Decentralization of Artificial Intelligence: Analyzing Developments in Decentralized Learning and Distributed AI Networks

Ishan Gupta
Student, Seth Anandram Jaipuria School, Kanpur

***Abstract*— While Artificial Intelligence models are traditionally centralized at all stages (development, training, testing, deployment etc.) throughout their lifecycle, there are various disadvantages and problems, as discussed in this paper, associated with such a centralized model. Convergence of various new technologies are facilitating the development of unconventional decentralized or distributed frameworks in which AI models learn from decentralized data; the problem-solving process can also be decentralized by breaking down the problem into subproblems which are handled separately by different AI programs specializing in them. This decentralization has been catalyzed by developments in Blockchain technology and Cryptography. The growth of Decentralized Artificial Intelligence (DAI) has been fueled by various factors and has often been termed as the "democratization" of AI. Emerging decentralized Machine Learning (ML) frameworks, Federated Learning and Distributed AI marketplaces are indicating the growth of DAI. This paper closely analyzes the functioning and growth of DAI systems and recent developments in the field. It also discusses various challenges that arise in a decentralized model along with numerous potential solutions.**

***Keywords*— Decentralized Artificial Intelligence, Federated Learning, Data Privacy, Blockchain.**

I. Introduction

Distributed Artificial Intelligence or DAI can be defined as a subfield of Artificial Intelligence which, unlike a traditional AI model, follows a decentralized and collaborative approach towards solving problems by breaking a complex problem into smaller subproblems and handling each one individually.

The models that handle these "subproblems", in a DAI system, are called the "agents" or "nodes" which are generally autonomous or semi-autonomous and are located geographically far away from each other. Furthermore, these agents are able to communicate with each other; enabling this collaborative approach and communication requires a specific protocol which is developed by various Decentralized AI networks. The characteristic feature of a DAI agent is its autonomy and ability to communicate and function independently.

In a decentralized model, there are various data privacy concerns that must be resolved; this is where cryptography (homomorphic encryption, SMPC, etc.) plays an important role. Additionally, Blockchain and Smart Contracts are being used to reward the developers and data scientists and enable the monetization of individual AI solutions (without the need to trust a central authority) available on the DAI network.

Such a decentralized model has various features that solve the challenges posed by a traditional centralized AI system; these features and challenges have been closely analyzed in this paper.

The field of Decentralized AI has recently witnessed numerous developments and can be viewed as a rapidly evolving industry. There are various organizations, projects, developers and researchers that are contributing to this development; however, all of them do not follow a common approach towards Decentralized AI leading to inconsistency in definitions and ambiguity surrounding the implementation and structure of these systems.

The core principle remains the same- decentralizing a traditionally centralized method of AI development. However, in recent years, two different approaches towards decentralization have emerged with both focusing on decentralized two different stages in the lifecycle of an AI model.

1) Decentralized data acquisition, learning and training-
This approach focuses on decentralizing the process of Deep Learning and training the AI model through Federated or Distributed Learning. Unlike traditional AI, instead of a large dataset compiled in a single location, multiple and relatively smaller datasets are used to train the AI model.
2) Decentralized model development and solution-
This approach focuses on decentralizing the process of developing solutions and creating the AI model. In this approach, the problem-solving process is decentralized by collaboration of various independent AI solutions specializing in different tasks. It can be compared to a "divide-and-conquer" algorithm which breaks problems into subproblems and handles them individually. Individual results are combined to reach a final solution.

Although both of the aforementioned approaches use decentralization to solve different problems, the central idea of democratizing the Artificial Intelligence ecosystem (which presently seems to be moving towards monopolization) is common. [1]

II. Conventional Centralized AI Model and Life Cycle

Traditionally, the life cycle of an AI Model is mostly centralized at every stage from data acquisition and learning to deployment.

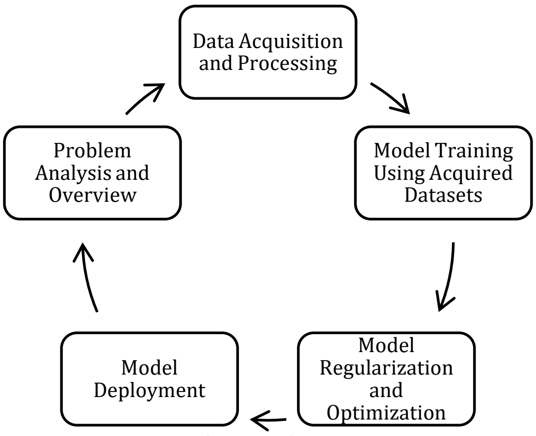

Fig. 1 Life cycle of an AI Model

Most Data Science projects developed in the last few decades follow this standard development pattern and life cycle. Multiple iterations of this end-to-end cycle might be required for creating an efficient model. In most processes involved in this lifecycle, some degree of decentralization is possible.

III. Disadvantages of Conventional Centralized AI

As demonstrated in Figure 1, the development of a decentralized AI model involves various stages. All these stages are traditionally centralized. This centralized approach of creating and deploying AI has resulted in numerous issues which have been discussed below.

A. Oligopoly in the AI ecosystem

The Artificial Intelligence industry has been witnessing monopolization in the past few years; a few “technology giants" (primarily Google, Apple, Facebook, Microsoft and Amazon) govern major portions of the industry and are largely responsible for spearheading the AI movement. Numerous upcoming ventures in the field of AI have also been acquired by these corporations. The main reason for this monopolization is the interdependence of various stages of AI development. [2]

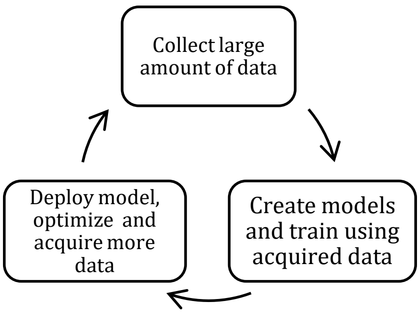

Fig. 2 Cyclic AI Development Process

As demonstrated in Fig. 2, access to huge datasets have enabled industry leaders to develop more accurate and successful AI models and further acquire data for training future models; this has often been termed as the "rich get richer" problem. The ownership of the acquired datasets is centralized; therefore, the large tech companies are the sole owners of these datasets. While these companies are able to successfully train and optimize AI models using these datasets, most new ventures lack the access to such datasets necessary for building a successful model. In a decentralized approach towards AI, multiple agents can contribute to the training of AI models through Federated Learning and Collaborative AI. [3]

It has been argued that this monopolization has been restricting the capabilities of AI in the area of social innovation. Large technology corporations have access to huge datasets, resource capabilities and exceptional professionals in data science and machine learning. Corporations are primarily developing AI for general commercial usage; a decentralized AI network can encourage innovation and efficient usage of AI in resolving major social issues and global challenges since it allows access to large decentrally-owned datasets and most importantly, a medium to collaborate for machine learning and model training.

B. Data privacy issue (centralized learning)

This is one of the most serious problems caused due to traditional centralized AI models and the undemocratic AI ecosystem. As discussed above, major tech corporations have access to large datasets acquired through various sources. Firstly, the ownership of data is centralized i.e. the companies are the owners of the data acquired from users. Secondly, the usage of data is generally not transparent i.e. a user is generally not informed about how and where their data is being used by the corporation. Today, data privacy is a major concern due to various cases of data leaks witnessed in the past few years.

Data privacy issues are particularly harmful when dealing with sensitive data. For example, in order to develop an AI model for hospitals which predicts diseases or deals with particular diagnostics, patient data and medical history would be required. However, medical history is very sensitive information and is vulnerable to misuse. Following a centralized AI approach, the organization creating the model would either fail to secure enough datasets to develop accurate models or would get access to sensitive medical records, both of which are undesirable situations. Increasing cases of data harvesting and data scandals are a matter of grave concern; centralized ownership of data (in traditional AI development) has often been termed as “unethical” and “harmful” as it puts users’ personal data at risk. [4]

Federated Learning or Collaborative Learning aims to tackle this problem through decentralized deep learning. The “incentive mechanism” and a few other components incorporated in Microsoft’s Decentralized and Collaborative

AI on Blockchain aims to prevent “bad data” in decentralized training.

C. Unnecessary efforts and reduced efficiency for developers

Due to the absence of frameworks for a decentralized collaborative AI development, traditional practices do not facilitate the sharing of AI models or tools. Due to this, unnecessary reduplication of AI models is required. Most AI tools that predict results and aim to optimize solutions require a collection of numerous models performing various specific functions. Again, large technology corporations have access to various such tools due to huge resources and competent teams of professionals.

Today, almost all industries have some requirement or application for AI services. From healthcare and finance to education and manufacturing, all industries can benefit from the use of AI in understanding consumer behaviours, personalizing features, predicting market responses etc. However, a significant percentage of businesses are unable to efficiently use AI to benefit their operations because of two reasons. Firstly, pre-existing models that can be purchased are usually not optimized enough for every business to produce desirable results. Secondly, most businesses, excluding large technology corporations, are unable to hire competent teams of AI professionals who are able to create customized AI models to suit their purpose. Through emerging decentralized AI networks, companies could actually combine various semi-autonomous AI services to create a custom model, without a lot of effort, which is able to fulfill the requirements.

If the AI ecosystem is democratized, such specific tools could be easily monetized and made available through different frameworks and networks. Furthermore, in a decentralized AI model, the different “agents” or, in this case, AI tools performing specific subtasks are autonomous or semi-autonomous. These agents can communicate with each other and get tasks performed as per the requirement. In such networks, the use of models to get results is monetized i.e. it requires certain “bounty” to be paid to the creator of the model.

This problem has been specifically targeted by decentralized AI marketplaces such as SingularityNET. In the past few years, various research teams and developers have been working on creating protocols, frameworks and networks for enabling the interoperability of various AI models to work together and for the developers of these models to be rewarded for the usage of their services.

IV. Machine Learning Using Decentralized Data

Traditionally, AI Models are trained using centralized data i.e. data stored in a central location. This, of course, raises various privacy concerns since entire datasets are accessible. Recently, various approaches to train AI Models without the need to compile all data in a central location have emerged.

A. Federated Learning

Federated Learning is one of the most promising techniques aimed at decentralizing the process of AI development. Collaborative or Federated Learning is a Machine Learning (ML) in which AI models are locally trained on various decentralized devices or clients holding their own local data samples. In this approach, data samples are not exchanged or compiled in a central server.

With the emergence of IoT (Internet of Things) and the growth of portable technology, there are billions of devices, which are connected to the internet, around the world. With Federated Learning, the huge data collected by these devices can help train very accurate and efficient AI models without the risk of data leaks and privacy concerns.

Each round of the process of Federated Learning can be summarized in the following five steps-

- The central server selects a statistical model to be trained and a subset of nodes or clients for the training process.
- The central server transmits the same model to all nodes selected (based on specific criterion) in the current round.
- The clients retrieve the model and locally train it using the local user data on the device.
- The nodes send the updated results back to the central server. Results from all selected clients are aggregated to get a final updated model.
- The updates are made to the original model and the process is repeated again with the updated model and a new set of clients. [5]

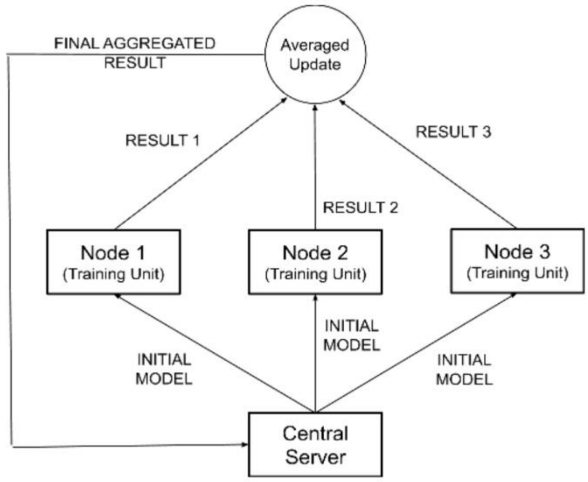

Fig. 3 Federated Learning Process

Notable points related to the process-

- Nodes are usually selected on the basis of numerous factors. For example, in the case of mobile devices-battery life, connectivity, network bandwidth, available computation power etc.
- As opposed to various ML training methods, in Federated Learning, datasets and mostly heterogeneous i.e. they differ in magnitude, distribution, etc.
- In a node, the resulting model might be iterated over the local data numerous times before averaging all nodes’ results in the central server. The number of times a client must train the model in each round before sending back

the updates depends upon various circumstantial factors. A February 2017 study related to Federated Learning analyzes two algorithmic approaches towards Federated Learning using decentralized data- FedSGD and FedAvg. SGD stands for Stochastic Gradient Descent, a common and effective approach often used in Machine Learning algorithms. FedSGD or FederatedSGD is a federated variant of the SGD algorithm with a few upgrades and additional parameters for batch size, clients etc. FedAvg stands for Federated Averaging algorithm; it works similar to the FedSGD but the nodes perform multiple updates using local data, with weights instead of gradients, before transmitting the results to the central server for weighted averaging. [6]

- Numerous iterations of the above process with different subsets of clients are usually performed in order to achieve desired accuracy. Generally, after a given number of rounds, the model stops showing improvement in accuracy.

In this approach of Machine Learning, a user's private data never actually leaves their personal device. The developers are able to train their models on the user's own device using their data. Since no transmission of data takes place, there is no possibility of data leaks. Additionally, the user data cannot be reconstructed from the results since it is encrypted. Therefore, Federated Learning tackles various problems such as data privacy, security, handling sensitive data etc. FL models might still have a risk of unintended memorization; this risk can be handled using differential privacy which might come at the cost of increased computations.

Federated Learning is also quite different from Distributed Learning. In Distributed Learning, it is a common assumption that the datasets are homogenous or evenly distributed; however, this is not true for Federated Learning. Since datasets are stored locally and the magnitude of data is dependent upon various unique factors related to the usage, datasets are mostly heterogeneous.

It must be noted that Federated Learning decentralizes three of the steps show in Figure 1-

- Data acquisition and processing
- Model training using acquired datasets
- Model regularization and optimization

However, Federated Learning also has a few limitations as listed below.

- Datasets are heterogeneous and non-uniformly distributed. Datasets vary a lot in magnitude and range and might have unintended effects on the model.
- Since datasets are not accessible, it is difficult to recognize parameters decreasing the accuracy or "bad data".
- Models trained through Federated Learning are mostly static i.e. changes to the model can only be provided with regular updates.
- Specific communication mechanisms are required to facilitate the training of models.
- Limited time is available for the models to be trained without affecting the node device's performance.
- Node devices must have required computational power and specifications to execute the training process locally.

Libraries and Applications:

1) TensorFlow Federated (TFF)-
TensorFlow is a renowned open source platform for developing Machine Learning applications. It provides various libraries, tools and resources for developers to develop ML-powered applications.
TensorFlow Federated (TFF) is an open-source framework developed by TensorFlow for machine learning on decentralized data. Besides Machine Learning, it also allows users to perform other decentralized computations such as aggregated analytics. The TFF platform has two layers- Federated Learning (FL) and Federated Core (FC).
2) Google's Gboard-
Gboard is Google's virtual keyboard for mobile devices. Next-word prediction is a crucial feature of the Gboard and a February 2019 study shows the results of training Gboard to give more accurate next-word predictions using Federated Learning. The study compares results of server-based training using SGD (Stochastic Gradient Descent) and training using the FederatedAveraging (FedAvg) algorithm. This is one of the first few real-world applications of Federated Learning. [7]
CIFG (Coupled Input-Forget Gates) is a variant of RNN (Recurrent Neural Network), a class of Artificial Neural Networks. CIFG is specifically optimized for mobile devices since it significantly reduces the number of computations and parameter set size without affecting the models performance.
The aforementioned study finally compares the accuracy and results from Federated CIFG and Server-based CIFG. The metric used to compare the two is "Recall", the ratio of the number of correct predictions to the total number of tokens. Furthermore, results from top-1 recall (one word suggestion) and top-3 recall (three word suggestions) were both considered separately. For both, the Federated CIFG model performed better than the server-based CIFG model. Most importantly, federated learning models protect the user's privacy.

B. *Microsoft's Decentralized and Collaborative AI on Blockchain*

In a July 2019 study, researchers at Microsoft proposed a framework called "Decentralized & Collaborative AI on Blockchain" for decentrally training and improving AI models. Presently, this framework is operational and publicly available as it has been open-sourced on GitHub. Using this framework, developers can host their model using smart contracts on a Blockchain. [8]

This framework consists of three primary components that facilitate its functioning-

- Incentive Mechanism-
  It coordinates smart contract rewards and validates the transactions made (Ethereum Blockchain-based).
- Data Handler-
  Stores data and meta-data on Blockchain and ensures it is publicly available.
- Machine Learning Model-
  It is concerned with predictions and training of models uploaded by developers.

Unlike the Federated Learning approach, in this framework, the process of training AI models and contributing datasets has been incentivized for the users who wish to share data and train models. Contributors can easily contribute to improve the model; in order to encourage contributions, several incentives (financial and non-financial) have been provided for the user within the framework. One of the unique features of this framework is the incentive mechanism to promote submission of "good data" which works simultaneously with a penalty mechanism to discourage submission of "bad data" by the contributors.

Initially, this framework has been implemented using Ethereum, an open source, Blockchain-based distributed computing platform offering smart contract functionality, as it provides the most optimum solution. However, it is suggested that the framework can also use another Blockchain if a better suited alternative is available.

The following 4 steps summarize the basic process of model training through this framework-

1. Developers upload a lightly pretrained (very limited accuracy) AI model on the Blockchain through a smart contract.
2. Contributors test the model by getting predictions for input data without any additional costs.
3. Contributors "stake" a required deposit in order to submit a data contribution for training purposes.
4. One of the following two scenarios takes place-
   4.1. If the data submitted is "good", the deposit is refunded with some additional incentive.
   4.2. If the data submitted is "bad", the deposit is lost as penalization.

Using a Blockchain instead of traditional source code hosting practices provides various advantages to this framework. Firstly, Blockchain is a Distributed Ledger Technology (DLT) and most importantly, it provides a "trustless system" i.e. there is no need to trust individual entities or a third party for any transactions, modifications, additions, etc. Blockchain technology has been discussed in a more detailed manner later in this paper. In the context of this framework, Blockchain provides a transparent and reliable method of sharing models. Customers and contributors can therefore view the model's smart contract due to its public availability. Since the usefulness or quality of a data provided might be a subjective metric in various cases, such transparency and fairness provided by smart contracts assures the data contributors that they would be fairly compensated for their contributions.

The aforementioned compensation is provided through any of the three components of the incentive mechanism-

- Gamification-
  In this mechanism, a non-monetary compensation is provided to the data contributor. It can be compared to the badges or points awarded on various platforms.
- Reward Mechanism-
  The contributor is provided some financial rewards based on the improvement in the accuracy of the model due to the data contributed. These financial rewards are supplied from a pool of reward funds provided by the development company or any entity that wishes to encourage the training of the model.
- Deposit, Refund, Penalty-
  Since a smart contract cannot legally compel a user to pay a penalty, the deposit submitted by the contributors before making the contribution is actually "staked" by them. In case the data provided is good, the deposit is returned and financial incentive is provided.

Microsoft claims that the vision of this project is to democratize the AI field.

Although this framework is quite progressive and it will certainly encourage developments in the field of Decentralized Artificial Intelligence, there are still quite a few limitations related to this mechanism-

- It cannot be used with unsupervised models; it is only available for supervised models i.e. models trained on labeled datasets containing both input and output parameters.
- It cannot be used with complicated models. It is presently viable for only simpler models with simple input formats.
- The incentive mechanisms do not offer a proper source of compensation for contributors who might not be encouraged to participate in the training process.

V. Interoperability Between AI Models in a Decentralized Network

In Decentralized AI, the handling of problems is different from traditional centralized AI systems because of the absence of a central server which coordinates all tasks. In recent years, various decentralized AI networks and frameworks have emerged. However, in such networks, there are various problems that arise.

The following sections discuss the architecture and problems associated with decentralized systems.

A. *Structure and Functioning*

Fig. 4 below demonstrates the functioning of a centralized multi-AI model.

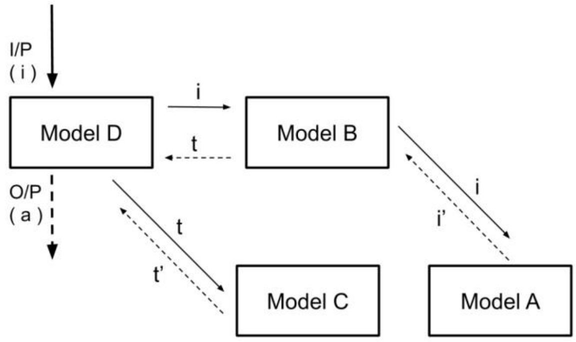

Fig. 4 Centralized Multi-AI Model

In such a model, the working of all AI services is coordinated by a central authority "X". For example, let us assume that Model A is a tool used for clarifying images containing handwritten text to increase legibility; Model B is an OCR (Optical Character Recognition) tool which reads handwritten text and changes it to machine-encoded text; Model C translates machine-encoded texts in French to English and vice-versa; Model D is a text-to-speech tool which processes machine-encoded text in English and reads it aloud.

To better understand how a centralized model would try and solve a multi-level problem, we can refer to Fig. 4. As demonstrated in Fig 4, if an unclear image "i" of a handwritten text in French would be fed as input to the central point X and the desired output is an audio file "a" containing the input text read aloud in English, then X would sequentially process the image through nodes A, B, C, and D to generate the desired output.

Fig. 5 below demonstrates the functioning of a decentralized multi-AI model.

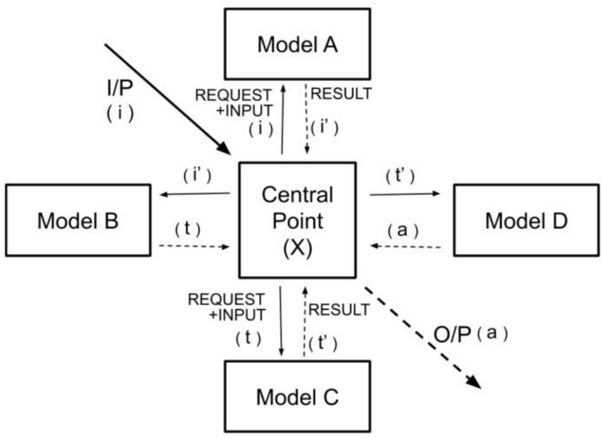

Fig. 5 Decentralized Multi-AI Model

In such a model, all nodes can independently interact with each other under given protocols or communication network guidelines. If the aforementioned text-to-speech problem is processed in a decentralized architecture, a different approach is observed. The input (an unclear image of a handwritten text in French (i)) is fed to Model D. Since the input is an image which D cannot process, it requests B to process the image and send back a machine-encoded text. B finds that the image is not clear and the text is not adequately legible thus B sends the image i to A; A sends back a clarified image i'. B processes the clarified image i' and extracts a machine-encoded text t from it and sends this text to D. D finds that the text t is in French; therefore, D sends this text t to C and C sends back the translated text t' to D. D now receives a machine-encoded text t' in English; it generates the desired output i.e. the audio file "a" containing the text read aloud in English.

Architecture of decentralized AI systems resembles Multi-agent systems and Decentralized Autonomous Organizations (DAOs). The nodes or agents (Model A, Model B, etc. in Fig. 5) in a decentralized AI system generally have the following properties-

- The nodes are independent i.e. all nodes are distinct entities.
- The nodes are autonomous or semi-autonomous i.e. capable of independent decisioning.
- The nodes are loosely coupled.
- The nodes might be located geographically far away from each other.

B. Challenges in decentralization

With such a system, there is a problem that arises- finding a suitable mechanism to incentivize developers for the use of their models. Since models operate with each other decentrally and problems are solved collaboratively, all models in a system are not owned by a central entity. [9]

Therefore, in a decentralized network, there is a requirement for the following two things-

- A fair way to compensate for the usage of models, without the need to trust a third party to manage or oversee transactions, through a democratic process.
- A method to make executions and transactions related to third-party AI models transparently available without having the need for a centralized authority.

The most optimum solution for both these problems is Distributed Ledger Technology (DLT) or specifically, Blockchain and smart contracts. The specifications and details of Blockchain technology is beyond the scope of this paper; however, in simple terms, a Blockchain is an immutable ledger which stores transactions in a verifiable way using decentralization and cryptographic hashing. One of the Blockchain-based platforms, most commonly used for facilitating transactions in decentralized AI networks, is Ethereum. Ethereum has its own crypto currency (Ether) and tokens for smart contracts.

Smart contracts are self-executing contracts or computer programs that facilitate or initiate a pre decided transaction when given conditions are met. These smart contracts are stored on the Blockchain for permanency and transparency.

Generally, Blockchain and smart contracts are used to host AI models in a decentralized network; for using a model, a user must pay a given amount or "bounty", usually in the form of crypto currency or tokens, to the creator of the model. Since it is based on a Blockchain, all transactions are immutably stored and it provides a reliable, verifiable, decentralized and efficient way of incentivizing AI models in a decentralized network.

Smart contracts are also used in decentralized learning frameworks in order to provide incentives to users who choose to train models using their own data. This method has been implemented in projects such as OpenMined.

However, in such a case, the model must be encrypted. Since the model is sent to the user's personal device (in Federated Learning), there is a risk of the model being stolen. To prevent this, AI models can be homomorphically encrypted before sending it to the users for training. Homomorphic encryption (HE) allows operations on cipher text without decryption; therefore, the model is completely protected during the local training. Although homomorphic encryption might seem ideal, it is still mostly theoretical, less efficient and not optimized for application. Alternatively, GAN cryptography or Adversarial Neural Cryptography, introduced in 2016, and SMP (Secure Multi-Party) computations provide more efficient approaches. SMPC is much more practical and efficient to work with Blockchain; in 2014, a study demonstrated implementation of SMPC on Bitcoin. [10] [11]

C. Frameworks and Organizations

In the past few years, various organizations have been venturing into the field of Decentralized Artificial Intelligence and Machine Learning. Most initiatives follow a common theme or a similar vision; however, they greatly differ in approach and features.

1) SingularityNET-
SingularityNET is a decentralized open market and network for AIs; it is a leading initiative in the field of Decentralized AI. It serves as both- a "commercial launchpad" for developers and a mechanism for "AIs to interoperate". Transactions in SingularityNET are based on the AGI token. Users can use this token to purchase AI services from the marketplace. According to SingularityNET's Whitepaper 2.0, in the future, AGI tokens might also be used as a basis to provide access to voting rights for the network's democratic governance. [12]
SingularityNET presently uses Ethereum as the underlying Blockchain. It provides the SingularityNET SDK for clients who wish to use the network's services in their applications. SingularityNET is certainly one of the most ambitious and progressive projects which aim to democratize the AI ecosystem.

2) OpenMined-
OpenMined is an open-source community which focuses on privacy preserving and accessible AI and provides a decentralized AI platform. OpenMined leverages cryptography techniques such as SMPC and HE and Blockchain technology to decentralize machine learning. Similar to Federated Learning, OpenMined's decentralized ML focuses on methods to train on sensitive and private user data without raising privacy concerns; additionally, like Microsoft's Decentralized and Collaborative AI, it aims to incentivize users to encourage them to train AI models uploaded by developers. OpenMined contains two primary components- PySyft library and PyGrid platform.

3) Decentralized Artificial Intelligence Alliance (DAIA)-
DAIA, the Decentralized AI Alliance is an alliance of various organizations that work with AI and Blockchain to develop decentralized applications, networks, protocols, etc. This alliance aims to formalize the association between the members who share a common vision for the future of AI. It aims to encourage collaborations amongst members, encourage new ventures in the field of DAI and accelerate the decentralization of AI.

VI. Conclusion

Decentralized AI is presently an active area of research and application; however, almost all projects in this field are still in their initial stage. The recent emergence and growth of DApps (Decentralized Applications) and Distributed Ledger Technologies (DLT) demonstrate the progress towards distributed computing.

In the current AI paradigm, the growth of AI is largely propelled by huge technology corporations; in view of frequent data scandals, data leaks and data security issues, Decentralized AI seems to provide a reliable solution to prevent such privacy issues without hindering the progress of AI. Contrary to common belief, besides new ventures and emerging startups, even large technology corporations such as Google and Microsoft are actively researching potential methods to decentralize processes related to AI.

This paper has vividly discussed the working and specifications of various decentralized AI systems and has closely analyzed numerous approaches, technological enablers, applications, frameworks, and practices related to Decentralized AI. Relatively modern Cryptography techniques and developments in Blockchain-based platforms have certainly catalyzed the growth of Decentralized AI. With so many initiatives, projects and organizations aiming to decentralize AI and making it more accessible, Decentralized AI seems to be a promising area of research and development with huge potential for growth in the coming years.

References

[1] Schneegans, Susan. (2018). Towards a monopolization of research in artificial intelligence?.

[2] Khan, A. (2018, September 21). The Tech Oligopoly — Part 1. Retrieved April 8, 2020, from https://blog.singularitynet.io/the-tech-oligopoly-part-1-5d76df9d09aa

[3] Rodriguez, J. (2018, April 5). Why Decentralized AI Matters Part I: Economics and Enablers. Retrieved April 8, 2020, from https://medium.com/datadriveninvestor/why-decentralized-ai-matters-part-i-economics-and-enablers-5576aeeb43d1

[4] Timan, T. &Z. Á. Mann (eds) (2019) "Data protection in the era of artificial intelligence. Trends, existing solutions and recommendations for privacy-preserving technologies", October 2019. BDVA.

[5] McMahan, B., & Ramage, D. (2017, April 6). Federated Learning: Collaborative Machine Learning without Centralized Training Data. Retrieved April 8, 2020, from https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[6] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Arcas Blaise Aguera y. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv:1602.05629v3 [cs.LG].

[7] Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., Ramage, D. (2019). Federated Learning for Mobile Keyboard Prediction. arXiv:1811.03604v2 [cs.CL].

[8] Harris, J. D., & Waggoner, B. (2019). Decentralized and Collaborative AI on Blockchain. 2019 IEEE International Conference on Blockchain (Blockchain). doi: 10.1109/blockchain.2019.00057

[9] Rodriguez, J. (2018, April 6). Why Decentralized AI Matters Part II: Technological Enablers. Retrieved April 8, 2020, from https://medium.com/datadriveninvestor/why-decentralized-ai-matters-part-ii-technological-enablers-a67e3115312e

[10] Abadi, M., & Andersen, D. G. (2016). Learning to Protect Communications with Adversarial Neural Cryptography. arXiv:1610.06918v1 [cs.CR].

[11] Andrychowicz, Marcin & Dziembowski, Stefan & Malinowski, Daniel & Mazurek, Lukasz. (2014). Secure Multiparty Computations on Bitcoin. Communications of the ACM. 59. 443-458. 10.1109/SP.2014.35.

[12] SingularityNET (2019, February). Whitepaper 2.0. Retrieved April 8, 2020, from https://public.singularitynet.io/whitepaper.pdf