Identifiable Information (PII). The operational platform for the UI System (100) may be heterogenous as various characteristics, such as storage, computational, and communication capabilities, of the plurality of Computing Devices (102) may differ from each other. This could be due to factors such as variability in hardware (i.e., central processing unit (CPU) and memory), network connectivity (i.e., 3G, 4G, 5G, Wi-Fi), and percentage of charge (i.e., battery level). Examples of the technological stack corresponding to the plurality of Computing Devices (102) may include, but not limited to, KotlinSyft (Android), SwiftSyft (iOS), Syft.js (javascript), and PySyft (Python).

[027] The first set of Computing Devices (102A) may be edge Computing Devices. These Computing Devices (102) may be used by enterprises in different capacities. For example, in service-based industries, such as the finance and e-commerce sector, these Computing Devices (102) may include, for example, mobile phones, tablets, iPads, palmtops, and laptops. In another example, for rescue initiatives, these Computing Devices (102) may include a rescue robot that may be configured to serve as the edge Computing Device, and it may be tasked with moving autonomously and collecting visual data in hard-to-reach areas. In yet another example, a firewall may serve as an edge device. In this case, the firewall determines network access and therefore serves as the entry point into that network. The first set of Computing Devices (102A) may be configured to be personalized and have network connectivity issues.

[028] The second set of Computing Devices (102B) may comprise servers, such as private enterprises and government database servers. These second set of Computing Devices (102) may be configured to have either less or no network availability issues. The private sector or enterprises are the businesses that may be owned by a private group or an individual. The database servers for different government agencies may be connected via a gateway to a secure government network and may be used for inter-agency access.

[029] The Central Server (104) may correspond to an FL server that may comprise suitable logic, circuitry, and interfaces that may be configured to manage the plurality of Computing Devices (102) to obtain the Optimal Global Model (110A). The Optimal Global Model (110A) may be obtained based on the secure aggregation of models (weights), i.e. various instances of Global Model (110), trained in an iterative process at the plurality of Computing Devices (102) using the Local Data (108). The Central Server (104) may be a cloud-based distributed service in an FL environment that may work with the FL population and be able to process rounds with the plurality of Computing Devices (102).

[030] In accordance with an embodiment, the Central Server (104) may be practised in a distributed cloud computing environment, where program modules may be located in both local and remote computer system storage media including memory storage devices and tasks are performed by remote processing devices that are linked through a communication network. As is known in the prior art, cloud computing is a model of service delivery for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, network bandwidth, servers, processing, memory, storage, applications, virtual machines, and services) that can be rapidly provisioned and released with minimal management effort or interaction with a provider of the service. The cloud computing model may include various characteristics, such as on-demand self-service, broad network access, rapid elasticity, and measured service. This model may offer "services" according to different models, such as Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), Mobile "backend" as a service (MBaaS), and Serverless computing or Function-as-a-Service (FaaS). The model may be deployed as

a private cloud, public cloud, hybrid cloud, or others (such as, community cloud, distributed cloud, multi-cloud, poly cloud, or big data cloud).

[031] Each of the plurality of users (106) may be associated with a corresponding Computing Device from the plurality of Computing Devices (102). Each user may exhibit individual behaviour and character data that may be collated for further assessment. The Optimal Global Model (110A) may correspond to a model artifact that is created based on the aggregation of a plurality of models received from the plurality of Computing Devices (102). A ML model reaches convergence when it achieves a state during training in which the loss settles to within an error range around the final value, which may be designated as the Optimal Global Model (110A). In other words, the current model converges when additional training will not improve the model. As known in the prior art, the loss corresponds to a summation of the errors made for each sample in training or validation sets. Loss is often used in the training process to find the "best" parameter values for the ML model (e.g. weights). During the training process, the goal is to minimize this value. The loss may be computed by a loss function, also known as a cost function, that considers the probabilities or uncertainty of a prediction based on how much the prediction varies from the true value. Said differently, the loss function gives a user a more nuanced view into how well the ML model is performing. Various examples of the loss functions may include, but not limited to, log loss, cross-entropy loss, mean squared error, likelihood loss, and the like.

[032] The Model (112) may correspond to a model artifact that is received by the plurality of Computing Devices (102) from the Central Server (104). The Model (112) may include weights and hyperparameters, amongst others, based on which the respective Computing Devices may be trained. The Model (112) in the first round may be referred to as an initial model. Subsequent models may be referred to as the current model, and the final model may be referred to as the Optimal Global Model (110A).

[033] Local data (108) that corresponds to the data associated with the corresponding user from the plurality of users (106), may include individual behaviour and character reliability data. In various instances, the Local Data (108) may include mobile data (e.g., call and messaging events) and browsing behaviours (e.g., websites and browsing

duration). The Local Data (108) could include social features related to how an individual conducts oneself in social networking sites, activity data features based on mobile usage and browsing behaviours. The Local Data (108) may also include visual features, including facial features and user-generated images/videos, trends in commercial transactions or social network characteristics in online platforms. It should be noted that the above are just a few examples of the Local Data (108). Other data may also be possible without any deviation from the scope of the disclosure.

[034] In an exemplary scenario, the proposed UI System (100) may be deployed to generate a revenue model, which when inferred may provide an estimated credit score of a borrower (for example, approximately 296 million of unbanked and underbanked population from South East Asia (SEA)) to a lender. In an embodiment, the proposed UI System (100) may provide the estimated credit score to both the borrower and the lender. In another embodiment, the proposed UI System (100) may provide the estimated credit score only to the lender. For such services, the UI System (100) may impose a small charge, for example, SGD1 per credit score request by the lenders. Accordingly, the annual revenue generated by the proposed UI System (100) may be approximately SGD296 millions. It may be a free service for the borrower. It should be noted that the above exemplary scenario should not be construed to be limiting. Additional exemplary scenarios in other domains, such as healthcare, mobile applications, manufacturing industries, human resource management, and the likes may also avail the benefits of the UI System (100) (generating different models according to the corresponding domains), without any deviation from the scope of the disclosure.

[035] In accordance with an embodiment, the proposed UI System (100) may be deployed to generate an improved credit score that leverages alternate data from the plurality of Computing Devices (102), such as mobile devices, as compared to traditional credit score that is based on financial capacity, bank account statement, and credit history from the bureau. Sample metadata extracted on the plurality of Computing Devices (102) to generate the improved credit score may include device and system information, call logs, contacts, messages (such as SMS), and financial apps. The proposed UI System (100) may be built

using advanced FL, privacy preserving technologies and is targeted towards unbanked and underbanked population. Examples of the alternate credit score generated by the proposed UI System (100) may be {300, 600} indicating high risk, {600, 650} indicating medium risk, and {650, 900} indicating low risk. It should be noted that the above instances are merely for exemplary purposes and should not be construed to be limiting the scope of the disclosure.

[036] FIG. 2A illustrates a schematic diagram (200A) of the Central Server (104), that may serve for FL and generating the Optimal Global Model (110A), in accordance with various embodiments of the disclosure. The Central Server (104) may include a plurality of actors that are universal primitives of concurrent computation. Such actors may be able to communicate with each other based on a message passing mechanism. Examples of such actors may include an FL Coordinator (202), a Selection Engine (204), a Secure Aggregator (206), an FL Portal (208), and a Persistent Storage Device (210). According to business needs, the FL Coordinator (202) and the FL Portal (208) included in the Central Server (104) may communicate with each other. The FL Coordinator (202) and the FL Portal (208) may also perform data interaction with the plurality of Computing Devices (102).

[037] The FL Coordinator (202) may comprise suitable logic, circuitry, and interfaces that may be configured to overall manage the FL and control the learning patterns. The FL Coordinator (202) may be a top-level actor that enables global synchronization and advancing rounds. In accordance with various embodiments, the Central Server (104) may include multiple FL Coordinators, and each one is responsible for an FL population of Computing Devices.

[038] In accordance with various embodiments, the FL Coordinator (202) may be configured to perform various tasks. In an example, the FL Coordinator (202) may train a current model, such as Model (112), using data stored at the Central Server (104). The data may include various parameters associated with the plurality of Computing Devices, client responses, training plan, and inference model. In another example, the FL Coordinator (202) may send/receive such data to/from the FL Portal (208). In yet another example, the FL Coordinator (202) may be configured to encrypt/decrypt the data to provide enhanced security through multiparty computation.

[039] An example of a technological stack corresponding to the FL Coordinator (202) may include Python Libraries. Examples of Python Libraries are: NumPy, Panada, Jupyter Notebook, and PyDP. NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Pandas is a software library written for the Python programming language for data manipulation and analysis. Pandas offers data structures and operations for manipulating numerical tables and time series. Jupyter Notebook is a web-based interactive computing platform. PyDP provides a set of ϵ -differentially private algorithms, which can be used to produce aggregate statistics over numeric datasets containing private or potentially sensitive information.

[040] The Selection Engine (204) may be configured to accept and forward connections from multiple Computing Devices. The multiple Computing Devices periodically receive information from the FL Coordinator (202) about how many Computing Devices are required for each FL population. After the Secure Aggregator (206) is spawned, the FL Coordinator (202) may instruct the Selection Engine (204) to forward a subset of the multiple Computing Devices to the Secure Aggregator (206). This allows the FL Coordinator (202) to efficiently allocate devices to FL tasks regardless of how many Computing Devices are available. The Selection Engine (204) may perform the selection of best-fitting Computing Devices in each iteration based on one or more criteria, such as resource-based, performance-based, and data-based. Such a selection may result in optimized resource usage, reduced communication latency, and enhanced Global Model quality.

[041] The Secure Aggregator (206) may be configured to manage the model exchange and aggregation security protocols to protect model security. In order to scale with the number of Computing Devices and update size, the Secure Aggregator (206) may dynamically decide to spawn one or more aggregators to which work is delegated. No information for a round is written to the Persistent Storage Device (210) until it is fully aggregated by the Secure Aggregator (206).

[042] The Secure Aggregator (206) may further handle the SMPC protocols for model exchanges and aggregations. The protocols may provide security proof to guarantee

that each Computing Device knows only its input and output. In one example, the homomorphic encryption method may encrypt the models and only allow authorized Computing Devices and the Central Server (104) to decrypt and access the models. In another example, pairwise masking and differential privacy methods may be applied to reduce the interpretability of the model by an unauthorized party. In some embodiments, an aggregation algorithm (e.g., federal averaging algorithm or secure multiparty computing algorithm) may be deployed in the Secure Aggregator (206).

[043] The FL Portal (208) may include one or more application program interfaces (APIs) that may provide a frontend for the FL Coordinator (202) and may act as an interface between the FL Coordinator (202) and the plurality of Computing Devices (102). In accordance with an embodiment, the FL Portal (208) may store a model to be trained, such as a current model, which is obtained by analyzing an FL task after the FL Coordinator (202) receives the FL task. The FL Portal (208) may be configured to store and manage various data objects associated with the use case being managed by the Central Server (104). The FL Portal (208) may be responsible for handling all the upload/download of models between the FL Coordinator (202) and the plurality of Computing Devices (102). More specifically, the FL Portal (208) may allow model parameters (such as the aggregated value of weights multiplied by the features, the weights and its gradients as well as the outcome labels) and hyperparameters (such as learning rate and mini-batch size used to control the learning process) to be uploaded/downloaded. Further, the FL Portal (208) may ensure that no feature-specific information is sent from the Computing Devices to the Central Server (104). The key technology of the FL Portal (208) may be mainly content storage and distribution technology.

[044] In accordance with various embodiments, the FL Portal (208) may be configured to perform various functionalities. In an example, the FL Portal (208) may provide APIs for the FL Coordinator (202) and the plurality of Computing Devices (102). The FL Portal (208), in conjunction with the Persistent Storage Device (210), may also store and manage data objects, handle uploading of models, such as Model (112), downloading of the

Updated Models (114), and other requests (such as model acquisition requests, selection requests, configuration requests and reporting requests).

[045] Between the FL Portal (208) and the FL Coordinator (202), multiple data exchanges occur till the model convergence is achieved. Such data exchanges include data related to the device response (for example, response ID, device-side weight updates, weighted features, device ID and training plan ID), device (for example, device ID, server-side features, and device target variables), inference model (Model ID, computed server weight, computed device-side weight, and status) and the training plan (training plan ID, computed device-side weight, required device IDs, weighted server-side features, and status).

[046] Various examples of technological stack corresponding to the FL Portal (208) may include, but are not limited to, Flask (Python), SQLAlchemy (Python), and SQLite. Flask (Python) is a micro web framework that does not require particular tools or libraries. Flask (Python) has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. SQLAlchemy is a library that facilitates the communication between Python programs and databases. Most of the time, such a library may be used as an Object Relational Mapper (ORM) tool that translates Python classes to tables on relational databases and automatically converts function calls to SQL statements. SQLite is a relational database management system that is not a client-server database engine. Rather, it is embedded into the end program.

[047] The Persistent Storage Device (210) may comprise suitable logic, circuitry, and interfaces that may be configured to store models and model checkpoints. The FL Coordinator (202) may include multiple iterations of distributed training. The FL Coordinator (202) may: 1) adjust the Global Model parameters of the current model (to be trained) in one round of distributed training, 2) update the maintained iteration times, 3) respond to the scenario when the updated iteration times reach the predefined number of times, and 4) output the Updated Model as a final training model. In accordance with certain embodiments, in scenarios when the updated iteration times do not reach the predefined number of times, the FL Coordinator (202) sends the updated model parameters serving as current Global

Model parameters to be used in the next round of distributed training to the Persistent Storage Device (210). In such embodiments, the Persistent Storage Device (210) sends the current Global Model parameters to be used in the next round of distributed training to the FL Portal (208) and obtains a storage address returned by the FL Portal (208). In response to a Computing Device passing an access verification, the Persistent Storage Device (210) may send the storage address to the Computing Device and the Computing Device responds to the received storage address and sends a model acquisition request to the FL Portal (208) based on the storage address so as to acquire current Global Model parameters to be used in the next round of distributed training.

[048] In accordance with an embodiment, the FL Portal (208) and the Persistent Storage Device (210) are respectively responsible for sending the model (for example current Global Model parameters) to the plurality of Computing Devices (102) and receiving the Updated Models (114) (for example updated model parameters) sent by the plurality of Computing Devices (102), so that the processes of sending the Model (112) to be trained to a Computing Device and receiving the Updated Models (114) trained by the Computing Devices are asynchronous and do not conflict with each other, data throughput is improved, and FL efficiency is improved.

[049] In accordance with an embodiment, the Persistent Storage Device (210) may send the current Global Model parameters to the FL Coordinator (202) in response to a model acquisition request initiated by the FL Coordinator (202). In some embodiments, the FL Coordinator (202) may interact with the Persistent Storage Device (210) in a message subscription manner or in a manner of periodically sending an acquisition request to acquire the current Global Model parameters.

[050] For every round, the FL Coordinator (202) may respond to the completion of the adjustment of the current Global Model parameters of the model to be trained, update the iteration times, send the updated iteration times and the updated current Global Model parameters to the plurality of Computing Devices (102), and wait for the plurality of Computing Devices (102) to initiate a model request for the next round of training. The FL Coordinator (202) may further determine whether the number of iterations reaches a

predefined number (corresponding to pre-specified hyper-parameter) after completing the model update and if so, may output the currently updated Global Model as a model that is finally trained.

[051] FIG. 2B illustrates a Schematic Diagram (200B) of a Computing Device from the plurality of Computing Devices (102), in accordance with various embodiments of the disclosure. The Schematic Diagram (200B) illustrates a Computing Device (102a), shown in the form of a general-purpose Computing Device that may include, but not limited to, a Network Interface (222), a User Interface (224), a Processor (226) (such as a CPU), and a Memory (228) communicatively coupled to each other via a communication bus.

[052] The Network Interface (222) may include a standard wired interface or a wireless interface (such as a Wi-Fi interface). The Network Interface (222) may be mainly to connect to the Central Server (104) and perform data communication with the Central Server (104). The User Interface (224) may include a display and an input unit such as a virtual keyboard. The Memory (228) may be a high-speed random access memory (RAM) or a non-volatile memory, such as a magnetic disk memory. The Memory (228) may also be a storage device independent of the foregoing Processor (226).

[053] The Processor (226) may be configured to manage and perform various functionalities. For example, the Processor (226) may manage the fine-tuning of the FL models, such as the Model (112), using the Local Data (108) stored in the Memory (228) of each of the plurality of Computing Devices (102). Upon request, the Processor (226) may further determine scores. In another example, the Processor (226) may manage data received and transmitted to the FL Portal (208) of the Central Server (104). In yet another example, the Processor (226) may manage the encryption Local Data (108) and an Updated Model from the Updated Models (114) by cryptographic masking to protect each participant's privacy from each other. The Processor (226) may also be configured to decrypt the encrypted data retrieved from the Memory (228).

[054] As shown in FIG. 2B, the Memory (228), as a computer storage medium, may include an Operating System (228A) and a Model Training Program (228B) based on FL. The Operating System (228A) may be a program that manages and controls the hardware

and software resources and supports the operation of the Computing Device (102a) based on FL and other software or programs. The Model Training Program (228B) may be called by the Processor (226) based on FL programs stored in the Memory (228) and execute a model parameter training method based on FL according to the embodiment of the present disclosure. Further, an Application (228C) may be executed by the Processor (226) to provide Local Data Store (228D) to an FL Runtime (228E) by implementing an application programming interface (API). An application's Local Data Store (228D) might, for example, be SQLite database. The Processor (226) may be configured to limit the total storage footprint of the Local Data Store (228D) and automatically remove old data after a predefined expiration time. The FL Runtime (228E), when provided the FL task, accesses appropriate Local Data Store (228D) to compute Global Model updates, or evaluate model quality on held out data. According to the programmatic configuration, the Application (228C) may configure the FL Runtime (228E) by providing a name of FL population and registering the corresponding Local Data Store (228D). Accordingly, a periodic runtime job may be scheduled for FL Runtime (228E). It should be noted that for training ML models on the Computing Device (102a), any negative impact on the user experience, data usage, or battery life is to be mitigated. The FL Runtime (228E) may request that the job scheduler only invokes the job when the Computing Device (102a) is idle, charging, and connected to an unmetered network, such as Wi-Fi. Once started, the FL Runtime (228E) may abort, freeing the allocated resources, if these conditions are no longer met.

[055] When invoked, the FL Runtime (228E) may contact the Central Server (104) to declare that the Computing Device (102a) is ready to run FL tasks for the given FL population. The Central Server (104) may decide whether an FL task is available for the FL population and may either return a training plan or a suggested time to check in later. If the Computing Device (102a) is selected, the FL Runtime (228E) receives the training plan, queries the Local Data Store (228D) for data requested by the training plan, and computes model updates and metrics in accordance with the training plan. Once the FL plan is executed, the FL Runtime (228E) may report computed model updates and metrics to the Central Server (104) and cleans up any temporary resources. In accordance with an

embodiment, the FL Runtime (228E) may run within the Application (228C) that configured it. In accordance with another embodiment, the FL Runtime (228E) may run within a centralized service hosted in another application.

[056] It should be understood that the Schematic Diagram (200B), as shown in FIG. 2B, does not constitute a limitation on the Computing Device (102a), and may include more or fewer components, a combination of some components, or differently arranged components than shown in the figure.

[057] In operation, multiple Computing Devices announce to Central Server (104) that for a given FL population, they are ready to run an FL task. The FL population may be specified by a globally unique name that identifies the learning problem, or an application, which is worked upon. The FL task may be a specific computation for the FL population, such as training to be performed with given parameters, or evaluation of trained models on Local Data (108), such as individual behavioural data and character reliability data of a user associated with each Computing Device from the multiple Computing Devices.

[058] From the multiple Computing Devices announcing availability to the Central Server (104), the Central Server (104) may be configured to select a subset of the plurality of Computing Devices (102) that may be invited to work on a specific FL task in round one. The Central Server (104) communicates to the selected plurality of Computing Devices (102) what computation to run with a training plan, and instructions for how to execute it. Once a round is established, the Central Server (104) transmits the current model and any other necessary state as a model checkpoint to each of the plurality of Computing Devices (102). Each participant then performs a local computation based on the current state and its local data and sends a model update in the form of a model checkpoint back to the Central Server (104). The Central Server (104) incorporates the model updates into its current model, and the process repeats until it converges to an Optimal Global Analytical Model.

[059] FIG. 3 illustrates an FL sequence depicting a sequence flow between various participants/actors, as disclosed in FIGs. 1, 2A, and 2B, in accordance with an embodiment of the disclosure. In particular, FIG. 3 illustrates a Sequence Diagram (300) for the UI System (100) in a FL fashion. The FL Sequence, as illustrated in the Sequence Diagram (300),

enables the plurality of Computing Devices (102) to advance the model of an FL population between rounds, each round including a Selection Phase (302), Configuration Phase (304), and Reporting Phase (306).

[060] With reference to FIG. 3, there is illustrated a first round of sequential flow. In accordance with the Selection Phase (302), out of multiple Computing Devices, such as five Computing Devices (120a,...,102e), three Computing Devices (102a,...102c) that meet the eligibility criteria may check-in to the Central Server (104) by opening a bidirectional stream. In accordance with an embodiment, the Central Server (104) may select the plurality of Computing Devices (102) from the multiple devices by calculating or estimating memory requirements, processing power, availability, network data transfer rate, and the like. Accordingly, the Central Server (104) may create a computational topology of an FL model for training/inference. If a Computing Device, such as two Computing Devices (102d and 102e), is not selected for participation, the Central Server (104) may respond with instructions to reconnect at a later point in time. In an exemplary scenario, the selection may be done by simple reservoir sampling. Notwithstanding, other methods may also be incorporated without any deviation from the scope of the disclosure.

[061] In accordance with the Configuration Phase (304), the Central Server (104) may be configured to read the initial model checkpoint from the Persistent Storage Device (210). The Central Server (104) may send a training plan, the initial model checkpoint, and the initial model to each of the three Computing Devices (102a,...102c). The training plan may include one part for each of the three Computing Devices (102a,...102c) and one part for the Central Server (104). The part for each of the three Computing Devices (102a,...102c) of the training plan may include information, such as selection criteria for training data in the storage device, instructions on how to batch data, how many epochs to run on a Computing Device, labels for the nodes in the graph which represent certain computations, such as loading and saving weights, and the like. The part for the Central Server (104) may include an aggregation logic. Upon receiving the training plan, the initial model checkpoint, and the initial model, the three Computing Devices (102a,...102c) may start performing on-device training, depicted by T1, T2, and T3 respectively.

[062] In accordance with the Reporting Phase (306), the Central Server (104) may wait for three Computing Devices (102a,...102c) to report updates. However, in a certain scenario, Computing Device (102b) does not report back in time due to device or network failure or does not react on configuration by the Central Server (104). In such scenario, the Central Server (104) may ignore the Computing Device (102b). As updates from the other two Computing Devices (102a and 102c) are received, the Central Server (104) may aggregate them, depicted by A1, using, for example federated averaging and instruct the Computing Devices when to reconnect. If enough Computing Devices report in time, the round will be successfully completed and the Central Server (104) may update its Global Model, otherwise the round is abandoned. It should be noted that for brevity, only initial round is discussed herein. The sequence diagrams for successive rounds are similar to the Sequence Diagram (300) of the initial round, every time updating the Global Model at the Central Server (104). The initial model may be addressed as a current model from the second round onwards, and the same sequence of steps may repeat until convergence is reached. During such round, an Optimal Global Model is obtained and may be deployed in desired problem application.

[063] FIG. 4A illustrates a Flowchart (400A) that depicts a first method performed at the Central Server (104) for adaptive collaborative learning, in accordance with an embodiment of the disclosure. The Flowchart (400A) is described in conjunction with the FIGs. 1 to 3 and 4B.

[064] At step 402 within the Flowchart (400A), the current model and the model checkpoint may be read from the Persistent Storage Device (210) to transmit to the plurality of Computing Devices (102). In accordance with an embodiment, the FL Coordinator (202) in the Central Server (104) may be configured to read the current model, for example the Model (112), and at least model checkpoint from the Persistent Storage Device (210). In accordance with an embodiment, for an initial round of iterations, the Model (112) may be addressed as an initial model to be trained. In accordance with another embodiment, for successive rounds of iterations, the Model (112) may be addressed as current model to being trained.

[065] In accordance with an embodiment, the model checkpoint may correspond to an initial model checkpoint for a first round of iterations during FL. In accordance with another embodiment, the model checkpoint may correspond to a current model checkpoint for successive rounds of iterations during FL. The model checkpoint may correspond to a serialized state of a participant (or an actor), such as the plurality of Computing Devices (102) and the Central Server (104), in a session. For example, for each Computing Device, corresponding states may contain local observations of the environment, the status of the Computing Devices including buffer states, allocated offloading bandwidth, and the like. Further, the state of the Central Server (104) may consist of the status of all Computing Devices.

[066] Prior to reading the model and the model checkpoint, multiple Computing Devices may announce to the Central Server (104) that they are ready to run an FL task for a given FL population. The FL population may be specified by a globally unique name that identifies the learning problem, or application, which is worked upon. The FL task may be a specific computation for the FL population, such as training to be performed with given hyperparameters, or evaluation of trained models on local data of a Computing Device.

[067] From the multiple Computing Devices announcing availability to the Central Server (104) during a certain time window, the Selection Engine (204) in the Central Server (104) selects a subset, i.e. the plurality of Computing Devices (102), on the specific FL task based on selection criteria, as disclosed above. Such handshaking between the Computing Devices and Central Server (104) may be referred to as a round of iterations. The plurality of Computing Devices (102) stays connected to the Central Server (104) for the duration of the round. However, for the next round, the Central Server (104) again selects another subset of Computing Devices, a new plurality of Computing Devices, or a combination of the two, on the specific FL task based on selection criteria, as disclosed above. Such continued engagement between the Computing Devices and the Central Server (104) may be referred to as a successive round of iterations.

[068] At step 404, the model, retrieved from the Persistent Storage Device (210), may be transmitted to the FL Portal (208). In accordance with an embodiment, the FL

Coordinator (202) in the Central Server (104) may be configured to transmit the model, such as the Model (112), retrieved from the Persistent Storage Device (210), to the FL Portal (208). In other words, once the round is established, the Central Server (104) may next send, to each Computing Device, the model parameters and any other necessary state as a model checkpoint. In accordance with an embodiment, the Central Server (104) may further transmit a training plan to inform the plurality of Computing Devices (102) about what computation to run.

[069] At step 406, the current model may be distributed to the plurality of Computing Devices (102) in accordance with the training plan. In accordance with an embodiment, the FL Portal (208) in the Central Server (104) may be configured to distribute the current model, such as the Model (112), to the plurality of Computing Devices (102) in accordance with the training plan.

[070] In accordance with various embodiments, data exchanged between the FL Coordinator (202) and the FL Portal (208) may comprise data corresponding to the plurality of Computing Devices (102), the training plan, the model to be trained – such as the Model (112), and a response generated by the plurality of Computing Devices (102), i.e. the Updated Models (114).

[071] In accordance with various embodiments, at the device-side, the availability of features of the objects with the outcome labels is in accordance with a defined configuration for adaptive learning. The Local Data (108) associated with the plurality of Computing Devices (102) correspond to objects with outcome labels. For example, the Local Data (108) may comprise at least alternate data from the corresponding Computing Device (i.e., social, financial, and device data stored in the corresponding Computing Device associated with the user). The social data may include individual behaviour and character reliability of a corresponding user associated with the corresponding Computing Device. The Local Data (108) may be determined based on automated analysis of device records, such as chats, messages, emoticons, voice messages, audio messages, videos, call logs, and contacts. For example, an NLP application may conveniently determine if the user is not in good terms with a local money lender or financial institution, or the user is facing certain health issues.

[072] In accordance with a first configuration that corresponds to the defined configuration for adaptive learning, the Central Server (104) comprises a plurality of features and the outcome labels, also referred to as target variables (y), dependent variables, or simply outcomes. In such an embodiment, as the first step, a set of initial weights may be computed by the Central Server (104) based purely on server-based learning. Further, the Central Server (104) may transmit a contribution of the features to an outcome function and weights, in combination with outcome labels to each of the plurality of Computing Devices (102). In an exemplary scenario, for the computation of the feature weights, the relative importance of each feature (with respect to the classification task) may be estimated, and corresponding weight may be assigned to it based on server learning. When properly weighted, an important feature may receive a larger weight than less important or irrelevant features.

[073] In accordance with a second configuration that corresponds to the defined configuration for adaptive learning, the Central Server (104) comprises only the outcome labels. In such an embodiment, the Central Server (104) may transmit the outcome label to each of the plurality of Computing Devices (102). The difference in the two approaches stems from the availability of the data. In the two configurations for adaptive learning, the input features are stored at different locations. Control passes to step 422 (in FIG. 4B) for further processing at each of the plurality of Computing Devices (102).

[074] At 408, control is received from step 428 (in FIG. 4B) due to which Updated Models (114) may be received from the plurality of Computing Devices (102) for aggregation using SMPC protocol. In accordance with an embodiment, the Secure Aggregator (206) in the Central Server (104) may be configured to receive the Updated Models (114) from the plurality of Computing Devices (102) for aggregation using SMPC protocol, known in the prior art. In some embodiments, an aggregation algorithm (e.g., federal averaging algorithm or secure multiparty computing algorithm) may be deployed in the Secure Aggregator (206).

[075] Once the Central Server (104) receives all the Updated Models (114) from all the plurality of Computing Devices (102), the Secure Aggregator (206) performs the model aggregation using SGD method based on the weights algorithms and gets the updated

Global Model. Accordingly, the current round of the FL training is completed, and the FL training continues until it reaches the maximum rounds on the Central Server (104), for example num_rounds in the server configuration file.

[076] In accordance with an embodiment, the Updated Models (114) may be received by the Secure Aggregator (206) from the plurality of Computing Devices using Secure Aggregation, which is a SMPC protocol. The SMPC protocol uses encryption to make the Updated Models (114) from the plurality of Computing Devices that are not inspected by the Central Server (104). The Secure Aggregator (206) reveals the aggregated sum once the Updated Models (114) from the plurality of Computing Devices are received in a sufficient number according to the training plan.

[077] In accordance with an embodiment, corresponding to the first configuration for adaptive learning, as the first step (as discussed in the previous step 406), the set of initial weights may be computed by the Central Server (104) based purely on server-based learning. The Central Server (104) may transmit a contribution of the features to the outcome function, and weights in combination with outcome labels to each of the plurality of Computing Devices (102). Further, the FL Coordinator (202) in the Central Server (104) may receive a gradient of the weights together with the contribution of device-related features to the outcome function from each of the plurality of Computing Devices (102). The contribution of such device-related features to the outcome function may be computed by each of the plurality of Computing Devices (102). Further, the gradient of the weights may be derived by each of the plurality of Computing Devices (102). Such collaborative process stops when convergence is achieved.

[078] In accordance with another embodiment, corresponding to the second configuration for adaptive learning, the Secure Aggregator (206) in the Central Server (104) may receive gradient of the weights together with the contribution of device-related features to the outcome function from each of the plurality of Computing Devices (102), similar to the steps in the first configuration. Additional steps may be performed in the second configuration. For example, weights may be updated at the Central Server (104).

Subsequently, such weights may be sent to the next Computing Device together with the outcome label. Such a collaborative process stops when convergence is achieved.

[079] In accordance with various embodiments of the disclosure, as described above, no sensitive and PII is communicated between the Central Server (104) and the plurality of Computing Devices (102), although the learning takes place across a server and device setting. In other words, instead of transmitting the features which are typically extracted from the plurality of Computing Devices (102) into the Central Server (104), the aggregated value of weights multiplied by the features, the weights and its gradients as well as the outcome label is communicated between the Central Server (104) and the plurality of Computing Devices (102). In this way, no feature-specific information is sent from the plurality of Computing Devices (102) to the Central Server (104). Such an approach realizes a hybrid privacy preserving mechanism in the UI System (100).

[080] At step 410, a Global Model may be generated based on model aggregation using SGD method on the Updated Models, which are received from the plurality of Computing Devices for aggregation, in accordance with the defined configuration for adaptive learning. In accordance with an embodiment, the FL Coordinator (202) may be configured to generate a Global Model (110) based on model aggregation using SGD method on the Updated Models (114), which are received from the plurality of Computing Devices (102) for aggregation, in accordance with the defined configuration for adaptive learning. The Global Model (110) may be re-distributed amongst the plurality of Computing Devices (102), selected by the Selection Engine (204) based on the selection criteria, for the next training round until convergence of the Global Model (110) is achieved. The Global Model (110) reaches convergence when it achieves a state during training in which loss [estimates?] settles to within an error range around the final value. Such a converged Global Model (110) may be designated as the Optimal Global Model (110A). In accordance with an embodiment, the SGD method may be used to aggregate logistic regression model parameters updates (referring from the Updated Models (114)) from the plurality of Computing Devices (102) and to arrive at the Optimal Global Model (110A) faster.

[081] The FL model training on the FL Coordinator (202) is summarized herein as the following self-explanatory algorithmic steps (described herein), according to which initial weights are computed at the server-side and evaluation metrics (such as loss and accuracy) are obtained (Step 0). Further, initial weight of the Computing Device is randomly initialized (Step 1). The FL Coordinator (202) sends a training plan to the FL Portal (208) (Step 2a). The FL Coordinator (202) sends target variables and the initial weight of the Computing Device to the FL Portal (208) (Step 2b). Accordingly, the contribution of the features to the outcome function and the initial server weight to the FL Portal (208) as per the (Step 2c). From the Computing Device, the FL Coordinator (202) receives the gradient of the weights together with the contribution of device-related features to the outcome function, or device-side weight updates (Step 2d). The FL Coordinator (202) will also receive the contribution of the device-related features to the outcome function and the initial Computing Device weight from the Computing Devices (Step 2e). Accordingly, the FL Coordinator (202) updates the server-side and Computing Device-side weights (Step 2f). Stopping condition is checked (Step 2g). If the condition is met, the collaboration process stops, and the convergence is achieved. The FL Coordinator (202) updates the server-side and Computing Device-side weights to the FL Portal (208). If the condition is not met, the control moves back to (Step 2a) for the next round.

Step 0: Train w^{Server} and obtain evaluation metrics.

Step 1: Randomly initialize $w^{\text{Computing Device}}$

Step 2: Start the FL training rounds. For each round:

Step 2a): Create and send a training plan to the FL Portal (208)

Step 2b): Send target variables and $w^{\text{Computing Device}}$ to the FL Portal (208)

Step 2c): Send $X^{\text{Server}}_x w^{\text{Server}}$ to the FL Portal (208)

Step 2d): Collect $(p(y|X) - y)x^k$ from the Computing Devices (x^k is the k-th device-side feature)

Step 2e): Collect $X^{\text{Computing Device}}_x w^{\text{Computing Device}}$

Step 2f): Update w^{Server} and $w^{\text{Computing Device}}$

Step 2g): Check stopping condition. Repeat step 2a.

Step 3: Send latest w^{Server} and $w^{\text{Computing Device}}$ to the FL Portal (208).

[082] At step 412, using the Optimal Global Model (110A) corresponding to converged Global Model (110), an outcome label classification may be estimated of a new unlabeled object characterized by corresponding features. In accordance with an embodiment, using the Optimal Global Model (110A) corresponding to converged Global Model (110), the FL Coordinator (202) may be configured to estimate outcome label classification of a new unlabeled object characterized by corresponding features. In accordance with an embodiment, an estimated outcome label classification of the new unlabeled object is used to determine, by a financial service, new financial data associated with the user. In another use case, using logistic regression, individual behavioural and character reliability may be accessed to classify potential borrowers into different categories, from the lowest predicted credit risk to the highest predicted credit risk.

[083] In accordance with an embodiment, an estimated outcome label classification of the new unlabeled object is used to determine, by an essential service provider, new data associated with the user. The essential service provider is one of a financial lender, a financial institution, a mobile service provider, a healthcare provider, a manufacturing unit, an autonomous vehicle system, or the like. For example, for a new loan application applied by a borrower, an outcome label classification may be 'Yes' estimated by the Central Server (104). So, for a financial lender, the borrower is eligible for the loan, which in turn financial data of the borrower for future reference.

[084] FIG. 4B illustrates a Flowchart (400B) that depicts a second method performed at a Computing Device for adaptive collaborative learning, in accordance with an embodiment of the disclosure. The Flowchart (400B) is described in conjunction with the FIGs. 1 to 4A illustrated and described in the disclosure.

[085] At step 422, control is received from step 406 (in FIG. 4A) and at least a current Model, such as the Model (112), and a model checkpoint may be received from the FL Portal

(208). In accordance with an embodiment, each of the plurality of Computing Devices (102) may be configured to receive at least the current Model, such as the Model (112) and the model checkpoint from the FL Portal (208) in response to a model acquisition request. Along with the current model parameters, hyperparameters, and a training plan may also be received by the plurality of Computing Devices (102).

[086] Prior to receiving the current model parameters corresponding and the model checkpoint, multiple Computing Devices that meet an eligibility criteria, may check-in to the Central Server (104) by opening a bidirectional stream. Multiple Computing Devices may announce to Central Server (104) that they are ready to run an FL task for a given FL population. The FL population may be specified by a globally unique name that identifies the learning problem, or application, which is worked upon.

[087] The plurality of Computing Devices (102) may be selected by the Central Server (104) based on certain criteria, such as an optimal number of participating devices. In case a Computing Device is not selected for participation, the Central Server (104) may respond with instructions to reconnect at a later point in time. In an exemplary scenario, the selection may be done by simple reservoir sampling. However, the disclosure may not be so limited and other suitable methods may be deployed to perform the selection of a set of Computing Devices from multiple Computing Devices.

[088] The plurality of Computing Devices (102) may be informed about what computation to run according to the training plan comprising instructions for how to execute it. Once a round is established, the plurality of Computing Devices (102) may receive the current model parameters corresponding and the model checkpoint.

[089] At step 424, the received current Model, such as the Model (112), may be trained using the Local Data (108) comprising alternate data from the corresponding Computing Device. In accordance with an embodiment, the Processor (226) in conjunction with the Memory (228), may be configured to train the received current Model, such as the Model (112), using the Local Data (108) comprising alternate data from the corresponding Computing Device. The alternate data may comprise of social, financial, and device data stored in the corresponding Computing Device associated with the user. For example, the

social data may include individual behaviour and character reliability of the user, financial data may include data from financial apps installed in the corresponding Computing Device, and the device data may include data from contacts, messages, call records, and overall device and system data.

[090] Each Computing Device may perform a local computation based on the global state retrieved from the model checkpoint and the Local Data (108). Examples of Local Data (108) may include, but not limited to, device and system information, call logs, contacts, SMS, and financial apps already installed for the UI System (100) corresponding to a credit risk assessment application. In accordance with an embodiment, the Local Data (108) may correspond to objects with outcome labels for the given application.

[091] In accordance with an embodiment, the Local Data (108) of each of the plurality of Computing Devices (102) may be encrypted such that the privacy of personal data in each Computing Device is protected from other Computing Devices. In accordance with an embodiment, each of the plurality of Computing Devices (102) may be further configured to fine tune the current Model, such as the Model (112), received from the Central Server (104).

[092] At step 426, an Updated Model, which is cryptographically masked, may be generated using the trained local model for FL. In accordance with an embodiment, the Processor (226) in conjunction with the Memory (228), may be configured to generate the Updated Model from the Updated Models (114), which is cryptographically masked, using the trained local Model (112) for FL. The Updated Model corresponds to information unrelated to the features in accordance with a privacy preserving learning process. The information may comprise the aggregated value of weights multiplied by the features of the labeled objects, the weights and corresponding gradients, and the outcome labels. Each Computing Device may be further configured to send an update back to the Central Server (104). The Updated Model may be transmitted to the Central Server (104) for further aggregation as described above in FIG. 4A.

[093] In accordance with an embodiment, the Updated Models (114) so generated may be transmitted to the Central Server (104) using Secure Aggregation, which is a SMPC protocol. The SMPC protocol uses encryption such that the Updated Models (114) from the

plurality of Computing Devices (102) cannot be inspected by the Central Server (104). Once the Updated Models (114) from the plurality of Computing Devices are received in a sufficient number according to the training plan, only the aggregation is disclosed.

[094] The FL model training on a Computing Device is summarized herein as following self-explanatory algorithmic steps, according to which the Computing Device decides to participate based on the received training plan (Step 0). Next, the outcome label (or target variable) is downloaded (Step 1). Following this, the contribution of the server-related features to the outcome function and server weight are also downloaded (Step 2). The computed weight of the Computing Device is then downloaded (Step 3). Further, device-related features are computed (Step 4). Accordingly, the contribution of the Computing Device-related features to the outcome function and Computing Device server weight is computed (Step 5). Based on the above, Computing Device-side weight update is computed and uploaded to the FL Portal (208) (Step 7).

Step 0: Check open training plan to confirm participation. If yes, control moves to step 1.

Step 1: Download $y^{\text{Computing Device}}$

Step 2: Download $X^{\text{Server}} \times w^{\text{Server}}$

Step 3: Download $w^{\text{Computing Device}}$

Step 4: Compute $X^{\text{Computing Device}}$ from Computing Device

Step 5: Compute and upload $X^{\text{Computing Device}}, w^{\text{Computing Device}}$

Step 6: Compute $P(y^{\text{Computing Device}} | X^{\text{Server}}, X^{\text{Computing Device}})$

Step 7: Compute and upload $(P(y^{\text{Computing Device}} | X^{\text{server}}, X^{\text{Computing Device}}) - y) \times k$.

[095] At 428, the Updated Models (114) may be transmitted to the Central Server (104). In accordance with an embodiment, the plurality of Computing Devices (102) may be configured to transmit the Updated Models (114) to the FL Portal (208) in the Central Server (104). In accordance with an embodiment, each of the plurality of Computing Devices (102)

may encrypt using the SMPC protocol for enhanced security as no information about the private data held by the each of the plurality of Computing Devices (102) can be inferred from the Updated Models (114) sent during the execution of the SMPC protocol. Further, any proper subset of adversarial colluding Computing Devices willing to share information or deviate from the instructions during the protocol execution should not be able to force honest Computing Devices to output an incorrect result. Control passes back to step 408 (in FIG. 4A).

[096] The proposed UI System and Method are highly advantageous and technologically significant over conventional and existing systems as the UI System (100) provides custom-built adaptive learning approaches using FL techniques. The custom-built logistic regression model is developed from the ground up that splits the learning into two parts: the server side and the edge device side. Traditional logistic regression methods require the dependent (binary) and independent data variables to be in a single location for the function to work. Through a custom-built logistic regression method, the UI System (100) achieves distributed and adaptive learning. Further, the UI System (100) provides the custom-built SGD method to aggregate the logistic regression model parameters updates from the edge devices and to arrive at the optimal model faster. Furthermore, multi-party computing encryption techniques deployed in the UI System (100) ensure the highest possible security on top of the FL/training approach. In particular, the proposed UI System and Method are driven by the recognition of the needs of marginalised (i.e. unbanked and underbanked) populations in the area of accessibility to financial services. However, the invention can also be applied to other essential services (where the assessment of individual behaviour and character reliability is required) at a reasonable cost while preserving personal data privacy as required by law.

[097] In accordance with various embodiments of the present disclosure, the proposed UI System may address a couple of challenges associated with such financial services and other essential services. Firstly, the marginalised population (i.e., straddling the poverty line and do not have access to financial services) do not have sufficient credit history to provide the formal and regulated financial service providers to qualify for a loan and at an

acceptable cost. Thus, such marginalised population falls prey to informal and unregulated financial service providers – the illegal money lenders. This ultimately contributes to a vicious circle of significant social issues. The proposed UI System and Method utilises credible behaviour and character reliability information of potential clients to predict reliable financial attributes, such as risk assessment and credit score, of a user to regulated financial service provider. This may result in a number of benefits for all parties. Examples of such benefits may include, but not limited to, cost effective borrowing rates, more efficient lending processes, more conducive banking environment, and improved accessibility to financial products and services. These are likely to result in improved and empowered lives of the marginalised populations.

[098] Further, secure aggregation, an SMPC protocol implemented in the UI System (100) aims to ensure input privacy and correctness. Input privacy implies that no private data and personal identifiable information held by the participants/actors can be inferred from the messages sent during the execution of the protocol. The only information that can be inferred about the private data is whatever could be inferred from seeing the output of the function alone. Further, correctness implies that any proper subset of adversarial colluding parties willing to share information or deviate from the instructions during the protocol execution would not be able to force honest parties to output an incorrect result. Either the honest parties are guaranteed to compute the correct output, or they abort if they find an error.

[099] Secondly, traditional analytical methods depend on the collection and storage of data in a central location, which is analysed for patterns and insights to train AI and ML models to provide foresights, and predictions of probable future behaviour and trends. While this approach is robust, it is dependent on owners of the data granting permission for their data to be stored and used. However, there are data privacy issues with the increasing incidents of data thefts and misuse. With the marginalised population, such data is quite limited. However, the proliferation of mobile devices, such as mobile phones, and use of social media are significantly high in most of such countries. Such data may be sufficient and leveraged by the proposed UI System and Method to predict with a high degree of accuracy, the behaviour of the mobile device user. Accordingly, sufficient data becomes available to

train AI and ML based analytical models to provide foresights, as opposed to just insights, of user reliability and behaviour. Further, a decentralized approach to behavioural analytics may be provided with the models residing in the users' mobile devices. Personal data will not be collected and stored in a remote service but remains on the users' mobile devices. Only the results of the analyses from the individual devices are collected anonymously to train the central model which resides in a secure remote location, such as the Central Server (104). Therefore, with more accurate foresight of borrowers' behaviour, regulated financial service providers would be more willing to offer loans and other services at a more reasonable price. Such a credible assistance provided by the proposed UI System and Method results in the empowerment of the marginalised population through being able to access reasonably priced financial services, thereby leading to a more inclusive society, contributing to the social and economic growth of the country.

[0100] In accordance with various embodiments of the present disclosure, the UI System (100) may be advantageous and implemented in a variety of applications, such as, healthcare, insurance, autonomous vehicles, or manufacturing tracking.

[0101] In the case of finance services, financial service providers utilise estimated data from the Optimal Global Model (110A) generated by the UI System (100) using FL approach to provide better services at lower costs in the areas of customer acquisition, credit underwriting and customer servicing. The UI System (100) may predict the default probability of a loan application (for example default ($y=1$) or non-defaults ($y=0$)) before loan disbursement for credit risk assessment based on social, financial, and device-related data of the user. Each loan application is associated with a variety of input features, such as, demographic information (age, profession, and the like), device and system information associated with the user, call records, contacts, messages, installed financial apps, employment status, and on-time payment of a utility bill. The UI System (100) uses a logistic regression model to calculate credit risk assessment, where the cumulative weighted contribution of features estimates the probability of default. Each feature x_i is associated with a coefficient or weight w_i as well as the weighting contribution of x_i to the final risk

assessment. The default risk probability $P\left(y = \frac{1}{x}\right)$ of the loan application associated with a feature vector X is therefore expressed as:

$$P\left(y = \frac{1}{x}\right) = \frac{1}{1 + \exp(X^T W)},$$

where $X^T W = \sum_{i=1}^n x_i w_i$, and

$W = \{w_1, \dots, w_n\}$ represents the model.

[0102] In case of healthcare services, healthcare and health insurance industry utilise estimated data from the Optimal Global Model (110A) generated by the UI System (100) using FL approach to provide better data diversity by gathering data from various locations (for example, hospitals, electronic health record databases) to diagnose rare diseases, while protecting sensitive data in the original source. In addition to diagnosis of rare diseases, it can also predict onsets of illness or endemics.

[0103] In case of autonomous vehicle services, service providers may utilise estimated data from the Optimal Global Model (110A) generated by the UI System (100) using FL approach to provide better and safer self-driving car experience with real-time data and predictions. The UI System (100) may provide real-time data analysis, predictions, and training thereof to self-driving machines (cars) such that the self-driven cars can be trained to respond to new situations. For example, trained and analysed real-time information about the traffic and roads may be transmitted to the autonomous vehicles. In some embodiments, analysed data of real-time decision making is transmitted to such vehicles through the embodiments of the present disclosure. The self-driven cars hence can continually learn and be trained through the UI System (100) in a FL environment.

[0104] In case of manufacturing services, manufacturing companies may utilise estimated data from the Optimal Global Model (110A) generated by the UI System (100) using FL approach to develop predictive maintenance models for equipment. The UI System (100) may handle the barriers of predictive maintenance by using local datasets. Examples of such barriers may be like customers do not want to share their personal data or export data problems from different countries/sites.

[0105] In case of mobile applications, companies may utilise estimated data from the Optimal Global Model (110A) generated by the UI System (100) using FL approach based on the data pool of smartphones without leaking personal data, such as for next-word prediction, face detection, voice recognition, and the like. For example, Google uses FL approaches to improve on-device ML models like “Hey Google” in Google Assistant which allows users to issue voice commands.

[0106] In a nutshell, the proposed UI System (100) and associated method along with various embodiments thereof may have various advantages associated therewith. The proposed UI System (100) and Method is based on advanced FL learning, distributed data management based on the availability of data, privacy preserving technologies, and enhanced security through Multi-Party Configuration. The first phase is targeted towards the unbanked and underbanked populations. The proposed UI System (100) and Method leverages alternate data from the Computing Devices associated with the users to generate alternate credit scores.

[0107] As utilised herein the terms software and/or firmware (“code”), platform, environment which may configure the hardware, be executed by the hardware, and/or otherwise be associated with the hardware. As utilised herein, “and/or” means any one or more of the items in the list joined by “and/or”. As an example, “x and/or y” means any element of the three-element set {(x), (y), (x, y)}. As another example, “x, y, and/or z” means any element of the seven-element set {(x), (y), (z), (x, y), (x, z), (y, z), (x, y, z)}. As utilised herein, the term “exemplary” means serving as a non-limiting example, instance, or illustration. As utilised herein, the terms “e.g.,” and “for example” set off lists of one or more non-limiting examples, instances, or illustrations.

[0108] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of embodiments of the disclosure. As used herein, the singular forms “a”, “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms “comprises”, “comprising”, “includes” and/or “including”, when used herein, specify the presence of stated features, integers, steps, operations, elements, and/or components,

but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0109] Further, many embodiments are described in terms of sequences of actions to be performed by, for example, elements of a Computing Device. It will be recognised that various actions described herein can be performed by specific circuits (e.g., application-specific integrated circuits (ASICs)), by program instructions being executed by one or more processors, or by a combination of both. Additionally, this sequence of actions described herein can be considered to be embodied entirely within any non-transitory form of a computer-readable storage medium having stored therein a corresponding set of computer instructions that upon execution would cause an associated processor to perform the functionality described herein. Thus, the various aspects of the disclosure may be embodied in a number of different forms, all of which have been contemplated to be within the scope of the claimed subject matter. In addition, for each of the embodiments described herein, the corresponding form of any such embodiments may be described herein as, for example, “logic configured to” perform the described action.

[0110] Another embodiment of the disclosure may provide a non-transitory machine and/or computer-readable storage and/or media, having stored thereon, a machine code and/or a computer program having at least one code section executable by a machine and/or a computer, thereby causing the machine and/or computer to perform the steps as described herein for UI System and Method thereof for adaptive collaborative learning.

[0111] The present disclosure may also be embedded in a computer program product, which comprises all the features enabling the implementation of the methods described herein, and which when loaded in a computer system is able to carry out these methods. Computer program in the present context means any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after either or both of the following: a) conversion to another language, code or notation; b) reproduction in a different material form.

[0112] Further, those of skill in the art will appreciate that the various illustrative logical blocks, modules, circuits, algorithms, and/or steps described in connection with the embodiments disclosed herein may be implemented as electronic hardware, computer software, firmware, or combinations thereof. To clearly illustrate this interchangeability of hardware and software, various illustrative components, blocks, modules, circuits, and steps have been described above generally in terms of their functionality. Whether such functionality is implemented as hardware or software depends upon the particular application and design constraints imposed on the overall system. Skilled artisans may implement the described functionality in varying ways for each particular application, but such implementation decisions should not be interpreted as causing a departure from the scope of the present disclosure.

[0113] The methods, sequences and/or algorithms described in connection with the embodiments disclosed herein may be embodied directly in firmware, hardware, in a software module executed by a processor, or in a combination thereof. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the prior art. An exemplary storage medium is coupled to the processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor.

[0114] While the present disclosure has been described with reference to certain embodiments, it will be noted understood by, for example, those skilled in the art that various changes and modifications could be made and equivalents may be substituted without departing from the scope of the present disclosure as defined, for example, in the appended claims. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the present disclosure without departing from its scope. The functions, steps and/or actions of the method claims in accordance with the embodiments of the disclosure described herein need not be performed in any particular order. Furthermore, although elements of the disclosure may be described or claimed in the singular, the plural is contemplated unless limitation to the singular is explicitly stated. Therefore, it is intended

that the present disclosure is not limited to the particular embodiment disclosed, but that the present disclosure will include all embodiments falling within the scope of the appended claims.

CLAIMS

What is claimed is:

1. A Unified Intelligence (UI) System, comprising:

a decentralised plurality of Computing Devices, each Computing Device configured to:

receive at least a current model and a model checkpoint from a Federated Learning FL portal;

train the received current model using local data comprising alternate data from corresponding Computing Device associated with a user,

wherein the local data correspond to objects with outcome labels,

wherein availability of features of the objects with the outcome labels is in accordance with a defined configuration, and

wherein the local data of each of the plurality of Computing Devices is encrypted such that personal data privacy of each Computing Device is protected from other Computing Devices; and

generate an updated model, which is cryptographically masked, using the trained model for FL; and

a Central Server comprising the FL Portal, a Persistent Storage Device and an FL Coordinator,

wherein the FL Portal comprising application program interfaces (APIs) configured to distribute the current model to the plurality of Computing Devices in accordance with a training plan;

wherein the Persistent Storage Device is configured to store the current model, which is retrieved by the FL Coordinator, and

wherein the FL Coordinator is configured to:

transmit the current model, retrieved from the Persistent Storage Device, to the FL Portal;

generate a Global Model based on model aggregation using Stochastic Gradient Descent (SGD) method on the Updated Models, which are received from the plurality of Computing Devices for aggregation, in accordance with the defined configuration for adaptive learning,

wherein the Global Model is re-distributed amongst the plurality of Computing Devices for next training round until convergence of the Global Model is achieved; and

estimate, using an Optimal Global Model corresponding to the converged Global Model, an outcome label classification of a new unlabeled object characterized by corresponding features,

wherein the estimated outcome label classification of the new unlabeled object is used to determine, by an essential service provider, new data associated with the user.

2. The UI System according to claim 1, wherein, in accordance with a first configuration that corresponds to the defined configuration for adaptive learning, the Central Server comprises a plurality of features and the outcome labels.

3. The UI System according to claim 2, wherein the Central Server is further configured to:

transmit a contribution of the features to an outcome function, and weights in combination with the outcome labels to each of the plurality of Computing Devices,

wherein a weight is a learnable parameter of a model; and

receive gradient of the weights together with contribution of device-related features to the outcome function from each of the plurality of Computing Devices,

wherein the contribution of device-related features to the outcome function is computed by each of the plurality of Computing Devices, and

wherein the gradient of the weights are derived by each of the plurality of Computing Devices.

4. The UI System according to claim 1, wherein, in accordance with a second configuration that corresponds to the defined configuration for adaptive learning, the Central Server comprises only the outcome labels.
5. The UI System according to claim 4, wherein the Central Server is further configured to:
 - transmit the outcome labels to each of the plurality of Computing Devices; and
 - receive gradient of the weights together with a contribution of device-related features to an outcome function from the plurality of Computing Devices,
 - wherein the contribution of device-related features to the outcome function is computed by each of the plurality of Computing Devices,
 - wherein the gradient of the weights are derived by each of the plurality of Computing Devices, and
 - wherein the gradient of the weights derived by each of the plurality of Computing Devices are sent to next Computing Devices together with the outcome labels.
6. The UI System according to claim 3, wherein, the Updated Model corresponds to information unrelated to the features in accordance with a privacy preserving learning process, and
 - wherein the information comprises updated set of aggregated value of weights multiplied by the features of labeled objects, the weights and corresponding gradients, and outcome labels.
7. The UI System according to claim 1, wherein the outcome label classification corresponds to a binary classification of the new unlabeled object in accordance with a custom-built logistic regression technique that uses a sigmoid function,
 - wherein the Optimal Global Model is a distributed and adaptive learning model.

8. The UI System according to claim 1, wherein data exchanged between the FL Portal and the FL Coordinator comprises data corresponding to the plurality of Computing Devices, the training plan, the current model, and a response generated by each of the plurality of Computing Devices.
9. The UI System according to claim 1, wherein each of the plurality of Computing Devices is further configured to fine tune the current model received from the FL Portal.
10. The UI System according to claim 1, wherein the FL Coordinator is further configured to:
 - read the current model and the model checkpoint from the Persistent Storage Device to transmit to the plurality of Computing Devices;
 - receive Updated Models from the plurality of Computing Devices for aggregation using Secure Multiparty Computation (SMPC); and
 - store a global checkpoint in the Persistent Storage Device.
11. The UI System according to claim 1, wherein the alternate data comprises at least social, financial, and device data stored in corresponding Computing Device associated with the user, and
 - wherein the social data corresponds to individual behaviour and character reliability.
12. A Unified Intelligence (UI) Method, comprising:
 - retrieving, by one or more processors in a Central Server, a current model and a model checkpoint from a Persistent Storage Device for a plurality of Computing Devices in a FL network;
 - distributing, by the one or more processors, the current model and the model checkpoint to a plurality of Computing Devices in accordance with a training plan;

receiving, by the one or more processors, updated and cryptographically masked models from the plurality of Computing Devices trained based on the received current model,

wherein the Updated Models are received for aggregation,

wherein the current model is trained by each of the plurality of Computing Devices using local data comprising alternate data from corresponding Computing Device associated with a user,

wherein the local data correspond to objects with outcome labels,

wherein availability of features of the objects with the outcome labels is in accordance with a defined configuration,

wherein the local data of each of the plurality of Computing Devices is encrypted such that personal data privacy of each Computing Device is protected from other Computing Devices;

generating, by the one or more processors, a Global Model based on model aggregation using SGD method on the Updated Models, which are received from the plurality of Computing Devices for aggregation, in accordance with the defined configuration for adaptive learning,

wherein the Global Model is re-distributed amongst the plurality of Computing Devices for next training round until convergence of the Global Model is achieved; and

estimating, by the one or more processors, using an Optimal Global Model corresponding to converged Global Model, an outcome label classification of a new unlabeled object characterized by corresponding features,

wherein the estimated outcome label classification of the new unlabeled object is used to determine, by an essential service provider, new data associated with the user.

13. The UI Method according to claim 12, wherein, in accordance with a first configuration that corresponds to the defined configuration for adaptive learning, the Central Server comprises a plurality of features and the outcome labels.

14. The UI Method according to claim 13, wherein the Method further comprising:

transmitting, by the one or more processors, a contribution of the features to an outcome function, and weights in combination with the outcome labels to each of the plurality of Computing Devices; and

receiving, by the one or more processors, gradient of the weights together with contribution of device-related features to the outcome function from each of the plurality of Computing Devices,

wherein the contribution of device-related features to the outcome function is computed by each of the plurality of Computing Devices, and

wherein the gradient of the weights are derived by each of the plurality of Computing Devices.

15. The UI Method according to claim 12, wherein, in accordance with a second configuration that corresponds to the defined configuration for adaptive learning, the Central Server comprises only the outcome labels.

16. The UI Method according to claim 15, wherein the Method further comprising:

transmitting, by the one or more processors, the outcome labels to each of the plurality of Computing Devices; and

receiving, by the one or more processors, gradient of the weights together with a contribution of device-related features to an outcome function from the plurality of Computing Devices,

wherein the contribution of device-related features to the outcome function is computed by each of the plurality of Computing Devices,

wherein the gradient of the weights are derived by each of the plurality of Computing Devices, and

wherein the gradient of the weights derived by each of the plurality of Computing Devices are sent to next Computing Devices together with the outcome labels.

17. The UI Method according to claim 12, wherein the outcome label classification corresponds to a binary classification of the new unlabeled object in accordance with a custom-built logical regression technique that uses a sigmoid function,
wherein the Optimal Global Model is a distributed and adaptive learning model.
18. The UI Method according to claim 12, wherein the alternate data comprises at least social, financial, and device data stored in corresponding Computing Device associated with the user, and
wherein the social data corresponds to individual behaviour and character reliability.
19. The UI Method according to claim 12, wherein, the Updated Model corresponds to information unrelated to the features in accordance with a privacy preserving learning process, and
wherein the information comprises updated set of aggregated value of weights multiplied by the features of labeled objects, the weights and corresponding gradients, and the outcome labels.
20. A non-transitory computer readable medium, having stored thereon, computer executable code, which when executed by a processor, cause the processor to execute operations, the operations comprising:
retrieving, by one or more processors in a Central Server, a current model and a model checkpoint from a Persistent Storage Device for a plurality of Computing Devices in a FL network;
distributing, by the one or more processors, the current model and the model checkpoint to a plurality of Computing Devices in accordance with a training plan;
receiving, by the one or more processors, updated and cryptographically masked models from the plurality of Computing Devices trained based on the received current model,

wherein the Updated Models are received for aggregation,

wherein the current model is trained by each of the plurality of Computing Devices using local data comprising alternate data from corresponding Computing Device associated with a user,

wherein the local data correspond to objects with outcome labels,

wherein availability of features of the objects with the outcome labels is in accordance with a defined configuration,

wherein the local data of each of the plurality of Computing Devices is encrypted such that personal data privacy of each Computing Device is protected from other Computing Devices;

generating a Global Model based on model aggregation using SGD method on the Updated Models, which are received from the plurality of Computing Devices for aggregation, in accordance with the defined configuration for adaptive learning,

wherein the Global Model is re-distributed amongst the plurality of Computing Devices for next training round until convergence of the Global Model is achieved; and

estimating, using an Optimal Global Model corresponding to converged Global Model, an outcome label classification of a new unlabeled object characterized by corresponding features,

wherein the estimated outcome label classification of the new unlabeled object is used to determine, by an essential service provider, new data associated with the user.