grammarly_764230

by Vbn

General metrics

51,535	7,476	365	29 min 54 sec	57 min 30 sec
characters	words	sentences	reading time	speaking time

Score

82

This text scores better than 82% of all texts checked by Grammarly

Writing Issues

271	123	148
Issues left	Critical	Advanced

Plagiarism

This text hasn’t been checked for plagiarism

Writing Issues

137	**Correctness**
3	Text inconsistencies
9	Comma misuse within clauses
29	Incorrect punctuation
50	Ungrammatical sentence
16	Determiner use (a/an/the/this, etc.)
1	Pronoun use
7	Confused words
2	Wrong or missing prepositions
14	Improper formatting
4	Incorrect phrasing
2	Incorrect noun number

12	**Engagement**
12	Word choice

118	**Clarity**
108	Paragraph can be improved
3	Hard-to-read text
1	Passive voice misuse
4	Intricate text
2	Wordy sentences

4	**Delivery**
2	Inappropriate colloquialisms
2	Tone suggestions

Unique Words

Measures vocabulary diversity by calculating the percentage of words used only once in your document

18%

unique words

Rare Words

Measures depth of vocabulary by identifying words that are not among the 5,000 most common English words.

49%

rare words

Word Length

Measures average word length

5.6

characters per word

Sentence Length

Measures average sentence length

20.5

words per sentence

grammarly_764230

HashLearner: A Secure Decentralized Learning Framework Based on Hash Graph

Federated learning enables collaborative model training without centralized data collection, but existing frameworks rely on a central server, introducing risks of single points of failure, adversarial manipulation, and privacy leakage. To[1] address these challenges, we propose HashLearner, a secure decentralized learning framework that utilizes the HashGraph consensus protocol for model aggregation without trusted authorities. HashLearner[1] introduces two key innovations: (i) a consensus-driven decentralized aggregation mechanism resilient to Byzantine adversaries, and (ii) a privacy-preserving shuffling strategy that mitigates gradient reconstruction and poisoning attacks. To[1] handle heterogeneous data distributions, the framework further employs transfer learning–based personalization. The[1] simulation results of HashLearner, tested on benchmark Kaggle datasets, demonstrate that the platform maintains high accuracy while significantly enhancing scalability, security and[2] privacy. These[1] findings demonstrate[3] that HashLearner provides a practical path toward scalable, privacy-preserving, and trustworthy decentralized federated learning.

Introduction

The deep learning community increasingly seeks decentralized solutions that allow[4] organizations to jointly train models and carry out[4] computations while operating in untrusted environments. Some[1] approaches to solve this problem have been proposed in recent years including[5] Federated Learning (FL)

\cite{refre01}, \cite{refre02}, \cite{refre05} \cite{refre0019} and Gossip Learning (GL) \cite{refre03}, \cite{refre09}, \cite{refre11}.

The increasing demand for privacy-preserving machine learning has advanced[6] the rise of FL, where[6] multiple organizations collaboratively train a shared model without exchanging raw data. By[1] keeping sensitive datasets local, FL mitigates risks associated with centralized data collection, such as unauthorized access, privacy breaches, and regulatory non-compliance. They[1] try to distribute the training phase through a decentralized network of nodes which is[7,8] multi-organizational which means[7,8] each node can have a different owner than other nodes.

In recent years, FL has emerged as a promising paradigm for collaborative machine learning, enabling multiple participants to train a shared model without exposing their private data. By[1] decentralizing the training process, FL addresses critical privacy concerns and reduces the risks associated with[9] data centralization. Despite[1] these advantages, most FL frameworks rely on a central aggregation server to coordinate training. This[1] dependence introduces critical limitations:

Single point of failure: If the server is compromised or fails, the system collapses.

Adversarial vulnerability: Malicious participants can launch poisoning or reconstruction attacks, such as those using generative adversarial networks (GANs).

Privacy risks: The server may infer sensitive information from model updates.

Scalability challenges: Centralized coordination struggles to handle large numbers of clients in real-world distributed networks.

While alternatives such as gossip learning and blockchain-based FL have been proposed, gossip learning suffers from slow convergence and inconsistency under heterogeneous data, whereas blockchain-based[10] solutions face scalability bottlenecks due to latency and resource-intensive consensus mechanisms.

This paper proposes a Secure Decentralized Learning Framework Based on HashGraph called HashLearner, which eliminates the dependency on a central server and leverages the[11] unique properties of HashGraph[11], a distributed ledger technology known for its high throughput, fairness, and Byzantine fault tolerance. By[1] integrating HashGraph into the FL process, the proposed framework ensures secure and efficient coordination among participants, even in the presence of adversarial actors. The[1] decentralized nature of HashGraph not only enhances the resilience of the[12] system but also mitigates the[12] risks associated with centralized control, such as data manipulation and unauthorized access. A[1] key contribution of this work is the framework's ability to counteract GAN attacks, which have become a significant threat in FL environments. GAN[1] attacks exploit the collaborative nature of FL to generate malicious updates that can compromise the integrity of the global model.

The proposed framework incorporates advanced probabilistic consensus mechanisms to detect and neutralize such attacks, ensuring the reliability and trustworthiness of the learning process. The[1] proposed framework offers several advantages over traditional FL systems, including improved scalability, enhanced security, and greater resistance to adversarial interference. By[1] removing the central server and leveraging HashGraph's decentralized architecture, the proposed approach paves the way for[13] more secure and[13] privacy-preserving machine learning in distributed environments. This[1] paper explores the design, implementation and[14] evaluation of the framework, demonstrating its effectiveness in mitigating GAN attacks and its potential to revolutionize the field of FL.

The rest of the paper is structured as follows: In[15] Section 2,[15] related work in this area is reviewed.[15] Section[1] 3 introduces some preliminary topics necessary for understanding our approach and[16] the proposed HashLearner platform is explained in detail. Section[1] 4 focuses on security and performance analysis of the proposed scheme. Finally,[1] Section 5 discusses the conclusions drawn from this research.

Related Works

Two prominent approaches in Decentralized[17] learning domain are Gossip Learning and Federated Learning, each offering unique advantages and challenges in decentralized settings. Gossip[1] Learning \cite{refre09} is a fully decentralized approach where participants (nodes) exchange model updates directly with their neighbors in a peer-to-peer manner. This[1] method eliminates the need for a central coordinator, making it highly scalable and resilient to single points of failure. Gossip[1] protocols rely on randomized communication to propagate updates across the network, ensuring that knowledge is disseminated efficiently. Gossip[1] learning faces two major challenges: (i) poor performance on heterogeneous datasets, where local data distributions vary significantly across nodes, and (ii) slower convergence compared to centralized or semi-decentralized FL approaches. In[1] gossip learning, each node trains its local model using its local dataset the same as[18,19] FL but[18] the point is that[19] there is no global initial model and[18,19] each node initializes the global model's initial[19] weights independently. After[1] training local models, each node tries to exchange its model's weights with some other nodes in multiple rounds and[20] at each round[20] it merges its model with the other node's model. After[1] some rounds step[21] by step[21] the models can be converged[22] more to some same[21,22] distribution across multiple nodes. Hence[1,23] GL might not be a good target in heterogeneous networks with varying computational resources and communication delays because of[24] the lack of a structured coordination mechanism and inconsistencies in model convergence \cite{refre11}. \\

On the other hand, FL has emerged as a widely adopted framework for decentralized learning, particularly in scenarios where data privacy is paramount. In[1] FL[25] the target[26] is to train a high quality[25,26] centralized shared model,[26] while training data is disbursed in[25,26] a large number of each customer[25] with an[25,26] untrustable and[25,26] relatively slow network connection[25,26] under the coordination of a central server. The[1] server aggregates local model updates from participants and distributes the global model back to them. \\

While FL has demonstrated success in applications such as mobile devices and healthcare, its reliance on a central server introduces several limitations. These[1] include vulnerability to single points of failure, potential privacy breaches, and susceptibility to poisoning and adversarial attacks, such as reconstruction[27] attack. Poisoning[1] attacks can be used for[28] two different targets. One[1] of them is to corrupt the model by inserting wrong and[30] corrupted data with a wrong[30] data distribution into the dataset which[29] is called passive[29] attack and the[29,30] other one[30] is for hiding[29,30] a backdoor inside the trained model by employing specific labels or data features in the dataset which[29] is called active[29] attack. In[1] the reconstruction attack, the adversary tries to reconstruct a shadow model based on[32] the target model weights or gradients to find out some info[32] about the data in the[32] dataset in different ways[32] such[31] as dataset[32] distribution, existence[32] of specific data pieces in the dataset[32], or some[32] properties of the dataset[32]. As[1] an[33] example, the adversary can gain the ability to[33] learn if[33] an image is in the dataset,[33] or know[33] the number of people with some[33] specific properties in the dataset. Reconstruction[1] attacks can be implemented by employing GANs.

In GAN attacks, malicious participants generate fake updates to manipulate the global model, compromising its integrity and performance. A[1] GAN comprises a generator and a discriminator trained in adversarial competition: the generator produces synthetic samples, while the discriminator attempts to distinguish them from real data. This[1] dynamic enables adversaries in FL to craft poisoned updates or reconstruct private data. On[1] the other hand, the discriminator tries to find out[34] the fake samples produced[34] by the generator. The[1] competition between these two parts leads to learning more[36] features and advancement of the generator to produce fake samples which[36] are[35] so similar to real samples. In[1] the reconstruction attack, the adversary can use GANs to produce[37] a shadow model and find[39] some info[39] about the dataset of a node as[38] mentioned before. Multiple[1] approaches as counter-attacks had[41] been proposed against these attacks such[40] as Encryption, Shuffling, Differential Privacy[41] and[40] in some ways[41] inserting noise into dataset[40] which[40] were helpful to[41] make the system more safe[40,41] against some of the mentioned attacks \cite{refre05}.

To address these challenges, recent research has explored the integration of[42] blockchain technology into FL, aiming[42] to decentralize the aggregation process and enhance security. Blockchain-based[1] FL systems replace the central server with a distributed ledger, enabling transparent and tamper-proof record-keeping of model updates. However[1], traditional blockchain systems often suffer from scalability issues, high latency, and energy inefficiency, which hinder their applicability in large-scale FL scenarios. In[1] this context, HashGraph emerges as a promising alternative to blockchain[43] for decentralized learning \cite{refre14}. HashGraph[1] \cite{refre17} is a distributed ledger technology that utilizes[44] a gossip-about-gossip[44] protocol and virtual voting to achieve consensus in a highly efficient and secure manner. Unlike[1] blockchain[43], HashGraph does not rely on resource-intensive proof-of-work mechanisms, making it more scalable and energy-efficient. Its[1] inherent properties, such as Byzantine fault tolerance, fairness, and asynchronous communication, make it suitable for decentralized learning frameworks.

HashGraph offers higher throughput than proof-of-stake blockchains and is therefore suitable for AI training networks in which[45] multiple models are trained[4] simultaneously. Moreover[1], it provides asynchronous Byzantine fault tolerance, which the HashGraph designers describe as the strongest attainable level of security in asynchronous consensus. Generally[1] this framework which uses HashGraph, is the best for implementing an AI training platform between different organizations whose goal is to train AI models or expand existing AI models everyday rapidly which needs high throughput, because they may want to transfer many models at the same time in each training round, and HashGraph's[46] higher throughput helps to increase aggregation and final model production speed efficiently.

HashGraph is a decentralized network architecture consisting of nodes[47] each holding[48] a copy of the HashGraph data including[47] its events[48] graph. HashGraph[1] can be used to store data in a secure and[50] decentralized manner[49] in which[50] it is[50] hard for attackers to corrupt its stability and consistency including[49] the order of events in the HashGraph[50].

This research tries[51] to improve decentralized learning approaches to aggregate[51] a global model in a decentralized manner[51]. As[1] an example in the FedHealth paper \cite{refre03}, the proposed approach needs a central server to aggregate the global model at the end of the training phase. As[1] shown in figure[52] \ref{fig1:sample}, FedHealth proposed a federated learning framework improved by transfer learning. It[1] is optimized for healthcare field model training and deep learning approaches. They[1] employed the transfer[54][53] learning to adapt the global aggregated model to each node's local dataset data[54] distribution by training the global aggregated model[54] on the local datasets separately on each node \cite{refre03}.

Related work includes blockchain-based federated learning (BCFL) \cite{refre13}, which explores similar decentralization concepts.

In federated learning frameworks, the aggregator plays a crucial role by collecting[55] and combining training results from local sources[55] to update the global model. However[1], this central aggregator poses reliability, security, and availability risks. A[1] compromised aggregator can jeopardize the entire federated learning system, making it vulnerable to adversarial attacks. Potential[1] threats include:

Malicious aggregation behavior

Network connectivity disruptions and denial-of-service attacks

External security breaches and exploits \cite{refre13}\\.

To address these primary challenges and other[56] related issues, they implemented blockchain technology to finalize and share models through[56] an immutable distributed ledger similar to HashLearner's approach using HashGraph. However[1], HashGraph achieves higher throughput than blockchain[43] due to its asynchronous consensus architecture.

Another blockchain-based FL framework \cite{refre14} secures IoT devices by utilizing them as blockchain nodes and employing a blockchain consensus mechanism for model sharing. While[1] conceptually similar to previous work, HashLearner demonstrates superior speed owing to its consensus mechanism.

As noted previously, federated learning architectures are inherently vulnerable to both distributed denial-of-service (DDoS) attacks and privacy violations, including model data leakage. These[1] vulnerabilities arise from malicious activities such as attempts to alter or steal confidential client data through manipulated model updates and weights. Furthermore[1], centralized FL systems face scalability challenges, particularly when handling the increasing volume of updates from growing IoT networks, which demand substantial computational resources \cite{refre15}, \cite{refre16}\\.

Blockchain technology, as a decentralized and immutable ledger system, offers several advantages, including decentralization, tamper resistance, and enhanced security. These properties effectively address the limitations of centralized FL systems that rely on a single server under a central authority. By eliminating dependence on a central server, the blockchain mitigates single points of failure and associated security risks.

The FeDis framework,[57] representing an approach that[57,58] integrates federated learning with a distributed ledger to address security and trust concerns in collaborative machine learning. The[1] core of the framework utilizes[59] federated learning to train a global model by iteratively aggregating local gradient models contributed by[59] various participants. A[1] notable feature of FeDis is its ability to handle heterogeneous data and devices, a common challenge in real-world federated systems. FeDis[1] enhances security and trust through the strategic use of[60] a distributed ledger. Participants'[1] local model weights are encrypted before being saved to the ledger. This[1] encrypted data is then fetched by the server[61] for aggregation to form the global model. This[1] process adds a layer of traceability and security, which is[62] intended to increase user trust in the system \cite{refre19}\\[62].

DFL is blockchain[63] architecture designed to improve the efficiency of federated learning (FL). Instead[1] of serving as a traditional distributed ledger, this blockchain[43] functions as a distributed proof of contribution to the machine learning (ML) model. A[1] key innovation is the[64] asynchronous block generation capability[64], allowing[64] each node to create its own blocks independently. This[1] approach significantly enhances the overall FL efficiency. The[1] architecture's stability and robustness are also addressed through a node reputation strategy and a weighted FedAvg implementation. These[1] features are particularly effective in mitigating the challenges of non-I.I.D. ([1]non-independent and identically distributed) datasets and defending against model poisoning attacks. From[1] an ML perspective, the primary contribution is an asynchronous, Gossip-based ML training method that uses the blockchain[43] specifically for storing[65] a verifiable proof of each node's contribution to the model \cite{refre20}.

Another study introduces a secure Federated Learning (FL) algorithm called MPCFL. MPCFL[1] is built on the principles of secure multi-party computation (MPC) and secret sharing. The[1] algorithm uses the Sharemind MPC framework to securely combine local model updates into a global model[66]. MPCFL[1] is designed to address common FL security issues, such as inference attacks, gradient leakage attacks, model poisoning, and model inversion \cite{refre21}.

The integration of blockchain[43] and FL enables the development of decentralized architectures with fortified[67] security, improved privacy preservation, and improved[67] reliability that is[67] particularly valuable in untrusted environments such as IoT networks that may contain adversarial nodes \cite{refre14}. HashGraph[1] delivers these same benefits while offering faster finalization and greater throughput.

The Swirlds HashGraph consensus mechanism provides a solution for replicated state machines, specifically designed to ensure Byzantine fault tolerance. Its[1] fairness characteristic is particularly notable, as adversaries find it extremely difficult to manipulate transaction ordering during the consensus process. The[1] system operates asynchronously without leaders, round-robin structures, or proof-of-work mechanisms, achieving eventual consensus with probability one while maintaining high performance in fault-free conditions.

At its core, the system utilizes a gossip protocol where participants do more than simply[68] broadcast transactions - they also share information about the gossiping process itself. Through[1] this process, participants collectively construct a HashGraph that records all gossip exchanges. This[1] structure enables Byzantine agreement through a "virtual voting" mechanism within the HashGraph consensus.

In this system, Alice doesn't directly send votes to Bob. Instead[1], Bob determines Alice's hypothetical vote[69] based on his knowledge of what Alice knows, gathered during the gossip process. This[1] innovative approach achieves fair Byzantine agreement on a comprehensive transaction order while maintaining minimal communication overhead beyond the transactions themselves. Operating[1] asynchronously, HashGraph establishes consensus without predefined paths. Its[1] probabilistic nature virtually guarantees that Byzantine agreement will always be reached \cite{refre17}.

Figure 2 illustrates a HashGraph consensus sample involving five nodes, along[70] with[70] the finalization of its events (blocks), while Figure 3 outlines the key components of the algorithms. The[1] gossip history is visualized as a directed graph, where the progression of the[71] gossip protocol depends on a graph-based representation[71]. In[1] this representation, each participant is mapped to a distinct vertical column of vertices (nodes). For[1] example, if node Alice receives gossip from node Bob including all of Bob's knowledge, this exchange is modeled and stored as a vertex in Alice's column. This[1] vertex then generates two downward-pointing edges, connecting it to the most recent prior gossip instances from both Alice and Bob.\\

HashLearner employs HashGraph as its consensus mechanism, eliminating the need for a central server (or cloud) in the federated learning process. A[1] central server could act as an adversary or attacker[72], introducing vulnerabilities such as backdoor planting (poisoning) or corrupting model performance. To[1] mitigate this, HashLearner introduces a validation step during aggregation, performed by multiple replicated multi-organizational nodes—similar to blockchain[43] through[73] a voting mechanism.

The system is fault-tolerant, which means that[74] it can withstand adversarial behavior as long as no more than half (50%) of the randomly chosen validators are malicious in any given aggregation step. If[1] more than half of the validators are adversaries, the voting process fails.

To improve aggregation performance especially[75] for resource-constrained IoT nodes with limited computation power, we introduced a sharding step. This[1,76] reduces the need for each node to aggregate large numbers of models simultaneously, enhancing efficiency and benefiting low-memory nodes in the federated network. The[1] primary weakness is that the system can fail if at least 33\% of the nodes are adversarial. Although[1] the aggregation process itself tolerates up to 50\% malicious voters, the underlying HashGraph consensus is aBFT (asynchronous Byzantine Fault Tolerant) and can only resist up to 33\% adversarial nodes. However,[1] due to the[77] asynchronous nature of HashGraph,[77] controlling virtual voting or executing cheating attacks is inherently difficult.

The key strength of this approach is its ability to remove reliance on an untrusted central server, enabling a masterless mesh of servers owned by different organizations to collaboratively train AI models without oversight from a central authority.[78] A[1] similar approach is now being adopted in AI training processes implemented on blockchains.

As mentioned in \cite{refre03} and \cite{refre0018}, FL is the[79] technique to overcome the[79] data isolation issue.[79] In[1] this way, it is possible to build models using the data contributed by users across the network. Yet[1] another key factor is the personalization of all the applied solutions. Even[1] if we can directly use the cloud model, it might not perform well on[80] a particular user. The[1] reason for this is the dissimilarity[81] in the data distribution of[81] the user and the server's storage. The[1] common model is trained on the server that[83] only[82] recognizes the distribution of the[83] features from[83] all the[83] users. Hence[1], it performs badly in recognizing[84] the details of specific users \cite{refre03}.

Transfer learning has been successfully applied in frameworks such as FedHealth, where trained models are adapted to local datasets to improve personalization and efficiency. HashLearner[1] builds on this idea by using the same dataset and CNN model, but eliminates the[85] reliance on a central server. Alternative[1] decentralized approaches have also been explored, most notably gossip learning. In[1] gossip learning, nodes exchange model updates randomly with peers, enabling fully decentralized training without central coordination. While[1] this design offers scalability and robustness, it suffers from slow convergence and the absence of a unique global model—each node retains its own final model, which only converges in distribution. Moreover[1], gossip learning cycles are asynchronous, and node selection relies on a sampling service \cite{refre04, refre09, refre10}. HashLearner[1] addresses these limitations by introducing a decentralized aggregation mechanism that preserves[86] scalability and robustness while ensuring[86] faster convergence and a shared global model without requiring[86] a central server.

In this paper[87] we build on these advancements by proposing a Secure Decentralized Learning Framework Based on HashGraph which is[87,88] called HashLearner, which[88] eliminates the central server in federated learning and leverages HashGraph's consensus mechanism to coordinate model updates. By doing so, our framework addresses the limitations of traditional FL systems, such as vulnerability to GAN attacks and single points of failure, while maintaining the privacy and scalability benefits of decentralized learning. The[1] integration of HashGraph ensures secure and efficient aggregation of model updates, even in adversarial environments, paving the way for a more robust and trustworthy federated learning paradigm.

Materials and Methods

In this section, we first introduce the dataset used in this paper,[89] [90] and then describe the details of the base CNN classifier that is[90] trained on each node. Afterwards,[1] The[92] main[91] novelty of the presented work[91] named[92] HashLearner[91] is described in detail.

Dataset Description

Similar to FedHealth, we use a dataset on Kaggle[93] \cite{refre18} referred to as[93] "human activity recognition with smartphones". For[1] this dataset preparation, a study was conducted involving 30 participants aged 19 to 48. The[1] subjects engaged in six different activities: walking on level ground, ascending stairs, descending stairs, sitting, standing, and lying down. Throughout[1] these activities, they wore a Samsung Galaxy S II smartphone secured at their waist. The[1] device's built-in accelerometer and gyroscope collected data on 3-axial linear acceleration and 3-axial angular velocity at a steady 50Hz frequency. To[1] ensure accurate data labeling, all sessions were recorded on video. The[1] resulting dataset was then randomly split,[94] with[94] 70\% allocated[94] for training purposes[94] and the remaining[94] 30\% reserved[94] for testing. Data[1] from the smartphone sensors underwent initial processing to reduce[95] noise. The[1] signals were then divided into fixed-width sliding windows of 2.56 seconds, with a 50\% overlap between consecutive windows (resulting in 128 readings per window). A[1] Butterworth low-pass filter was employed[96] to separate the acceleration signal into its body motion[96] and gravitational components. Given[1] that gravitational force is primarily associated with low-frequency signals, a filter with a[97] 0.3 Hz cutoff was applied. For[1] each window, a set of features was extracted by computing various time and frequency domain[98] variables. For[1] a fair comparison with FedHealth, we similarly extracted 5[99] topics from the mentioned dataset (i.e.[100] subject IDs 26 to 30) and assumed[101] them as the[101] isolated customers (corporations or nodes) which[101] cannot[100] share records due to privacy security[101]. Each[1] node has about 400 training samples. Also[1,102] it is worth mentioning that the dataset we used is a raw[102] non-preprocessed dataset which[102] is harder to train and[102] it[103] shows the power of the proposed approach and[103] is a harder task in comparison[103] to visual datasets like[102] some image processing prepared[103] datasets.

The proposed method

As mentioned in Introduction[104], the main purpose of this study is proposing[104,105] a novel approach for removing[105] the cloud server in[105] FedHealth to increase security without loss of[105] performance. For[1] fair comparisons with FedHealth, we should use a shallow CNN classifier just like the[107] FedHealth[106] for each node to classify six categories for[107] human activity recognition[107]. However,[1] the main contribution of this paper[108] named[109] HashLearner[108] is described in section 4-2-2. HashLearner[1] completely removes the cloud server in[110] FedHealth in a decentralized manner to avoid attacks on the server[110]. The[1] details of the aforementioned methods are explained in the following subsections. The[1] rest of this section of the paper presents the required[111] details needed[111] for the implementation of[111] the HashLearner platform. Since[1] this paper is based on an industrial software project[112] the source code of this work is also available at \cite{refre12}.

The Base CNN classifier

The deep learning model used in this paper is a Convolutional Neural Network (CNN) which[113] is a good option for extracting local data[114] relations between data[114] parts in[114] the input of the neural network[114]. For[1] fair comparisons, we adopted the same CNN classifier in FedHealth as shown in Figure 4. This[1] architecture comprises convolution, pooling, and fully connected (dense) layers. So[1], the system[116] after[115] the[116] training of[116] the model[115] freezes the convolution[116] and pooling layers and transfers the features of the local dataset to the global model more effectively by using transfer learning and training the fully connected layers placed[116] at the end of the deep learning model. Hyperparameters[1] of the base CNN classifier are summarized in Table 1.

HashLearner

The proposed approach in this research is named HashLearner which employs HashGraph and secure decentralized random voting to create a secure[117] and consistent way to share local models and aggregated model in multi-step aggregation (the number of steps depends on the number of nodes as the main purpose of the multi-step aggregation is to distribute the computation cost as a tradeoff[118] with security of the aggregation process). The[1] training phase of the proposed HashLearner algorithm follows the steps in Figure 5. Now[1], each step in this diagram is described in detail.

(1) Generate an initial model with random weights on each node separately as an initial global model.

(2) Each node submits its global model to HashGraph in a decentralized manner. Using[1] HashGraph in this step helps the network to prevent adversary nodes from corrupting the sharing process.

(3) Do an election between nodes to generate a shared[119] consistent[119] random[119] ordered array of node ids[119].

(4) Each node merges models existing on the HashGraph by merging[120] layers of nodes based on the elected order of nodes which[121] leads to a single unique global model (containing collaboratively random[121] generated weights) on all nodes. So,[1][122] now all nodes have the same global model to do the training[122] on.

(5) Each node trains the global model on its local dataset resulting[123] in a local[124] model. Then,[1] it submits its local model to the HashGraph.

(6) After all local models exist on the HashGraph, nodes do another election. The[1] result of the election is another shared random ordered array of node ids.[125] The[1] models self-shard the node list based on each node position in the produced order (e.g.[126] if node count is 100[126] we can have 4[127] shards of 25 grouped nodes each shard as a subarray of the shared order. [129][1] 0 to 24[128] : shard1, 25 to 49: shard2, etc[129]).[1] After the[132] sharding, in each shard[130] the first 3[131] nodes in the shared order are considered as[132] shard aggregators which[130] will validate each other's work proof.[132] Each[1] shard aggregator picks local models of nodes in its shard and aggregate[133,134] them simultaneously (in this research's implementation, averaging is employed[134] for aggregation but[133] any other aggregation can be used in[134] the shard). Afterwards,[1] all aggregators of all shards[135] submit their produced[135] shard models to the HashGraph.

(7) Nodes do another election to produce a shared random ordered array of node ids[136] and specify the first 3[137] nodes in the array as the global aggregators. Global[1] aggregators pick[138] all shard models of[138] all shard nodes from[138] the

HashGraph. They[1] compare shard models of each shard simultaneously and pick the most voted one (the most repeated model in the list of shard models of each shard). Then[1] they aggregate the elected shard models (each shard has one elected shard model). They[1] submit the aggregated global model to the HashGraph simultaneously.

(8) All nodes pick the aggregated global models and choose the most repeated one (most[140] voted) locally and separately as the global[141] model.

(9) Repeat steps 2 to 8 for any number of rounds for more convergence of the model. It[1] is notable that[142] training on local[143] dataset in each round applies transfer learning to the system helping[144] the global model to adapt to distribution[145] of different local datasets

Note: In this specific example[146] the sharding hyper-parameters such as shard size, shard aggregator count and[147] global aggregator count are not optimized for security and[148] aggregator count in shard and global aggregation must be more than 2[149] nodes but[150] the algorithm remains the same and[151] just changing this parameters[152] in settings[153] of the system will do the whole work needed for optimization and increasing security. So[1], there is no need for any[156] change in[156] the software code to increase security of[154,156] the algorithm and[154] more nodes can be adapted to this algorithm easily just[156] by changing the config object containing the 3[155] mentioned hyper-parameters[154].

This algorithm guarantees that all global models on[157] all nodes are the same[157] after the execution of[157] these steps. The[1] election of producing a shared[158] random[158] ordered array of node ids[158] uses the "commit & reveal scheme".

In commit[159] & reveal scheme:

(1) Nodes generate a random number, then they hash their number.

(2) They share their produced hash number.

(3) Then they share the produced random number.

(4) Each node validates the election by comparing the random number and its hash for each of the peers. If[1] all hashes match their random numbers[160] then the election is validated.

(5) Each node does XOR[161] operation on the group of random numbers and produces a random number. The[1] election guarantees that all nodes now have the same number.

(6) Finally[162] each node scales the produced number from a range of maximum number with the same bit count as the number of nodes to the number of nodes which[162] results in an index in range[162] 0 to node_count[162] - 1.

(7) Now nodes[163] consider the node with the same index in the nodes list as the elected node for this round and append its id[163] to the shared random[163] ordered array of node ids[163].

(8) Repeat the steps[164] 1 to 7 for node_count times to produce a shared random order of nodes as the election result for different use cases in this system.

As mentioned in the Dataset section, we have conducted a specific setup to show the ability of[166] the proposed method[165] in which[166] the node count wasn't high[166] so[165] that the shard size and aggregator count was[165,166] the same. But[1] the most functionality of the HashLearner network is in a high node count environment (e.g 100[167] nodes with 4 or 5 shards of 25 or 20 nodes in each shard) resulting[168] in distribution of aggregation in different shards on a nodes[169] network in[170] which each node has limited resources (due to limited resources[171] they can not aggregate all local models on their devices.)[172]

If we assume we have 5[173] nodes: Node A, Node B, Node C, Node D, Node E, each with a subject as local[174] dataset, after the shard election[174] they were mapped to indices 0 to 4 as Node 0, ..., 4. Table[1] 2 shows how the[176] sharding and aggregator[175] choosing happened[176].

It is worth mentioning[177] that GAN-based attacks aimed at reconstructing models can be prevented by shuffling peer layers of models between nodes in a peer-to-peer manner, one by one[177]. After[1] sharding is completed[178], each node randomly selects another node and exchanges a specific layer of its model with them.[178] This[1,179] ensures that while the models are shuffled, the aggregation result remains unchanged, and the model's integrity is preserved. Additionally[1], poisoning the model via GAN-based attacks is ineffective because aggregation is validated through consensus—the process is replaced, and the final result is determined by majority vote.

Results and Discussions

We used a well-known classification performance measure called[180] accuracy for comparing the[180] FedHealth and HashLearner. Accuracy is the percentage of correct predictions made by a machine learning model out of the total number of predictions.

In this experiment, we have run the FedHealth and HashLearner methods 20 times, separately. The[1] accuracy of the final global model for FedHealth and HashLearner is 92.26\% ± 0.98 and 93.57 ± 0.51, respectively. It[1] is notable that accuracy[181] of HashLearner[182] almost remained[182] the same in spite of[182] removing the

FedHealth cloud server and[181] the small changes in the numbers is[181,182] due to some randomness of[182] calculations and model generations and means[181,182] the performance of models[182] remained the same (not improved and not[182] corrupted). The[1] accuracy, precision, recall and[183] f1-score of each 5[183,184] subject nodes are shown in Table 4. As[1] shown in the table, the mean accuracies of[186] all subjects are so[186] close which[185] demonstrates[186] the convergence of the global model on[186] each node. In addition to the above experiment, we statistically analyzed the difference of[18] competing methods using a paired t-test. For[1] using the paired t-test, there is a precondition[188] that both random variables should[188] follow the[188] normal distribution. Therefore,[1] we need an additional statistical test[189] named[190] Anderson-Darling[189] test for[189,190] the null hypothesis that the accuracies obtained for FedHealth and HashLearner are from normal distributions. After[1] applying Anderson-Darling[191] test, the null hypothesis did[191,192] not reject[191] with[191] alpha level[191,192] 5\%.[191] Therefore,[1] the precondition for applying[193] the paired t-test is satisfied. The[1] null hypothesis for paired[194] t-test is that the accuracy of HashLearner and FedHealth comes from independent random samples from normal distributions with equal means and equal but unknown variances. After[1] applying this test, the null hypothesis is rejected with alpha level[195,196] [195]5% and p-value[195] 5.6e-6[195] means[195,196] that the accuracy of our proposed HashLearner is statistically better than FedHealth.

Performance Analysis of the Proposed Method

The experiments in our study are based[197] on a Google Colab virtual machine using[197] a T4 GPU with[197] 12 GB RAM. It[1] is provable that the convergence speed of the proposed approach in this paper is the same as that of typical federated learning methods (e.g., FedHealth) because the only modification from basic federated learning is the shuffling process. This[1] shuffling does not alter the median (or average) aggregation result, as the same layers are exchanged peer-to-peer between nodes. Sharding[1] merely distributes the computational process across nodes without affecting the final result. Mathematically[1], the median of 10 numbers can be calculated either directly or by splitting them into two shards of 5 numbers each, first computing the median of each shard and then taking the median of those two results. The[1] outcome remains identical.

Furthermore, the HashGraph consensus and voting mechanism does not modify the calculation itself; it only secures the process. Thus[1], there is no difference between traditional federated learning aggregation and HashLearner's aggregation in terms of results. However[1], HashLearner eliminates the need for a central server (third-party moderation), making it more secure and suitable for decentralized training across multiple organizations or communities without relying on a central authority.

At each round of training[198] an approximate time of 3~4 seconds had been[199] added as[199] the election, gossiping and[198] multi-step aggregation phase[199] in a simulated local environment.

Although in a practical environment model[200] serialization can be a bottleneck for the node communications and the transfer of models between nodes, multiple solutions can be employed to solve this problem such[201] as:

Parallelized serialization (as the target object is a list of matrices).

Parallelized model network transfer using multiple connections and channels

These 2[202] approaches can lead to lower model[204] transfer time over the network[204] and a faster HashGraph model submit[203,204] . It[1] is worth mentioning[206] that time[205] taken[206] to finalize blocks in a HashGraph will increase[206] logarithmically relative to[206] the number of nodes participating in the learning process. Using[1] sharding will increase both the[207] consensus speed[207] and the[207] aggregation speed as it decreases[20] the number of nodes participating in the consensus and also[207] the number of models being aggregated. It[1] is worth mentioning[212] that decreasing[208,209,212] the number of nodes in the consensus will do[212] a trade-off[118] between speed and security which[210] means[212] that it will decrease[211,212] security to increase speed. Larger[1] shards are slower to finalize block[214] and aggregate models but[213] are more secure as more nodes participate in consensus and[213] more validators can be chosen for aggregating the[214] models in[214] the shard.

Scalability and throughput: Due to Swirlds[215] HashGraph paper proofs, the HashGraph consensus has higher throughput than other multi party[215,216] computation or Blockchain alternatives. Also[1,217] HashGraph makes[218] peers able[218] to submit blocks concurrently which[217] increases scalability efficiently[218]. Also[1,219] the sharding algorithm proposed by[220] our paper distributes load and computation bottlenecks and improve[219,220] scalability compared to other papers mentioned in related work.

HashLearner has higher throughput than FeDis \cite{refre19} and DFL \cite{refre20} because instead[221] of Blockchain[43], it relies on hashgraph with asynchronous gossiping and block finalization and[221] uses sharding to distribute computation[222] load across the network at the same time[222]. In[1] ideal[223] situation[224] HashLearner can average models in less than 10 seconds for a single round averaging (inside shard or globally) and[225] in real word[226,227] situation due[228] to network latency, this process can be degraded but[229] it is worth mentioning that this degradation is being applied to all approaches including[230] Blockchains and there[231][2] is always a gap between ideal and real statistics. Also[1,233] HashLearner outperforms MPCFL \cite{refre21} in throughput and also[234] provides higher security as it uses a distributed ledger and higher[233] level of verification.

Security and Privacy Analysis of the Proposed method[235]

Federated learning is designed to enable[236] collaborative model training across distributed nodes without sharing raw data. However[1], the involvement of an honest but curious cloud server, a server that follows the protocol correctly but may attempt to infer sensitive information from the shared data, introduces significant security risks. As[1] mentioned earlier[237] the work presented in \cite{refre03} is vulnerable to three critical attack vectors: GAN attacks, reconstruction attacks, and poisoning attacks. HashLearner[1] Uses[238,239] a decentralized consensus mechanism (HashGraph) to ensure secure and[239] transparent model aggregation and reduces[239] the risk of single-point failures and attacks targeting a central authority. In[1] this part of this paper, a formal security analysis of HashLearner[240] platform is drawn:

In this part of the paper[241] we formalize the security guarantees of the proposed method under standard cryptographic frameworks. We[1] consider a decentralized federated learning system consisting of a set of nodes $\mathcal{N} = \{P_1, P_2, \dots, P_n\}$, where each node P_i holds a private dataset D_i. The goal is to collaboratively train a global model M without exposing raw data. Each[1] training round proceeds as follows:

Each node trains a local model M_i on D_i.

Updates are obfuscated using \textit{peer-to-peer shuffling}.

Nodes broadcast updates through the HashGraph gossip-about-gossip protocol.

HashGraph consensus determines the final ordering of events.

Aggregation produces the updated global model.

We assume a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ controlling up to $t < n/3$ nodes, consistent with the fault tolerance of HashGraph consensus. The[1] adversary may attempt:
Reconstruction attacks: Inferring private data from shared updates (e.g., gradient inversion, GAN-based attacks).
Poisoning backdoor attacks: Injecting manipulated updates to degrade accuracy or embed targeted misclassifications.
Message manipulation: Delaying, reordering, or dropping messages to bias consensus.
Honest-but-curious behavior: Following the protocol while attempting to extract information from others' updates.

The adversary is computationally bounded, and communication channels are authenticated but not confidential. We define the ideal functionality $\mathcal{F}_{HL}$ for HashLearner:[1]
Inputs: Each party P_i provides its local update M_i.
Process:
$\mathcal{F}_{HL}$ applies randomized shuffling to anonymize updates.
An incorruptible consensus oracle determines the order of updates.
Secure aggregation produces the global model M.
Outputs: The global model M is delivered to all parties.

In the ideal world, adversaries see only their own inputs and the final global model. But[1] real-world execution of HashLearner:
Shuffling is implemented through peer-to-peer exchanges.

Consensus is achieved via HashGraph's gossip-about-gossip with virtual voting.

Updates are broadcast and aggregated in the agreed-upon order.

Here, adversaries may observe update flows and corrupted nodes' contributions[242].

Security Argument Based on Ideal--Real Simulation

Assuming authenticated channels and the Byzantine resilience of[243] HashGraph consensus, HashLearner securely realizes $\mathcal{F}_{HL}$ against any PPT adversary controlling fewer than $n/3$ nodes. This[1,244] is true since:

Consensus integrity: HashGraph achieves asynchronous Byzantine fault tolerance; thus, adversaries cannot bias the ordering of honest updates except with negligible probability.

Confidentiality of updates and reconstruction attacks: Randomized shuffling ensures adversaries cannot link specific updates to honest nodes and infer[246] [245] private data except with negligible probability.

Simulatability: Any adversary's view in the real world (messages, corrupted updates, final model) can be simulated in the ideal world using only the global model and corrupted inputs. Hence[1], the distributions are indistinguishable.

Poisoning resistance: Because aggregation requires consensus, adversarial influence is bounded, preventing domination of the global model. also[1,247] it ensures that no single node can manipulate the aggregation result.

Resource-efficiency balanced with security: Nodes with limited resources such[2] as IoT nodes[248] can easily participate without overburdening a central server, as the aggregation workload is securely shared.

Peer-to-peer communication: Reduced latency and improved scalability compared to centralized systems and[249] removed central[249] server leading[249] to elimination[249] of central[249] communication and data transfer proxy that could be insecure and vulnerable to data manipulation.

Transparent and auditable: HashGraph's consensus mechanism ensures that all transactions such[250] as model updates[250] are recorded and verifiable, reducing the risk of malicious behavior of nodes and[250] HashLearner can easily act as intrusion[250] detection system IDS confronting malicious nodes in poisoning attacks.

Mitigates GAN attacks: By distributing the aggregation process and obfuscating local updates, HashLearner reduces the risk of data leakage.

HashGraph consensus security: HashGraph is fault-tolerant against up to 33\% adversarial nodes, as it is an \textit{asynchronous Byzantine Fault-Tolerant (aBFT)} protocol—the highest security level for asynchronous consensus. This[1] security comes with the added benefit of high throughput, enabling HashGraph to finalize multiple blocks simultaneously while outperforming synchronous blockchains in speed.

Therefore, HashLearner is secure in the ideal--real paradigm.

Privacy Leakage Analysis}

Although complete privacy cannot be guaranteed in gradient-based learning, we analyze[251] the reduction in leakage using empirical evaluation[251].

Baseline FL (FedAvg): it[252,253] is vulnerable to membership inference attack[252,253] and has the lowest privacy in this comparison. its[1,254,255] core principle is that clients send only their model updates, not their raw data, to a central server. However[1], these updates, which are essentially averaged gradients, contain information about the data used to calculate them. Attackers[1] can exploit this by using various techniques to reverse-engineer the updates and infer private information \cite{refre22}.

Blockchain-based FL: The use of immutable ledgers enhances high[256] level of privacy defense but[256] because no native obfuscation is applied, model weights recorded on the ledger remain exposed to potential inference attacks. In[1] other words, in many blockchain-based FL designs, the encrypted or unencrypted model updates (gradients) are stored on a public ledger. Even[1] if a third party cannot manipulate the data, they can access it. An[1] adversary can analyze these publicly available gradient updates over time to perform attacks such as gradient inversion attacks[257] or membership inference attacks[257], similar to those in traditional FedAvg. Also[1,258] a blockchain-based system cannot inherently prevent participants from colluding. If[1] a malicious client colludes with a miner node on the blockchain[43], they can still leak sensitive information. For[1] example, a group of clients can aggregate their gradients and share them with an outside party, or a malicious client can submit malicious updates to reconstruct another client's data \cite{refre23}.

HashLearner: The privacy of shuffling[259] system is close to random guessing, due to peer to peer[259] shuffling of the model weights and consensus-driven aggregation. This[1] shuffling process hides the identity of individual local models. Even[1] though the weights are shuffled, the final result is the same as if they had not been, because a non-weighted average is used to combine the models, which does not affect the final[260] outcome. This[1,261] is because only the[262] peer layers (e.g., the third layer of one model with[262] the third layer of another) are shuffled with each other[262]. The[1] shuffling makes it impossible for someone to link specific data to a particular model. It[1] is worth mentioning[263] that a higher number of[263] nodes and a higher number of[263] layers in a deep learning model increase shuffling security and make it harder for an adversary to identify the real[263] models and infer data. Privacy[1] is inherently preserved by the decentralized nature of HashGraph and the[265] multi-step aggregation process based on a[265] probabilistic process,[264,265] hence[264] No[264,265] reliance on a trusted third party is achieved[265].

Thus, HashLearner significantly reduces the adversary's advantage while maintaining model utility. This[1] framework offers significant advantages over traditional cloud-based FL systems in terms of security, privacy, scalability, higher throughput, transparency and[266] validation. Also[1,267] it eliminates[268] risks associated with a single point of failure and adversarial behavior. this[1] points[269,27] and properties makes[269] HashLearner a highly secure and scalable solution for federated learning. In[1] contrast, cloud-based federated learning systems remain vulnerable to attacks targeting the central server and require additional privacy-preserving measures to mitigate risks. HashLearner's[1] decentralized approach represents a paradigm shift in federated learning, addressing the limitations of traditional systems and paving the way for more secure and privacy-preserving collaborative learning frameworks.

Conclusion

In this work, we introduced HashLearner, a secure and decentralized federated learning framework that eliminates the reliance on a central server by leveraging the HashGraph consensus protocol. The[1] framework integrates three key components: (i) decentralized aggregation via Byzantine fault-tolerant consensus, (ii) a peer-to-peer shuffling mechanism that obfuscates local model updates and mitigates reconstruction attacks, and (iii) a transfer learning–based personalization method to adapt the aggregated global model to heterogeneous client datasets. Our[1] evaluation demonstrates that HashLearner maintains accuracy comparable to conventional FL while significantly improving robustness against poisoning and inference attacks, reducing privacy leakage, and scaling efficiently in distributed environments. These[1] findings indicate that HashLearner provides a practical pathway toward[271] trustworthy and[271] scalable federated learning in adversarial and resource-constrained settings.

1.	*. To; . HashLearner; . The; . These; . Some; . By; . They; . Despite; . This; . A; . GAN; . Section; . Finally; . Gossip; . In; . After; . Hence; . Poisoning; . One; . As; . Reconstruction; . On; . Multiple; . Blockchain-based; . However; . HashGraph; . Unlike; . Its; . Moreover; . Generally; . It;...*	Text inconsistencies	Correctness
2.	, and	Comma misuse within clauses	Correctness
3.	~~demonstrate~~ → indicate	Word choice	Engagement
4.	*The deep learning community increasingly seeks decentralized solutions that allow organizations to jointly train models and carry out computations while operating in untrusted environments.*	Paragraph can be improved	Clarity
5.	, including	Incorrect punctuation	Correctness
6.	*The increasing demand for privacy-preserving machine learning has advanced the rise of FL, where multiple organizations collaboratively train a shared model without exchanging raw data.*	Paragraph can be improved	Clarity
7.	*They try to distribute the training phase through a decentralized network of nodes which is multi-organizational which means each node can have a different owner than other nodes.*	Ungrammatical sentence	Correctness
8.	*They try to distribute the training phase through a decentralized network of nodes which is multi-organizational which means each node can have a different owner than other nodes.*	Paragraph can be improved	Clarity
9.	*By decentralizing the training process, FL addresses critical privacy concerns and reduces the risks associated with data centralization.*	Paragraph can be improved	Clarity

10.	. In contrast, blockchain-based	Hard-to-read text	Clarity
11.	*This paper proposes a Secure Decentralized Learning Framework Based on HashGraph called HashLearner, which eliminates the dependency on a central server and leverages the unique properties of HashGraph, a distributed ledger technology known for its high throughput, fairness, and Byzantine fault tol...*	Paragraph can be improved	Clarity
12.	*The decentralized nature of HashGraph not only enhances the resilience of the system but also mitigates the risks associated with centralized control, such as data manipulation and unauthorized access.*	Paragraph can be improved	Clarity
13.	*By removing the central server and leveraging HashGraph's decentralized architecture, the proposed approach paves the way for more secure and privacy-preserving machine learning in distributed environments.*	Paragraph can be improved	Clarity
14.	, and	Comma misuse within clauses	Correctness
15.	*The rest of the paper is structured as follows: In Section 2, related work in this area is reviewed.*	Paragraph can be improved	Clarity
16.	, and	Incorrect punctuation	Correctness
17.	the Decentralized	Determiner use (a/an/the/this, etc.)	Correctness
18.	*In gossip learning, each node trains its local model using its local dataset the same as FL but the point is that there is no global initial model and each node initializes the global model's initial weights independently.*	Ungrammatical sentence	Correctness

19.	*In gossip learning, each node trains its local model using its local dataset the same as FL but the point is that there is no global initial model and each node initializes the global model's initial weights independently.*	Paragraph can be improved	Clarity
20.	*After training local models, each node tries to exchange its model's weights with some other nodes in multiple rounds and at each round it merges its model with the other node's model.*	Ungrammatical sentence	Correctness
21.	*After some rounds step by step the models can be converged more to some same distribution across multiple nodes.*	Ungrammatical sentence	Correctness
22.	*After some rounds step by step the models can be converged more to some same distribution across multiple nodes.*	Paragraph can be improved	Clarity
23.	Hence,	Incorrect punctuation	Correctness
24.	*Hence GL might not be a good target in heterogeneous networks with varying computational resources and communication delays because of the lack of a structured coordination mechanism and inconsistencies in model convergence \cite{refre11}.*	Paragraph can be improved	Clarity
25.	*In FL the target is to train a high quality centralized shared model, while training data is disbursed in a large number of each customer with an untrustable and relatively slow network connection under the coordination of a central server.*	Ungrammatical sentence	Correctness
26.	*In FL the target is to train a high quality centralized shared model, while training data is disbursed in a large number of each customer with an untrustable and relatively slow network connection under the coordination of a central server.*	Paragraph can be improved	Clarity

27.	a reconstruction	Determiner use (a/an/the/this, etc.)	Correctness
28.	~~be used for~~ → target	Paragraph can be improved	Clarity
29.	*One of them is to corrupt the model by inserting wrong and corrupted data with a wrong data distribution into the dataset which is called passive attack and the other one is for hiding a backdoor inside the trained model by employing specific labels or data features in the dataset which is called a...*	Ungrammatical sentence	Correctness
30.	*One of them is to corrupt the model by inserting wrong and corrupted data with a wrong data distribution into the dataset which is called passive attack and the other one is for hiding a backdoor inside the trained model by employing specific labels or data features in the dataset which is called a...*	Paragraph can be improved	Clarity
31.	, such	Incorrect punctuation	Correctness
32.	*In the reconstruction attack, the adversary tries to reconstruct a shadow model based on the target model weights or gradients to find out some info about the data in the dataset in different ways such as dataset distribution, existence of specific data pieces in the dataset, or some properties of ...*	Paragraph can be improved	Clarity
33.	*As an example, the adversary can gain the ability to learn if an image is in the dataset, or know the number of people with some specific properties in the dataset.*	Paragraph can be improved	Clarity
34.	*On the other hand, the discriminator tries to find out the fake samples produced by the generator.*	Paragraph can be improved	Clarity

35.	~~which are~~ → that are	Pronoun use	Correctness
36.	*The competition between these two parts leads to learning more features and advancement of the generator to produce fake samples which are so similar to real samples.*	Paragraph can be improved	Clarity
37.	~~produce~~ → make, create	Word choice	Engagement
38.	, as	Incorrect punctuation	Correctness
39.	*In the reconstruction attack, the adversary can use GANs to produce a shadow model and find some info about the dataset of a node as mentioned before.*	Paragraph can be improved	Clarity
40.	*Multiple approaches as counter-attacks had been proposed against these attacks such as Encryption, Shuffling, Differential Privacy and in some ways inserting noise into dataset which were helpful to make the system more safe against some of the mentioned attacks \cite{refre05}.*	Ungrammatical sentence	Correctness
41.	*Multiple approaches as counter-attacks had been proposed against these attacks such as Encryption, Shuffling, Differential Privacy and in some ways inserting noise into dataset which were helpful to make the system more safe against some of the mentioned attacks \cite{refre05}.*	Paragraph can be improved	Clarity
42.	*To address these challenges, recent research has explored the integration of blockchain technology into FL, aiming to decentralize the aggregation process and enhance security.*	Paragraph can be improved	Clarity
43.	*blockchain; Blockchain*	Text inconsistencies	Correctness

44.	*HashGraph \cite{refre17} is a distributed ledger technology that utilizes a gossip-about-gossip protocol and virtual voting to achieve consensus in a highly efficient and secure manner.*	Paragraph can be improved	Clarity
45.	*HashGraph offers higher throughput than proof-of-stake blockchains and is therefore suitable for AI training networks in which multiple models are trained simultaneously.*	Paragraph can be improved	Clarity
46.	~~, and HashGraph's~~ → . Hashgraph's	Hard-to-read text	Clarity
47.	*HashGraph is a decentralized network architecture consisting of nodes each holding a copy of the HashGraph data including its events graph.*	Ungrammatical sentence	Correctness
48.	*HashGraph is a decentralized network architecture consisting of nodes each holding a copy of the HashGraph data including its events graph.*	Paragraph can be improved	Clarity
49.	*HashGraph can be used to store data in a secure and decentralized manner in which it is hard for attackers to corrupt its stability and consistency including the order of events in the HashGraph.*	Ungrammatical sentence	Correctness
50.	*HashGraph can be used to store data in a secure and decentralized manner in which it is hard for attackers to corrupt its stability and consistency including the order of events in the HashGraph.*	Paragraph can be improved	Clarity
51.	*This research tries to improve decentralized learning approaches to aggregate a global model in a decentralized manner.*	Paragraph can be improved	Clarity
52.	~~figure~~ → Figure	Confused words	Correctness
53.	~~the~~ transfer	Determiner use (a/an/the/this, etc.)	Correctness

54.	*They employed the transfer learning to adapt the global aggregated model to each node's local dataset data distribution by training the global aggregated model on the local datasets separately on each node \cite{refre03}.*	Paragraph can be improved	Clarity
55.	*In federated learning frameworks, the aggregator plays a crucial role by collecting and combining training results from local sources to update the global model.*	Paragraph can be improved	Clarity
56.	*To address these primary challenges and other related issues, they implemented blockchain technology to finalize and share models through an immutable distributed ledger similar to HashLearner's approach using HashGraph.*	Paragraph can be improved	Clarity
57.	*The FeDis framework, representing an approach that integrates federated learning with a distributed ledger to address security and trust concerns in collaborative machine learning.*	Ungrammatical sentence	Correctness
58.	which	Paragraph can be improved	Clarity
59.	*The core of the framework utilizes federated learning to train a global model by iteratively aggregating local gradient models contributed by various participants.*	Paragraph can be improved	Clarity
60.	*FeDis enhances security and trust through the strategic use of a distributed ledger.*	Paragraph can be improved	Clarity
61.	*This encrypted data is then fetched by the server for aggregation to form the global model.*	Passive voice misuse	Clarity
62.	*This process adds a layer of traceability and security, which is intended to increase user trust in the system \cite{refre19}\\.*	Paragraph can be improved	Clarity

63.	a blockchain	Determiner use (a/an/the/this, etc.)	Correctness
64.	*A key innovation is the asynchronous block generation capability, allowing each node to create its own blocks independently.*	Paragraph can be improved	Clarity
65.	to store	Paragraph can be improved	Clarity
66.	to combine local model updates into a global model securely	Inappropriate colloquialisms	Delivery
67.	*The integration of blockchain and FL enables the development of decentralized architectures with fortified security, improved privacy preservation, and improved reliability that is particularly valuable in untrusted environments such as IoT networks that may contain adversarial nodes \cite{refre14}.*	Paragraph can be improved	Clarity
68.		Tone suggestions	Delivery
69.	~~vote~~ → ballot	Word choice	Engagement
70.	*Figure 2 illustrates a HashGraph consensus sample involving five nodes, along with the finalization of its events (blocks), while Figure 3 outlines the key components of the algorithms.*	Paragraph can be improved	Clarity
71.	*The gossip history is visualized as a directed graph, where the progression of the gossip protocol depends on a graph-based representation.*	Paragraph can be improved	Clarity
72.	~~or attacker~~	Paragraph can be improved	Clarity

73.	*To mitigate this, HashLearner introduces a validation step during aggregation, performed by multiple replicated multi-organizational nodes—similar to blockchain through a voting mechanism.*	Paragraph can be improved	Clarity
74.	~~which means that~~ → meaning	Paragraph can be improved	Clarity
75.	, especially	Incorrect punctuation	Correctness
76.	*This*	Intricate text	Clarity
77.	*However, due to the asynchronous nature of HashGraph, controlling virtual voting or executing cheating attacks is inherently difficult.*	Paragraph can be improved	Clarity
78.	to train AI models without oversight from a central authority collaboratively	Inappropriate colloquialisms	Delivery
79.	*As mentioned in \cite{refre03} and \cite{refre0018}, FL is the technique to overcome the data isolation issue.*	Paragraph can be improved	Clarity
80.	~~on~~ → for	Wrong or missing prepositions	Correctness
81.	*The reason for this is the dissimilarity in the data distribution of the user and the server's storage.*	Paragraph can be improved	Clarity
82.	*The common model is trained on the server that only recognizes the distribution of the features from all the users.*	Ungrammatical sentence	Correctness
83.	*The common model is trained on the server that only recognizes the distribution of the features from all the users.*	Paragraph can be improved	Clarity
84.	~~recognizing~~ → identifying, remembering	Word choice	Engagement

85.	while eliminating	Paragraph can be improved	Clarity
86.	*HashLearner addresses these limitations by introducing a decentralized aggregation mechanism that preserves scalability and robustness while ensuring faster convergence and a shared global model without requiring a central server.*	Paragraph can be improved	Clarity
87.	*In this paper we build on these advancements by proposing a Secure Decentralized Learning Framework Based on HashGraph which is called HashLearner, which eliminates the central server in federated learning and leverages HashGraph's consensus mechanism to coordinate model updates.*	Ungrammatical sentence	Correctness
88.	*In this paper we build on these advancements by proposing a Secure Decentralized Learning Framework Based on HashGraph which is called HashLearner, which eliminates the central server in federated learning and leverages HashGraph's consensus mechanism to coordinate model updates.*	Paragraph can be improved	Clarity
89.	paper**,**	Comma misuse within clauses	Correctness
90.	*In this section, we first introduce the dataset used in this paper, and then describe the details of the base CNN classifier that is trained on each node.*	Paragraph can be improved	Clarity
91.	*Afterwards, The main novelty of the presented work named HashLearner is described in detail.*	Ungrammatical sentence	Correctness
92.	*Afterwards, The main novelty of the presented work named HashLearner is described in detail.*	Paragraph can be improved	Clarity

93.	*Similar to FedHealth, we use a dataset on Kaggle \cite{refre18} referred to as "human activity recognition with smartphones".*	Paragraph can be improved	Clarity
94.	*The resulting dataset was then randomly split, with 70\% allocated for training purposes and the remaining 30\% reserved for testing.*	Paragraph can be improved	Clarity
95.	*Data from the smartphone sensors underwent initial processing to reduce noise.*	Paragraph can be improved	Clarity
96.	*A Butterworth low-pass filter was employed to separate the acceleration signal into its body motion and gravitational components.*	Paragraph can be improved	Clarity
97.	*Given that gravitational force is primarily associated with low-frequency signals, a filter with a 0.3 Hz cutoff was applied.*	Paragraph can be improved	Clarity
98.	*For each window, a set of features was extracted by computing various time and frequency domain variables.*	Paragraph can be improved	Clarity
99.	~~5~~ → five	Improper formatting	Correctness
100.	*For a fair comparison with FedHealth, we similarly extracted 5 topics from the mentioned dataset (i.e. subject IDs 26 to 30) and assumed them as the isolated customers (corporations or nodes) which cannot share records due to privacy security.*	Ungrammatical sentence	Correctness
101.	*For a fair comparison with FedHealth, we similarly extracted 5 topics from the mentioned dataset (i.e. subject IDs 26 to 30) and assumed them as the isolated customers (corporations or nodes) which cannot share records due to privacy security.*	Paragraph can be improved	Clarity

102.	*Also it is worth mentioning that the dataset we used is a raw non-preprocessed dataset which is harder to train and it shows the power of the proposed approach and is a harder task in comparison to visual datasets like some image processing prepared datasets.*	Ungrammatical sentence	Correctness
103.	*Also it is worth mentioning that the dataset we used is a raw non-preprocessed dataset which is harder to train and it shows the power of the proposed approach and is a harder task in comparison to visual datasets like some image processing prepared datasets.*	Paragraph can be improved	Clarity
104.	*As mentioned in Introduction, the main purpose of this study is proposing a novel approach for removing the cloud server in FedHealth to increase security without loss of performance.*	Ungrammatical sentence	Correctness
105.	*As mentioned in Introduction, the main purpose of this study is proposing a novel approach for removing the cloud server in FedHealth to increase security without loss of performance.*	Paragraph can be improved	Clarity
106.	one used by FedHealth	Incorrect phrasing	Correctness
107.	*For fair comparisons with FedHealth, we should use a shallow CNN classifier just like the FedHealth for each node to classify six categories for human activity recognition.*	Paragraph can be improved	Clarity
108.	*However, the main contribution of this paper named HashLearner is described in section 4-2-2.*	Ungrammatical sentence	Correctness
109.	*However, the main contribution of this paper named HashLearner is described in section 4-2-2.*	Paragraph can be improved	Clarity

110.	*HashLearner completely removes the cloud server in FedHealth in a decentralized manner to avoid attacks on the server.*	Paragraph can be improved	Clarity
111.	*The rest of this section of the paper presents the required details needed for the implementation of the HashLearner platform.*	Paragraph can be improved	Clarity
112.	project,	Incorrect punctuation	Correctness
113.	, which	Incorrect punctuation	Correctness
114.	*The deep learning model used in this paper is a Convolutional Neural Network (CNN) which is a good option for extracting local data relations between data parts in the input of the neural network.*	Paragraph can be improved	Clarity
115.	*So, the system after the training of the model freezes the convolution and pooling layers and transfers the features of the local dataset to the global model more effectively by using transfer learning and training the fully connected layers placed at the end of the deep learning model.*	Ungrammatical sentence	Correctness
116.	*So, the system after the training of the model freezes the convolution and pooling layers and transfers the features of the local dataset to the global model more effectively by using transfer learning and training the fully connected layers placed at the end of the deep learning model.*	Paragraph can be improved	Clarity
117.	~~secure~~ → safe	Word choice	Engagement
118.	*tradeoff; trade-off*	Text inconsistencies	Correctness
119.	*(3) Do an election between nodes to generate a shared consistent random ordered array of node ids.*	Ungrammatical sentence	Correctness

120.	~~merging~~ → combining, integrating	Word choice	Engagement
121.	*(4) Each node merges models existing on the HashGraph by merging layers of nodes based on the elected order of nodes which leads to a single unique global model (containing collaboratively random generated weights) on all nodes.*	Ungrammatical sentence	Correctness
122.	*So, now all nodes have the same global model to do the training on.*	Paragraph can be improved	Clarity
123.	, resulting	Incorrect punctuation	Correctness
124.	~~local~~ → regional	Word choice	Engagement
125.	~~ids~~ → IDs	Confused words	Correctness
126.	*The models self-shard the node list based on each node position in the produced order (e.g. if node count is 100 we can have 4 shards of 25 grouped nodes each shard as a subarray of the shared order. 0 to 24 : shard1, 25 to 49: shard2, etc).*	Ungrammatical sentence	Correctness
127.	~~4~~ → four	Improper formatting	Correctness
128.	24 :	Improper formatting	Correctness
129.	*The models self-shard the node list based on each node position in the produced order (e.g. if node count is 100 we can have 4 shards of 25 grouped nodes each shard as a subarray of the shared order. 0 to 24 : shard1, 25 to 49: shard2, etc).*	Paragraph can be improved	Clarity
130.	*After the sharding, in each shard the first 3 nodes in the shared order are considered as shard aggregators which will validate each other's work proof.*	Ungrammatical sentence	Correctness
131.	~~3~~ → three	Improper formatting	Correctness

132.	*After the sharding, in each shard the first 3 nodes in the shared order are considered as shard aggregators which will validate each other's work proof.*	Paragraph can be improved	Clarity
133.	*Each shard aggregator picks local models of nodes in its shard and aggregate them simultaneously (in this research's implementation, averaging is employed for aggregation but any other aggregation can be used in the shard).*	Ungrammatical sentence	Correctness
134.	*Each shard aggregator picks local models of nodes in its shard and aggregate them simultaneously (in this research's implementation, averaging is employed for aggregation but any other aggregation can be used in the shard).*	Paragraph can be improved	Clarity
135.	*Afterwards, all aggregators of all shards submit their produced shard models to the HashGraph.*	Paragraph can be improved	Clarity
136.	~~ids~~ → IDs	Confused words	Correctness
137.	~~3~~ → three	Improper formatting	Correctness
138.	*Global aggregators pick all shard models of all shard nodes from the HashGraph.*	Paragraph can be improved	Clarity
139.	~~pick~~ → choose, select	Word choice	Engagement
140.	the most	Determiner use (a/an/the/this, etc.)	Correctness
141.	~~global~~ → worldwide	Word choice	Engagement
142.	~~It is notable that~~ → Notably,	Wordy sentences	Clarity
143.	a local	Determiner use (a/an/the/this, etc.)	Correctness

144.	, helping	Incorrect punctuation	Correctness
145.	the distribution	Determiner use (a/an/the/this, etc.)	Correctness
146.	example,	Incorrect punctuation	Correctness
147.	, and	Comma misuse within clauses	Correctness
148.	, and	Comma misuse within clauses	Correctness
149.	~~2~~ → two	Improper formatting	Correctness
150.	, but	Incorrect punctuation	Correctness
151.	, and	Incorrect punctuation	Correctness
152.	these parameters	Incorrect noun number	Correctness
153.	the settings	Determiner use (a/an/the/this, etc.)	Correctness
154.	*So, there is no need for any change in the software code to increase security of the algorithm and more nodes can be adapted to this algorithm easily just by changing the config object containing the 3 mentioned hyper-parameters.*	Ungrammatical sentence	Correctness
155.	~~3~~ → three	Improper formatting	Correctness
156.	*So, there is no need for any change in the software code to increase security of the algorithm and more nodes can be adapted to this algorithm easily just by changing the config object containing the 3 mentioned hyper-parameters.*	Paragraph can be improved	Clarity

157.	*This algorithm guarantees that all global models on all nodes are the same after the execution of these steps.*	Paragraph can be improved	Clarity
158.	*The election of producing a shared random ordered array of node ids uses the "commit & reveal scheme".*	Ungrammatical sentence	Correctness
159.	the commit	Determiner use (a/an/the/this, etc.)	Correctness
160.	numbers,	Incorrect punctuation	Correctness
161.	an XOR	Determiner use (a/an/the/this, etc.)	Correctness
162.	*(6) Finally each node scales the produced number from a range of maximum number with the same bit count as the number of nodes to the number of nodes which results in an index in range 0 to node_count - 1.*	Ungrammatical sentence	Correctness
163.	*(7) Now nodes consider the node with the same index in the nodes list as the elected node for this round and append its id to the shared random ordered array of node ids.*	Ungrammatical sentence	Correctness
164.	~~the~~ steps	Determiner use (a/an/the/this, etc.)	Correctness
165.	*As mentioned in the Dataset section, we have conducted a specific setup to show the ability of the proposed method in which the node count wasn't high so that the shard size and aggregator count was the same.*	Ungrammatical sentence	Correctness
166.	*As mentioned in the Dataset section, we have conducted a specific setup to show the ability of the proposed method in which the node count wasn't high so that the shard size and aggregator count was the same.*	Paragraph can be improved	Clarity
167.	, 100	Incorrect punctuation	Correctness

168.	, resulting	Incorrect punctuation	Correctness
169.	~~nodes~~	Incorrect noun number	Correctness
170.	of nodes in	Incorrect phrasing	Correctness
171.	resources,	Incorrect punctuation	Correctness
172.	~~.)~~ →).	Incorrect punctuation	Correctness
173.	~~5~~ → five	Improper formatting	Correctness
174.	*If we assume we have 5 nodes: Node A, Node B, Node C, Node D, Node E, each with a subject as local dataset, after the shard election they were mapped to indices 0 to 4 as Node 0, ..., 4.*	Ungrammatical sentence	Correctness
175.	the aggregator	Determiner use (a/an/the/this, etc.)	Correctness
176.	*Table 2 shows how the sharding and aggregator choosing happened.*	Paragraph can be improved	Clarity
177.	*It is worth mentioning that GAN-based attacks aimed at reconstructing models can be prevented by shuffling peer layers of models between nodes in a peer-to-peer manner, one by one.*	Paragraph can be improved	Clarity
178.	*After sharding is completed, each node randomly selects another node and exchanges a specific layer of its model with them.*	Paragraph can be improved	Clarity
179.	*This*	Intricate text	Clarity
180.	*We used a well-known classification performance measure called accuracy for comparing the FedHealth and HashLearner.*	Paragraph can be improved	Clarity

181.	*It is notable that accuracy of HashLearner almost remained the same in spite of removing the FedHealth cloud server and the small changes in the numbers is due to some randomness of calculations and model generations and means the performance of models remained the same (not improved and not corrup...*	Ungrammatical sentence	Correctness
182.	*It is notable that accuracy of HashLearner almost remained the same in spite of removing the FedHealth cloud server and the small changes in the numbers is due to some randomness of calculations and model generations and means the performance of models remained the same (not improved and not corrup...*	Paragraph can be improved	Clarity
183.	*The accuracy, precision, recall and f1-score of each 5 subject nodes are shown in Table 4.*	Ungrammatical sentence	Correctness
184.	~~5~~ → five	Improper formatting	Correctness
185.	, which	Incorrect punctuation	Correctness
186.	*As shown in the table, the mean accuracies of all subjects are so close which demonstrates the convergence of the global model on each node.*	Paragraph can be improved	Clarity
187.	~~of~~ → between	Wrong or missing prepositions	Correctness
188.	*For using the paired t-test, there is a precondition that both random variables should follow the normal distribution.*	Paragraph can be improved	Clarity
189.	*Therefore, we need an additional statistical test named Anderson-Darling test for the null hypothesis that the accuracies obtained for FedHealth and HashLearner are from normal distributions.*	Ungrammatical sentence	Correctness

190.	*Therefore, we need an additional statistical test named Anderson-Darling test for the null hypothesis that the accuracies obtained for FedHealth and HashLearner are from normal distributions.*	Paragraph can be improved	Clarity
191.	*After applying Anderson-Darling test, the null hypothesis did not reject with alpha level 5\%.*	Ungrammatical sentence	Correctness
192.	*After applying Anderson-Darling test, the null hypothesis did not reject with alpha level 5\%.*	Paragraph can be improved	Clarity
193.	~~applying~~ → using	Word choice	Engagement
194.	the paired	Determiner use (a/an/the/this, etc.)	Correctness
195.	*After applying this test, the null hypothesis is rejected with alpha level 5% and p-value 5.6e-6 means that the accuracy of our proposed HashLearner is statistically better than FedHealth.*	Ungrammatical sentence	Correctness
196.	*After applying this test, the null hypothesis is rejected with alpha level 5% and p-value 5.6e-6 means that the accuracy of our proposed HashLearner is statistically better than FedHealth.*	Paragraph can be improved	Clarity
197.	*The experiments in our study are based on a Google Colab virtual machine using a T4 GPU with 12 GB RAM.*	Paragraph can be improved	Clarity
198.	*At each round of training an approximate time of 3~4 seconds had been added as the election, gossiping and multi-step aggregation phase in a simulated local environment.*	Ungrammatical sentence	Correctness

199.	*At each round of training an approximate time of 3~4 seconds had been added as the election, gossiping and multi-step aggregation phase in a simulated local environment.*	Paragraph can be improved	Clarity
200.	, model	Incorrect punctuation	Correctness
201.	, such	Incorrect punctuation	Correctness
202.	~~2~~ → two	Improper formatting	Correctness
203.	~~submit~~ → submission	Incorrect phrasing	Correctness
204.	*These 2 approaches can lead to lower model transfer time over the network and a faster HashGraph model submit.*	Paragraph can be improved	Clarity
205.	the time	Determiner use (a/an/the/this, etc.)	Correctness
206.	*It is worth mentioning that time taken to finalize blocks in a HashGraph will increase logarithmically relative to the number of nodes participating in the learning process.*	Paragraph can be improved	Clarity
207.	*Using sharding will increase both the consensus speed and the aggregation speed as it decreases the number of nodes participating in the consensus and also the number of models being aggregated.*	Paragraph can be improved	Clarity
208.		Tone suggestions	Delivery
209.	~~decreasing~~ → reduce, reducing	Word choice	Engagement
210.	, which	Incorrect punctuation	Correctness
211.	~~decrease~~ → reduce	Word choice	Engagement

212.	*It is worth mentioning that decreasing the number of nodes in the consensus will do a trade-off between speed and security which means that it will decrease security to increase speed.*	Paragraph can be improved	Clarity
213.	*Larger shards are slower to finalize block and aggregate models but are more secure as more nodes participate in consensus and more validators can be chosen for aggregating the models in the shard.*	Ungrammatical sentence	Correctness
214.	*Larger shards are slower to finalize block and aggregate models but are more secure as more nodes participate in consensus and more validators can be chosen for aggregating the models in the shard.*	Paragraph can be improved	Clarity
215.	*Scalability and throughput: Due to Swirlds HashGraph paper proofs, the HashGraph consensus has higher throughput than other multi party computation or Blockchain alternatives.*	Ungrammatical sentence	Correctness
216.	*Scalability and throughput: Due to Swirlds HashGraph paper proofs, the HashGraph consensus has higher throughput than other multi party computation or Blockchain alternatives.*	Paragraph can be improved	Clarity
217.	*Also HashGraph makes peers able to submit blocks concurrently which increases scalability efficiently.*	Ungrammatical sentence	Correctness
218.	*Also HashGraph makes peers able to submit blocks concurrently which increases scalability efficiently.*	Paragraph can be improved	Clarity
219.	*Also the sharding algorithm proposed by our paper distributes load and computation bottlenecks and improve scalability compared to other papers mentioned in related work.*	Ungrammatical sentence	Correctness

220.	*Also the sharding algorithm proposed by our paper distributes load and computation bottlenecks and improve scalability compared to other papers mentioned in related work.*	Paragraph can be improved	Clarity
221.	*HashLearner has higher throughput than FeDis \cite{refre19} and DFL \cite{refre20} because instead of Blockchain, it relies on hashgraph with asynchronous gossiping and block finalization and uses sharding to distribute computation load across the network at the same time.*	Ungrammatical sentence	Correctness
222.	*HashLearner has higher throughput than FeDis \cite{refre19} and DFL \cite{refre20} because instead of Blockchain, it relies on hashgraph with asynchronous gossiping and block finalization and uses sharding to distribute computation load across the network at the same time.*	Paragraph can be improved	Clarity
223.	an ideal	Determiner use (a/an/the/this, etc.)	Correctness
224.	situation,	Incorrect punctuation	Correctness
225.	, and	Incorrect punctuation	Correctness
226.	~~real word~~ → a word	Incorrect phrasing	Correctness
227.	~~word~~ → real-world	Confused words	Correctness
228.	, due	Incorrect punctuation	Correctness
229.	, but	Incorrect punctuation	Correctness
230.	, including	Incorrect punctuation	Correctness
231.	, and	Incorrect punctuation	Correctness
232.	~~and there~~ → . There	Hard-to-read text	Clarity

233.	*Also HashLearner outperforms MPCFL \cite{refre21} in throughput and also provides higher security as it uses a distributed ledger and higher level of verification.*	Ungrammatical sentence	Correctness
234.	*Also HashLearner outperforms MPCFL \cite{refre21} in throughput and also provides higher security as it uses a distributed ledger and higher level of verification.*	Paragraph can be improved	Clarity
235.	~~method~~ → Method	Confused words	Correctness
236.	~~is designed to enable~~ → enables	Paragraph can be improved	Clarity
237.	earlier,	Incorrect punctuation	Correctness
238.	~~Uses~~ → uses	Confused words	Correctness
239.	*HashLearner Uses a decentralized consensus mechanism (HashGraph) to ensure secure and transparent model aggregation and reduces the risk of single-point failures and attacks targeting a central authority.*	Paragraph can be improved	Clarity
240.	the HashLearner	Determiner use (a/an/the/this, etc.)	Correctness
241.	paper,	Comma misuse within clauses	Correctness
242.	*Here, adversaries may observe update flows and corrupted nodes' contributions.*	Paragraph can be improved	Clarity
243.	Byzantine-resilient	Paragraph can be improved	Clarity
244.	*This*	Intricate text	Clarity

245.	~~and infer~~ → and infer	Improper formatting	Correctness
246.	*Confidentiality of updates and reconstruction attacks: Randomized shuffling ensures adversaries cannot link specific updates to honest nodes and infer private data except with negligible probability.*	Paragraph can be improved	Clarity
247.	*Poisoning resistance: Because aggregation requires consensus, adversarial influence is bounded, preventing domination of the global model. also it ensures that no single node can manipulate the aggregation result.*	Ungrammatical sentence	Correctness
248.	*Resource-efficiency balanced with security: Nodes with limited resources such as IoT nodes can easily participate without overburdening a central server, as the aggregation workload is securely shared.*	Ungrammatical sentence	Correctness
249.	*Peer-to-peer communication: Reduced latency and improved scalability compared to centralized systems and removed central server leading to elimination of central communication and data transfer proxy that could be insecure and vulnerable to data manipulation.*	Ungrammatical sentence	Correctness
250.	*Transparent and auditable: HashGraph's consensus mechanism ensures that all transactions such as model updates are recorded and verifiable, reducing the risk of malicious behavior of nodes and HashLearner can easily act as intrusion detection system IDS confronting malicious nodes in poisoning atta...*	Ungrammatical sentence	Correctness
251.	*Although complete privacy cannot be guaranteed in gradient-based learning, we analyze the reduction in leakage using empirical evaluation.*	Paragraph can be improved	Clarity

252.	*Baseline FL (FedAvg): it is vulnerable to membership inference attack and has the lowest privacy in this comparison.*	Ungrammatical sentence	Correctness
253.	*Baseline FL (FedAvg): it is vulnerable to membership inference attack and has the lowest privacy in this comparison.*	Paragraph can be improved	Clarity
254.	~~its~~ → Its	Improper formatting	Correctness
255.	~~its~~ → Its	Confused words	Correctness
256.	*Blockchain-based FL: The use of immutable ledgers enhances high level of privacy defense but because no native obfuscation is applied, model weights recorded on the ledger remain exposed to potential inference attacks.*	Ungrammatical sentence	Correctness
257.	*An adversary can analyze these publicly available gradient updates over time to perform attacks such as gradient inversion attacks or membership inference attacks, similar to those in traditional FedAvg.*	Paragraph can be improved	Clarity
258.	Also,	Comma misuse within clauses	Correctness
259.	*HashLearner: The privacy of shuffling system is close to random guessing, due to peer to peer shuffling of the model weights and consensus-driven aggregation.*	Ungrammatical sentence	Correctness
260.	~~final~~	Wordy sentences	Clarity
261.	*This*	Intricate text	Clarity
262.	*This is because only the peer layers (e.g., the third layer of one model with the third layer of another) are shuffled with each other.*	Paragraph can be improved	Clarity

263.	*It is worth mentioning that a higher number of nodes and a higher number of layers in a deep learning model increase shuffling security and make it harder for an adversary to identify the real models and infer data.*	Paragraph can be improved	Clarity
264.	*Privacy is inherently preserved by the decentralized nature of HashGraph and the multi-step aggregation process based on a probabilistic process, hence No reliance on a trusted third party is achieved.*	Ungrammatical sentence	Correctness
265.	*Privacy is inherently preserved by the decentralized nature of HashGraph and the multi-step aggregation process based on a probabilistic process, hence No reliance on a trusted third party is achieved.*	Paragraph can be improved	Clarity
266.	, and	Comma misuse within clauses	Correctness
267.	Also,	Comma misuse within clauses	Correctness
268.	~~it eliminates~~ → it eliminates	Improper formatting	Correctness
269.	*this points and properties makes HashLearner a highly secure and scalable solution for federated learning.*	Ungrammatical sentence	Correctness
270.	~~this points~~ → This points	Improper formatting	Correctness
271.	*These findings indicate that HashLearner provides a practical pathway toward trustworthy and scalable federated learning in adversarial and resource-constrained settings.*	Paragraph can be improved	Clarity