

**SYSTEMS AND METHODS FOR FEDERATED LEARNING USING
PEER-TO-PEER NETWORKS**

BACKGROUND OF THE INVENTION

1. **Field of the Invention**

[0001] Embodiments generally relate to systems and methods for federated learning using peer-to-peer networks.

2. **Description of the Related Art**

[0002] In the era of distributed computing, scale of data and computational resources are often handled by distribution of workloads across multiple systems in horizontal direction. Distributed computing, and in particular distributed machine learning, opens many exciting opportunities. It also introduces new challenges in areas where data privacy and data security are important. Designing a resilient, highly available, and robust ecosystem for machine learning is equally challenging. Federated learning lays the foundation for implementing machine learning on a distributed landscape where heterogeneous machines can participate in collaborative manner.

[0003] Federated learning generally involves an “aggregator node” and set of “participant nodes” in a federated network. Current available implementations have a centralized design of an aggregator node and various participant nodes, and they form a star topology for communication. The participant nodes send their local model gradients to the aggregator node, and the aggregator node composes all the received models together into a singular global model. This singular global model is then sent back to all participant nodes. In this setup, the participants benefit from common learning, without knowing about all the underlying data.

[0004] While federated learning does take care of privacy, security, and anonymity, aggregator nodes become the single point of failure in the federated

network. This poses a risk of downtime in case of unwanted failures, which may lead to the stoppage of critical process on the network.

SUMMARY OF THE INVENTION

[0005] Systems and methods for federated learning using peer-to-peer networks are disclosed. According to an embodiment, a method for federated learning using peer-to-peer networks may include: (1) electing, by a plurality of participant nodes in a peer-to-peer network, one of the participant nodes as a collaborator node using a consensus algorithm; (2) generating, by the collaborator node, a collaborator node public key and a collaborator node private key; (3) broadcasting, by the collaborator node and to the plurality of participant nodes, the collaborator node public key; (4) generating, by each participant node, a participant node public key and a participant node private key, wherein each participant node generates a new participant node public key and a new participant node private key for each communication with the collaborator node; (5) encrypting, by each participant node, a message comprising a parameter for a local machine learning model for the participant node and the participant node public key with the collaborator node public key; (6) broadcasting, by each participant node, the encrypted message on the peer-to-peer network; (7) decrypting, by the collaborator node, each of the encrypted messages with the collaborator node private key; (8) updating, by the collaborator node, an aggregated machine learning model with the decrypted parameters for the local machine learning models; (9) encrypting, by the collaborator node, a plurality of update messages each comprising an update from the aggregated machine learning model with each participant node's public key; (10) broadcasting, by the collaborator node, the plurality of messages on the peer-to-peer network; (11) decrypting, by each of the participant nodes, one of the plurality of messages with the participant node private key for the participant node; and (12) updating, by each of the participant nodes, the local machine learning model for the participant node with the update.

[0006] In one embodiment, the consensus algorithm may be the Raft consensus algorithm.

[0007] In one embodiment, the parameter may include information related to model exchange, local machine learning model weights, and/or the local machine learning model.

[0008] In one embodiment, the parameter may include clear data and/or synthetic data.

[0009] In one embodiment, the collaborator node may perform model aggregation using the decrypted parameters.

[0010] In one embodiment, the collaborator node may train the aggregated machine learning model using the decrypted parameters.

[0011] In one embodiment, the participant node is the collaborator node for a limited period.

[0012] In one embodiment, the collaborator node may broadcast a heartbeat to the participant nodes.

[0013] In one embodiment, the participant nodes may elect a new collaborator node in response to the collaborator node being inactive.

[0014] According to another embodiment, a system may include a plurality of participant nodes, each participant node associated with a local machine learning model and a peer-to-peer network connecting the plurality of participant nodes. The plurality of participant nodes may elect one of the plurality of participant nodes as a collaborator node using a consensus algorithm; the collaborator node may generate a collaborator node public key and a collaborator node private key and may broadcast the collaborator node public key to the plurality of participant nodes; each participant node may generate a participant

node public key and a participant node private key, wherein each participant node generates a new participant node public key and a new participant node private key for each communication with the collaborator node, may encrypt a message comprising a parameter for a local machine learning model for the participant node and the participant node public key with the collaborator node public key, and may broadcast the encrypted message on the peer-to-peer network; the collaborator node may decrypt each of the encrypted messages with the collaborator node private key, may update an aggregated machine learning model with the decrypted parameters for the local machine learning models, may encrypt a plurality of update messages each comprising an update from the aggregated machine learning model with each participant node's public key, and may broadcast the plurality of messages on the peer-to-peer network; each of the participant nodes may decrypt one of the plurality of messages with its participant node private key for the participant node and may update its local machine learning model with the update.

[0015] In one embodiment, the consensus algorithm is the Raft consensus algorithm.

[0016] In one embodiment, the parameter may include information related to model exchange, local machine learning model weights, and/or the local machine learning model.

[0017] In one embodiment, the parameter may include clear data and/or synthetic data.

[0018] In one embodiment, the collaborator node may perform model aggregation using the decrypted parameters.

[0019] In one embodiment, the collaborator node may train the aggregated machine learning model using the decrypted parameters.

[0020] In one embodiment, the participant node may be the collaborator node for a limited period.

[0021] In one embodiment, the collaborator node may broadcast a heartbeat to the participant nodes.

[0022] In one embodiment, the participant nodes may elect a new collaborator node in response to the collaborator node being inactive.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] For a more complete understanding of the present invention, the objects and advantages thereof, reference is now made to the following descriptions taken in connection with the accompanying drawings in which:

[0024] Figure 1 depicts a system for federated learning using peer-to-peer networks according an embodiment;

[0025] Figure 2 depicts a method for federated learning using peer-to-peer networks according an embodiment;

[0026] Figure 3 depicts a state diagram for participant nodes in a peer-to-peer network according to an embodiment.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0027] Embodiments are directed to systems and methods for federated learning using peer-to-peer network for decentralized orchestration of model weights.

[0028] The disclosure of Indian Patent Application No. 202011050561 filed November 20, 2020 is hereby incorporated, by reference, in its entirety.

[0029] Embodiments are generally directed to a resilient and highly available aggregator service through decentralization with federated learning using peer-to-peer communication among the participant nodes in a federated network. All of the participant nodes may have equivalent capabilities on the network, and all may be able to communicate with each other. At any given point of time, the network may have only one of the participant nodes acting as a “leader” (or collaborator node) while the rest of the participant nodes act as “follower nodes.” In case the leader node becomes unavailable or unresponsive (e.g., it crashes), the network conducts a new leader election to select a leader node among the remaining participant nodes. This may follow a Raft consensus algorithm to conduct election and assign leaders.

[0030] The leader node may be responsible for acting as transient aggregator for the network. In embodiments, to distribute the workload among the participant nodes, and give a fair chance to every participant node within the network to assume the role of the “leader node,” a new leader node may be elected after a predefined time interval is elapsed.

[0031] Referring to Figure 1, a system for federated learning using peer-to-peer network is disclosed according to an embodiment. System 100 may include a plurality of participant nodes 110 (e.g., 110₁, 110₂, 110₃, ... 110_N) in a peer-to-peer network, where each participant node 110 may communicate directly with each of the other participant nodes 110. Each participant node 110 may maintain local model 112 (e.g., 112₁, 112₂, 112₃, ... 112_N), and one of the participant nodes (e.g., participant 110₁,) may be elected to be a leader node (or collaborator node) may generate aggregated model 114 for the participant nodes in the federated network.

[0032] Each node may execute a computer program, application, script, etc. (not shown) that may control operations of the respective node.

[0033] Each participant node 110 may participate as a leader node (collaborator node), a follower node, or a candidate node. To facilitate the communication between participant nodes 110, system 100 provides peer-to-peer communication between participant nodes 110. The communications may be facilitated by making remote procedure calls using a TCP layer of networking.

[0034] Because the network is a peer-to-peer network, all messages are broadcast to the network. Based on the message type, if encrypted, only the valid recipient can read the message. If the message is not encrypted, all recipients may read the message and take necessary actions.

[0035] As in any peer-to-peer network, all of the participant nodes 110 need to be aware of all the participant nodes 110 present in the network, along with firewalls open for seamless TCP based communication. The RSA public and private keys involved in encryption are dynamic due to the transient nature of the leader. For example, the pair of RSA keys may be rotated with every message, initiated by one of the participant nodes.

[0036] In case of completion of election, the leader node announces its RSA public key to the network, so that it can be used by the participant nodes to send messages, such as local machine learning model parameters, to the leader node.

[0037] Referring to Figure 2, a method for federated learning using peer-to-peer networks is disclosed according to an embodiment.

[0038] In step 205, the participant nodes in peer-to-peer network may elect one of the participant nodes to be a leader node, or collaborator node, using, for example, the Raft consensus algorithm. An example state diagram for each participant node based on the Raft consensus algorithm is provided in Figure 3.

[0039] Figure 3 depicts a state diagram for participant nodes according to an embodiment. Figure 3 depicts three states for each node in the peer-to-peer

network: State 1, where the node is a follower, or participant, node; State 2, where the node is a candidate node; and State 3, wherein the node is a leader, or collaborator, node.

[0040] Each node may be initialized to be a follower or participant node (State 1). From State 1, the follower node may initiate an election to be a leader node, thereby becoming a candidate node (State 2). In State 2, the node may become a leader node (State 3), no leader may be elected and it may remain a candidate node for another election (State 2), or another node may be elected to be a leader, and the node may return to being a follower, or participant, node (State 1). In State 3, if the node identifies a node with a higher priority, or resigns, the node may return to being a follower, or participant, node (State 1).

[0041] The nodes may use the Raft consensus algorithm to elect a leader node (State 3). State 3 is designed to be transient. If there are no failures, the elected node stays as leader node for a period T, which may be defined by, for example, a network administrator. After the period T is elapsed, the leader node resigns, becomes a follower node (State 1), thereby indirectly initiating an election.

[0042] An example of the Raft consensus algorithm is provided in Ongaro, et al. "In search of an understandable consensus algorithm." In Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference (USENIX ATC '14), Gibson, G. and Zeldovich (Eds.): 305-329, the disclosure of which is incorporated, by reference, in its entirety.

[0043] In embodiments, all the participant nodes may expect a heartbeat from the leader node (e.g., every 5 seconds) to ensure the leader node is functioning properly. In case heartbeat fails, which means the leader node is inactive for some arbitrary reason, one or more of the follower nodes may promote themselves to be candidate nodes (State 2). After a fair election, one of the candidate nodes is elected as the leader node, while the other candidate nodes return to State 1.

[0044] Referring again to Figure 2, in step 210, the collaborator node may generate a public and private key pair, and in step 215 may share its public key with the other participant nodes in peer-to-peer network.

[0045] In step 220, the participant nodes in the peer-to-peer network may generate their public and private keys, and in step 225, each participant node may encrypt a message with the collaborator node's public key and send to the collaborator node. In one embodiment, the participant nodes may generate new public/private key pairs for each communication, which prevents the collaborator node from identifying an incoming message from a specific node. This dynamic message encryption-decryption thus provides anonymity for the participant nodes in the peer-to-peer network from message flow perspective.

[0046] In one embodiment, the message may be broadcast to nodes (i.e., all participant nodes and the leader node) in the peer-to-peer network. The leader node is the only node that can decrypt the message with its private key.

[0047] In one embodiment, the message may include parameters for a local machine learning model such as information related to model exchange, local weights, the local machine learning model itself, clear data, synthetic data, etc.

[0048] In one embodiment, the message may further include the participant node's public key. Thus, each participant node may encrypt the parameters and its public key with the collaborator node's public key.

[0049] In step 230, the collaborator node may receive the messages and decrypt the messages using its private key.

[0050] In step 235, the collaborator node may perform an operation on the received parameters, such as model aggregation, to create or update an aggregated machine learning model, etc. Notably, the collaborator node may include parameters and/or information for its local model in the operation.

[0051] In another embodiment, the collaborator node may train the aggregated machine learning model with data received from the participant nodes as well as its own data.

[0052] In step 240, the collaborator node may encrypt a message with each participant node's public key and broadcast the messages on the peer-to-peer network. In one embodiment, the message may include updates to the aggregated machine learning model.

[0053] In step 245, each participant node may decrypt the message from collaborator node, and may take any necessary actions with the contents of the message, such as updating its local machine learning model. Although the participant nodes receive all messages from the collaborator node, they can only decrypt the message encrypted with their public key.

[0054] The process may continue until a condition is reached, such as the local machine learning models have converged, etc.

[0055] The disclosure of Behera, Monik et al, (2021). "Federated Learning using Peer-to-peer Network for Decentralized Orchestration of Model Weights." TechRxiv. Preprint. (available at doi.org/10.36227/techrxiv.14267468.v1).

[0056] Hereinafter, general aspects of implementation of the systems and methods of embodiments will be described.

[0057] Embodiments of the system or portions of the system may be in the form of a "processing machine," such as a general-purpose computer, for example. As used herein, the term "processing machine" is to be understood to include at least one processor that uses at least one memory. The at least one memory stores a set of instructions. The instructions may be either permanently or temporarily stored in the memory or memories of the processing machine. The processor executes the instructions that are stored in the memory or memories in order to

process data. The set of instructions may include various instructions that perform a particular task or tasks, such as those tasks described above. Such a set of instructions for performing a particular task may be characterized as a program, software program, or simply software.

[0058] In one embodiment, the processing machine may be a specialized processor.

[0059] In one embodiment, the processing machine may a cloud-based processing machine, a physical processing machine, or combinations thereof.

[0060] As noted above, the processing machine executes the instructions that are stored in the memory or memories to process data. This processing of data may be in response to commands by a user or users of the processing machine, in response to previous processing, in response to a request by another processing machine and/or any other input, for example.

[0061] As noted above, the processing machine used to implement embodiments may be a general-purpose computer. However, the processing machine described above may also utilize any of a wide variety of other technologies including a special purpose computer, a computer system including, for example, a microcomputer, mini-computer or mainframe, a programmed microprocessor, a micro-controller, a peripheral integrated circuit element, a CSIC (Customer Specific Integrated Circuit) or ASIC (Application Specific Integrated Circuit) or other integrated circuit, a logic circuit, a digital signal processor, a programmable logic device such as a FPGA, PLD, PLA or PAL, or any other device or arrangement of devices that is capable of implementing the steps of the processes disclosed herein.

[0062] The processing machine used to implement embodiments may utilize a suitable operating system.

[0063] It is appreciated that in order to practice the method of the embodiments as described above, it is not necessary that the processors and/or the memories of the processing machine be physically located in the same geographical place. That is, each of the processors and the memories used by the processing machine may be located in geographically distinct locations and connected so as to communicate in any suitable manner. Additionally, it is appreciated that each of the processor and/or the memory may be composed of different physical pieces of equipment. Accordingly, it is not necessary that the processor be one single piece of equipment in one location and that the memory be another single piece of equipment in another location. That is, it is contemplated that the processor may be two pieces of equipment in two different physical locations. The two distinct pieces of equipment may be connected in any suitable manner. Additionally, the memory may include two or more portions of memory in two or more physical locations.

[0064] To explain further, processing, as described above, is performed by various components and various memories. However, it is appreciated that the processing performed by two distinct components as described above, in accordance with a further embodiment, may be performed by a single component. Further, the processing performed by one distinct component as described above may be performed by two distinct components.

[0065] In a similar manner, the memory storage performed by two distinct memory portions as described above, in accordance with a further embodiment, may be performed by a single memory portion. Further, the memory storage performed by one distinct memory portion as described above may be performed by two memory portions.

[0066] Further, various technologies may be used to provide communication between the various processors and/or memories, as well as to allow the processors

and/or the memories to communicate with any other entity; i.e., so as to obtain further instructions or to access and use remote memory stores, for example. Such technologies used to provide such communication might include a network, the Internet, Intranet, Extranet, LAN, an Ethernet, wireless communication via cell tower or satellite, or any client server system that provides communication, for example. Such communications technologies may use any suitable protocol such as TCP/IP, UDP, or OSI, for example.

[0067] As described above, a set of instructions may be used in the processing of embodiments. The set of instructions may be in the form of a program or software. The software may be in the form of system software or application software, for example. The software might also be in the form of a collection of separate programs, a program module within a larger program, or a portion of a program module, for example. The software used might also include modular programming in the form of object-oriented programming. The software tells the processing machine what to do with the data being processed.

[0068] Further, it is appreciated that the instructions or set of instructions used in the implementation and operation of embodiments may be in a suitable form such that the processing machine may read the instructions. For example, the instructions that form a program may be in the form of a suitable programming language, which is converted to machine language or object code to allow the processor or processors to read the instructions. That is, written lines of programming code or source code, in a particular programming language, are converted to machine language using a compiler, assembler or interpreter. The machine language is binary coded machine instructions that are specific to a particular type of processing machine, i.e., to a particular type of computer, for example. The computer understands the machine language.

[0069] Any suitable programming language may be used in accordance with the various embodiments. Further, it is not necessary that a single type of instruction or single programming language be utilized in conjunction with the operation of the system and method. Rather, any number of different programming languages may be utilized as is necessary and/or desired.

[0070] Also, the instructions and/or data used in the practice of embodiments may utilize any compression or encryption technique or algorithm, as may be desired. An encryption module might be used to encrypt data. Further, files or other data may be decrypted using a suitable decryption module, for example.

[0071] As described above, the embodiments may illustratively be embodied in the form of a processing machine, including a computer or computer system, for example, that includes at least one memory. It is to be appreciated that the set of instructions, i.e., the software for example, that enables the computer operating system to perform the operations described above may be contained on any of a wide variety of media or medium, as desired. Further, the data that is processed by the set of instructions might also be contained on any of a wide variety of media or medium. That is, the particular medium, i.e., the memory in the processing machine, utilized to hold the set of instructions and/or the data used in embodiments may take on any of a variety of physical forms or transmissions, for example. Illustratively, the medium may be in the form of paper, paper transparencies, a compact disk, a DVD, an integrated circuit, a hard disk, a floppy disk, an optical disk, a magnetic tape, a RAM, a ROM, a PROM, an EPROM, a wire, a cable, a fiber, a communications channel, a satellite transmission, a memory card, a SIM card, or other remote transmission, as well as any other medium or source of data that may be read by the processors.

[0072] Further, the memory or memories used in the processing machine that implements embodiments may be in any of a wide variety of forms to allow the

memory to hold instructions, data, or other information, as is desired. Thus, the memory might be in the form of a database to hold data. The database might use any desired arrangement of files such as a flat file arrangement or a relational database arrangement, for example.

[0073] In the systems and methods, a variety of “user interfaces” may be utilized to allow a user to interface with the processing machine or machines that are used to implement embodiments. As used herein, a user interface includes any hardware, software, or combination of hardware and software used by the processing machine that allows a user to interact with the processing machine. A user interface may be in the form of a dialogue screen for example. A user interface may also include any of a mouse, touch screen, keyboard, keypad, voice reader, voice recognizer, dialogue screen, menu box, list, checkbox, toggle switch, a pushbutton or any other device that allows a user to receive information regarding the operation of the processing machine as it processes a set of instructions and/or provides the processing machine with information. Accordingly, the user interface is any device that provides communication between a user and a processing machine. The information provided by the user to the processing machine through the user interface may be in the form of a command, a selection of data, or some other input, for example.

[0074] As discussed above, a user interface is utilized by the processing machine that performs a set of instructions such that the processing machine processes data for a user. The user interface is typically used by the processing machine for interacting with a user either to convey information or receive information from the user. However, it should be appreciated that in accordance with some embodiments of the system and method, it is not necessary that a human user actually interact with a user interface used by the processing machine. Rather, it is also contemplated that the user interface might interact, i.e., convey and receive information, with another processing machine, rather than a human user.

Accordingly, the other processing machine might be characterized as a user. Further, it is contemplated that a user interface utilized in the system and method may interact partially with another processing machine or processing machines, while also interacting partially with a human user.

[0075] It will be readily understood by those persons skilled in the art that embodiments are susceptible to broad utility and application. Many embodiments and adaptations of the present invention other than those herein described, as well as many variations, modifications and equivalent arrangements, will be apparent from or reasonably suggested by the foregoing description thereof, without departing from the substance or scope.

[0076] Accordingly, while embodiments present invention has been described here in detail in relation to its exemplary embodiments, it is to be understood that this disclosure is only illustrative and exemplary of the present invention and is made to provide an enabling disclosure of the invention. Accordingly, the foregoing disclosure is not intended to be construed or to limit the present invention or otherwise to exclude any other such embodiments, adaptations, variations, modifications or equivalent arrangements.

CLAIMS

What is claimed is:

1. A method for federated learning using peer-to-peer networks, comprising:
 - electing, by a plurality of participant nodes in a peer-to-peer network, one of the participant nodes as a collaborator node using a consensus algorithm;
 - generating, by the collaborator node, a collaborator node public key and a collaborator node private key;
 - broadcasting, by the collaborator node and to the plurality of participant nodes, the collaborator node public key;
 - generating, by each participant node, a participant node public key and a participant node private key, wherein each participant node generates a new participant node public key and a new participant node private key for each communication with the collaborator node;
 - encrypting, by each participant node, a message comprising a parameter for a local machine learning model for the participant node and the participant node public key with the collaborator node public key;
 - broadcasting, by each participant node, the encrypted message on the peer-to-peer network;
 - decrypting, by the collaborator node, each of the encrypted messages with the collaborator node private key;
 - updating, by the collaborator node, an aggregated machine learning model with the decrypted parameters for the local machine learning models;
 - encrypting, by the collaborator node, a plurality of update messages each comprising an update from the aggregated machine learning model with each participant node's public key;
 - broadcasting, by the collaborator node, the plurality of messages on the peer-to-peer network;

decrypting, by each of the participant nodes, one of the plurality of messages with the participant node private key for the participant node; and

updating, by each of the participant nodes, the local machine learning model for the participant node with the update.

2. The method of claim 1, wherein the consensus algorithm is the Raft consensus algorithm.

3. The method of claim 1, wherein the parameter comprises information related to model exchange, local machine learning model weights, and/or the local machine learning model.

4. The method of claim 1, wherein the parameter comprises clear data and/or synthetic data.

5. The method of claim 1, wherein the collaborator node performs model aggregation using the decrypted parameters.

6. The method of claim 1, wherein the collaborator node trains the aggregated machine learning model using the decrypted parameters.

7. The method of claim 1, wherein the participant node is the collaborator node for a limited period.

8. The method of claim 1, wherein the collaborator node broadcasts a heartbeat to the participant nodes.

9. The method of claim 1, wherein the participant nodes elect a new collaborator node in response to the collaborator node being inactive.

10. A system, comprising:
a plurality of participant nodes, each participant node associated with a local machine learning model; and
a peer-to-peer network connecting the plurality of participant nodes;
wherein:

the plurality of participant nodes elect one of the plurality of participant nodes as a collaborator node using a consensus algorithm;

the collaborator node generates a collaborator node public key and a collaborator node private key;

the collaborator node broadcasts the collaborator node public key to the plurality of participant nodes;

each participant node generates a participant node public key and a participant node private key, wherein each participant node generates a new participant node public key and a new participant node private key for each communication with the collaborator node;

each participant node encrypts a message comprising a parameter for a local machine learning model for the participant node and the participant node public key with the collaborator node public key;

each participant node broadcasts the encrypted message on the peer-to-peer network;

the collaborator node decrypts each of the encrypted messages with the collaborator node private key;

the collaborator node updates an aggregated machine learning model with the decrypted parameters for the local machine learning models;

the collaborator node encrypts a plurality of update messages each comprising an update from the aggregated machine learning model with each participant node's public key;

the collaborator node broadcasts the plurality of messages on the peer-to-peer network;

each of the participant nodes decrypts one of the plurality of messages with its participant node private key for the participant node; and

each of the participant nodes updates its local machine learning model with the update.

11. The system of claim 10, wherein the consensus algorithm is the Raft consensus algorithm.

12. The system of claim 10, wherein the parameter comprises information related to model exchange, local machine learning model weights, and/or the local machine learning model.

13. The system of claim 10, wherein the parameter comprises clear data and/or synthetic data.

14. The system of claim 10, wherein the collaborator node performs model aggregation using the decrypted parameters.

15. The system of claim 10, wherein the collaborator node trains the aggregated machine learning model using the decrypted parameters.

16. The system of claim 10, wherein the participant node is the collaborator node for a limited period.

17. The system of claim 10, wherein the collaborator node broadcasts a heartbeat to the participant nodes.

18. The system of claim 10, wherein the participant nodes elect a new collaborator node in response to the collaborator node being inactive.