Machine Learning for All: A More Robust Federated Learning Framework

Chamatidis Ilias[1] and Spathoulas Georgios[1,2]

[1]*Department of Computer Science and Biomedical Informatics, University of Thessaly, Greece*

[2]*Center for Cyber and Information Security, Norwegian University of Science and Technology, Gjovik, Norway*

Keywords: Deep Learning, Federated Learning, Blockchain, Security, Privacy, Integrity, Incentives.

Abstract: Machine learning and especially deep learning are appropriate for solving multiple problems in various domains. Training such models though, demands significant processing power and requires large data-sets. Federated learning is an approach that merely solves these problems, as multiple users constitute a distributed network and each one of them trains a model locally with his data. This network can cumulatively sum up significant processing power to conduct training efficiently, while it is easier to preserve privacy, as data does not leave its owner. Nevertheless, it has been proven that federated learning also faces privacy and integrity issues. In this paper a general enhanced federated learning framework is presented. Users may provide data or the required processing power or participate just in order to train their models. Homomorphic encryption algorithms are employed to enable model training on encrypted data. Blockchain technology is used as smart contracts coordinate the work-flow and the commitments made between all participating nodes, while at the same time, tokens exchanges between nodes provide the required incentives for users to participate in the scheme and to act legitimately.

1 INTRODUCTION

Machine learning field has recently attracted a lot of interest. Advancements in hardware and algorithmic breakthroughs have made it easier and faster to process large volumes of data. In particular deep learning scheme, trains neural networks with a large number of nodes and multiple hidden layers. Taking advantage of the parallel processing capabilities of modern graphic cards, deep learning became quickly the main option for training large machine learning models upon big data data-sets.

Another relevant advancement, federated learning refers to multiple nodes which train models locally and then fuse these partial models into a single one. The resulting distributed network has a lot more processing power than a single machine, so it can perform faster and cope with larger volumes of data. Another critical issue is the collection of data to train the model. Traditionally data is gathered at a single host and training is carried out locally, but in federated learning the training happens at users' devices, so data does not need to be sent to a central server and thus privacy of the data holder is preserved. Although federated learning seems very interesting, it still has problems such as coordination of the whole process, privacy of the users data and performance issues.

Personal data are being gathered and used for training machine learning models. This happens with or without users' consent and usually gives them no control over the resulting models. For example, data such as biometrics, text input and location coordinates are private personal data, but are required in order to train models for biometric authentication, text prediction or navigation services respectively. Federated learning offers a solution to the problem mentioned above, because no central server gathers the users' data. In this scheme, models are trained locally at the users devices, without any third parties accessing their data and users only share the resulting trained models.

In this paper we present an approach for enhancing federated learning model in terms of privacy, management and integrity. Specifically we discuss the use of homomorphic encryption, blockchain technology and integrity mechanisms, in order to construct a more robust scheme for federated learning. Specifically Section 2 discusses deep learning and federated learning in more detail, Section 3 presents some of the most serious threats for the current model of fed-

Ilias, C. and Georgios, S.
Machine Learning for All: A More Robust Federated Learning Framework.
DOI: 10.5220/0007571705440551
In *Proceedings of the 5th International Conference on Information Systems Security and Privacy (ICISSP 2019)*, pages 544-551
ISBN: 978-989-758-359-9

erated learning, Section 4 discusses the main points of the proposed methodology and Section 5 discusses related research efforts through recent years. In Section 6 our future plans are presented and Section 7 discusses our conclusions.

2 MACHINE LEARNING

In this Section both deep learning and federated learning paradigms are presented.

2.1 Deep Learning

Deep learning is a rapidly growing field of machine learning. Because of the breakthrough in algorithms and also of the fact that hardware has recently become more efficient and less expensive to build, deep learning models have recently been massively employed in various applications. Deep learning is essentially the paradigm of training very large neural networks, with multiple hidden layers consisting of numerous nodes, by the use of very large data-sets (Deng et al., 2014).

The main advantage of deep neural networks is the large number of neurons allowing them to learn a great depth of detail of the data, used as input. Because of their ability to efficiently adapt to any dataset, deep learning networks are called universal approximators, for their ability to efficiently solve diverse problems. Thus deep learning is used in different domains, such as computer vision (Zeiler, 2013), speech recognition (Deng et al., 2013), natural language processing (Collobert and Weston, 2008), audio recognition (Noda et al., 2015), social network filtering (Liu et al., 2013), machine translation (Na, 2015), bioinformatics (Min et al., 2017), drug design (Jing et al., 2018) and biometric authentication (Chamatidis et al., 2017). Learning in deep networks can be supervised or unsupervised. Also there is an hierarchical relation between the nodes of the hidden layers, so each layer learns to a different abstraction of the data outputted by the previous layer, thus resulting into higher accuracy.

2.2 Federated Learning

As deep learning became more popular, researchers started experimenting on different problems, an important conclusion was quickly evident; the accuracy of the results of deep learning networks is highly coupled with the size of the training data-set. Larger networks are characterized by higher degree of freedom and thus more data is required for their training. The need for larger data-sets and consequently more computing power is an important issue. Also the form of the computing power required (GPU clusters) made it harder for individual users, to utilize proposed algorithms. The previous factors practically made deep learning available only to large corporations and organizations with the required resources for researching and developing systems under the aforementioned restrictions.

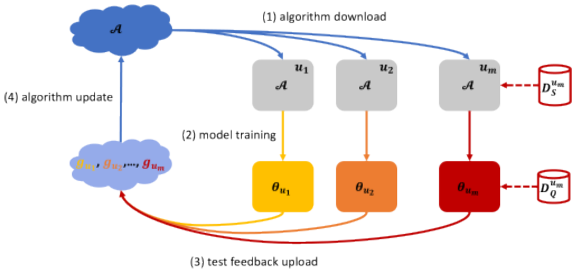

Figure 1: Architecture of a federated deep learning network.

Additionally several issues have recently emerged regarding the collection of data required for the training of deep networks. Training of such networks requires real-world data, which in most of the cases is personal data. Such data are usually captured by services that interact with multiple users, with or without the consent of the latter. Most data that is created by people is considered as personal information and it is protected by law, so a difficult legislative process is needed to collect and use this data.

Even if the data used is anonymous, it has been proved that significant privacy leaks may occur through various linking attacks (Backes et al., 2016). Recently Netfix released a data-set (Netflix prize (Bennett et al., 2007; Zhou et al., 2008)) consisting of 500,000 anonymous movie ratings, in an attempt to optimize their predictions algorithms. Authors in (Narayanan and Shmatikov, 2008) successfully demonstrated that it is possible to find a Netflix subscribers records in this data-set and furthermore uncover their political preferences along with other sensitive information about them.

As private information leaks become common, users become more concerned about the personal data they share. Also these collected data-sets may be too large to be processed or saved locally, while the training of the network is a resources demanding task, and requires large training times even for simple tasks. A proposed architecture that aims to solve the problem of data collection and concurrently make required computing resources available to more users is federated learning, also referred to as collaborative learning or distributed learning.

Federated learning architecture is usually based on the assumption that a node needs to train a machine learning model and that this training is partially committed in multiple other nodes. The main node, also identified as coordinator collects the trained models and combines those into a single model. The rest of the nodes train the partial models upon data that reside locally and send the trained models to the coordinator to be fused into a single final model.

Federated learning architecture is characterized by two significant changes with respect to the traditional model, where a single node has to collect all data and then conduct the required training of the model. Firstly there is no need for a single node to collect and store all the data locally. Secondly the processing power required is split among different parties, so it is easier to concentrate the required resources. This characteristic enables any node with no significant computing power available to participate into the scheme. In an example Google applied federated learning, using a mobile light version of Tensorflow (Abadi et al., 2015) and conducted text prediction tasks across its mobile platforms. Deep learning models were trained locally in mobile devices of users and then the results were fused in the Google cloud AI.

During training of the model first the coordinator node notifies the other nodes of a specific initial model to be used. Then the all other nodes train the same initial mode locally with data-sets they own. Finally, when they finish training, only the trained model is shared to the coordinator so that the final model can be calculated by aggregating all partial models.

3 SECURITY AND PRIVACY THREATS

Personal users' data are of great interest to companies, as they can use those to train models for targeted advertisement or product suggestion. At the same time, users tend to share personal information online in increasing rates with out thinking that their data might be used by someone else with out their consent. As technology and the internet progressed personal users' data have become a commodity for companies that is sold between them. Federated learning seems to be a reasonable solution to that issue. To begin with, there is no central entity that collects all the data. Instead, data is processed locally from its original owner and it is not required to be handled by any third party.

Federated learning seems to be the next big advancement in the deep learning research field, although there are many issues that need to be addressed, in order to be used in a real world use cases. Privacy and security problems still exist, despite the fact that the users share the minimum required data for the training process. In the following sections, these threats are analyzed.

3.1 Data Leak with Adversarial Attack

In federated learning environment, users train their own deep learning models locally. After training their model they sent the local gradients to the coordinator, in order to fuse those into a global model. As it is demonstrated in (Melis et al., 2018), through an adversarial attack it is possible to extract information about users' data set just by having access to the local gradients they sent to the coordinator.

3.2 General Adversarial Network Attack

This kind of attack is also executed by a malicious adversary node that pretends to be honest. This node tries to extract information about data classes he does not own by influencing the learning process to deceive the victim into releasing further information about the targeted data.

The adversary node has a General Adversarial Network(GAN) (Hitaj et al., 2017) generating data samples that look similar to the class of the victim. Then these fake classes are injected into the federated learning network, which then needs to work harder to distinguish between the real and the fake classes and thus reveals more information about the real data set than initially designed. The adversary node by forcing the victim to train between the real data and the generated fake data, can learn the distribution of the data, without accessing to the actual data points.

3.3 User Linkability Attacks

Another type of attack, as demonstrated in (Orekondy et al., 2018), is user linkability attack. In a federated learning environment, users share only the gradient to the central node, but it is possible to link these parameters to the users they belong to and extract information about them.

In this attack the malicious adversary node knows a set of gradient updates and the corresponding training data of the user. The adversary can learn to identify the user based on the produced gradient updates. In this kind of attack even adding random noise to the data doesn't seem to mitigate it. This type of attack proves that every communication during the training process has to be protected from privacy related attacks.

3.4 Data Poisoning

Another technique, as shown in (Bagdasaryan et al., 2018), aims to make a federated learning system composed of Convolutional Neural Networks(CNN) classifying images, to under-achieve in terms of classification accuracy. The attacker forces the CNN to miss-classify certain car images by feeding ”poisoned” images to the CNN.

Another experiment is also described, in a federated learning word predictor, where multiple cell-phones share text predictions that were trained locally in their devices, to update the text predictor of the service in a server. Authors showed that the attacker can force the text predictor to show a desired word as it's prediction. In fact with less than 1% of the total nodes being malicious the attacker can achieve 100% accuracy in the task. This attack can be mitigated by applying differential privacy to the individual nodes.

4 PROPOSED METHODOLOGY

We propose a general framework that will enable users to construct machine learning work-flows by sharing data, processing power and problems' definitions.

Each user of the system may have :

- A problem to solve by using machine learning methodology
- Data that can be used to solve such problems
- Processing power to conduct the required training

In practice each one of the users may posses one of the aforementioned assets/issues or even two of them. If a user has a problem, data to solve it and the required processing power then he can produce the solution on his own. In every other case he may take advantage of the proposed system. There are mainly three different scenarios to take into account :

- A user has a problem to solve, without data or required processing power
- A user has a problem to solve, has the required data but lacks processing power
- A user has a problem to solve has the required processing power but lacks required data

To tackle the problems of federated learning, we propose a general framework that will protect privacy of personal data, orchestrate the procedure and offer incentives for users to participate and also provide integrity and quality checking mechanisms for the resulting trained model. Data exchanged between nodes have to be encrypted, nodes that participate by providing data or processing power have to be rewarded, while in order for the produced models to be useful the integrity of the procedure has to be crosschecked.

The design of the system is going to be based on three main pillars, which are homomorphic encryption, blockchain technology and integrity checking mechanisms.

4.1 Privacy

4.1.1 Homomorphic Encryption

Homomorphic encryption algorithms allow for operations on the ciphertext, that correspond to actual operations on the original plain text. There are multiple algorithms that differ with regards to the level of processing allowed and to their efficiency. In general, data is decrypted with the private key and it is possible for a node that does not posses this private key to conduct specific processing on encrypted data and produce an encrypted result. This result is then returned to the initial data owner and he can decrypt that to get the result of the processing conducted in plain text form.

Homomorphic encryption has been studied as a mechanism for achieving privacy preserving machine learning. The model trainer can use sensitive private data in encrypted form, to train an encrypted machine learning model. Then he can return the trained model to the data owner to decrypt that. There is a lot of ongoing research in this field, as it could eventually enable outsourcing the training procedure of machine learning models to cloud services. Recently homomorphic encryption has been successfully used in deep learning and machine learning tasks (Takabi et al., 2016; Li et al., 2017; Rouhani et al., 2018; Aono et al., 2017).

4.1.2 Proposed Encryption Scheme

A proposed encryption scheme is depicted in Figure 2. In this example there are five different nodes **A,B,C,D,E**.

- Node **A** has a problem to solve
- Nodes **B,C** have the appropriate data
- Nodes **D,E** have processing power

Hypothesizing that node **A** needs to train a machine learning model, to solve his problem, he will utilize the resources offered by other nodes to do so.

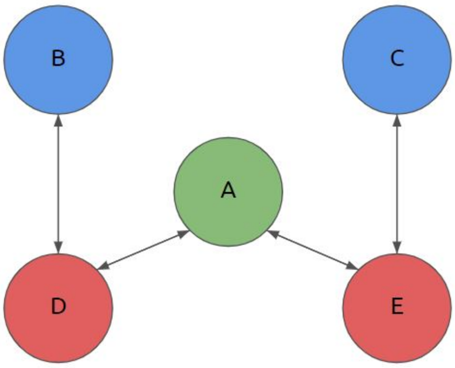

Figure 2: The system architecture with 3 kinds of nodes.

The procedure that the nodes need to go through is as follows:

1. Node A creates a pair of public and private keys and distributes the public one to other nodes
2. Nodes **B** and **C** encrypt their data with the common public encryption key
3. Nodes **B** and **C** distribute their encrypted data to nodes **D** and **E**
4. Node **A** defines a training model and encrypts its parameters
5. Node **A** sends the model to nodes **D** and **E**
6. Nodes **D** and **E** train the model with the data they hold and then return the encrypted model parameters to node **A**
7. Node **A** checks if the model has been successfully trained
8. If not, node **A** updates model parameters and steps 5 to 7 are repeated
9. If yes, the model is decrypted and used by node **A**.

Through the aforementioned steps and given the fact that no nodes are maliciously colluding, no node gets to know the personal data of nodes **B** and **C**. At the end of the procedure, node **A** has received the required trained machine learning model, with conducting minimal processing on its side.

Adding an encryption layer to federated learning approach in order to ensure privacy of participating nodes will of course create a processing overhead. On the other the proposed scheme will enable the participation of numerous additional, privacy conscious, processing nodes that would not take part in a simple federated learning collaboration. The processing overhead will be shared among multiple nodes and thus it will be more feasible to conduct large scale machine learning projects. Homomorphic encryption will have a minimal effect on the classification ratio of the produced models. The encrypted models produced by homomorphic approaches are characterized by a bearable reduction of classification ratio with respect to that of normally trained models.

4.2 Management

4.2.1 Blockchain Technology

Blockchain is an emerging technology that was made widely known to the research community through Bitcoin in 2008 (Nakamoto, 2008). Blockchain is a immutable, distributed and transparent shared ledger. This ledger is stored in every node of a peer to peer network of similar nodes. The action to append information to the ledger is called a transaction, and such transactions are stored into structures called blocks. The latter contain the transactions and a reference to the previous block (i.e, the hash of the previous block), thus forming a chain of blocks that cannot be altered under normal circumstances. For a transaction to be stored in the blockchain, it needs to be first validated by miners who compute a complex math function which is called proof-of-work and then add the new block to the chain. The miners gain a reward when they calculate the block and respectively the users who commit transactions pay a small transaction fee.

Blockchain technology has been used in many industrial and research applications, mainly through the use of smart contracts. Smart contracts are computer programs that can be executed on a virtual machine based on blockchain. The integrity of both the state and the execution of such programs is guaranteed by blockchain technology. In this sense, smart contracts are ideal for orchestrating the collaboration between parties that do not trust each other.

4.2.2 Procedure Orchestration

A user announces that he needs to solve a specific problem by training a model. He has to collaborate with a number of data owners and data processors to do so.

He instantiates a smart contract, through which he sets all the required details. He defines the network initial parameters and the data structure for each data point. He also sets a specific number of data points to be used as input and also the maximum time that each training epoch shall last and the maximum number of epochs allowed. Additionally he sets a maximum budget (in terms of tokens) he is willing to spent on both data owners and data processors. During the creation of the contract, he is obliged to send to the contract the sum of the two maximum budget amounts of tokens he has set. Finally he sets the expiration time for the bidding procedure.

Nodes that hold compatible data points may bid with the number of data points they can provide. They also set the number of tokens per data point they require for providing their data. Nodes that are able to provide the required processing power bid with the number of data points they are able to process in the specified epoch maximum time duration. Additionally they set the number of tokens per data point processing they require.

When the biding ends the contract automatically checks all submitted offers and resolves the bidding. In practice it selects the most profitable set of data owners and data processors for the user and publishes the results.

After the successful end of the procedure the contract pays the proper amount of tokens to the data owners and data processors and returns the rest of the tokens to the user. If a data processor fails to submit training results in the pre-set maximum time, then a penalty is activated and the final amount of tokens that is finally sent to that node gets reduced.

User is given the opportunity to back off from the procedure at each training epoch. Specifically at the end of each epoch the user decides to continue or not with the training procedure. This is done through the smart contract and in practice continuing to the next iteration means that every party agrees that the procedure is acceptable as legitimate by this point. If the user decides to back off before the procedure ends or because one of the other nodes seems to misbehave, then the payments go through according to the procedure up until this specific iteration and the user receives the partially trained model.

4.3 Integrity

An important factor for the successful training of the model is data integrity. Any change in data transferred between data owners and processing nodes and model parameters transferred between the user and the processing nodes may significantly alter the final result. Additionally, processing nodes shall be monitored in terms of the results they produce in order to mitigate abnormal behaviors such as trying to avoid processing by returning random parameters or even intentionally training the model with artificial data in order to maliciously alter the final result.

In order to establish an integrity mechanism for the behavior of the processing nodes, a modification has to be made to the procedure orchestration, analyzed in Subsection 4.2.2. Specifically the processing nodes are required to transfer to the contract an amount of tokens, as a stake, equal to their possible maximum earnings, during the bid process. Eventually, if the procedure goes on normally, then at the end they are getting paid for their service, according to the commitments made at the bidding phase and they also have their stake returned. If they are detected to act maliciously or abnormally during the process, then they do not get paid, while their stake is paid out evenly to all other participating nodes.

A mechanism to validate if processor nodes are malicious and trying to corrode the process by providing false results or detect they not doing any work at all is to compare the gradients that they return at each training epoch. Individuals processor nodes train their version of the model locally, upon their share of training data, and return the resulting error parameters to the user. Given that the training data points are evenly split among processing nodes, the parameters returned from processing nodes should converge to similar values for all nodes, after some training iterations. Additionally the values returned for these parameters should gradually stabilize for each node. Nodes that return values with high variance, in consequent training epochs, probably misbehave. The user can monitor the parameters returned from each processing node, for each training iteration, and decide upon the criteria mentioned for any misbehaving node.

An alternative mechanism could be applied in cases where the update of the model is conducted upon the average of errors for all data points. Specifically the integrity of the procedure carried out in the data processors side can be tested upon the error results of distinct data points. If the user is able to do some processing on his side, then he can test the initial epoch model with random data points out of the ones distributed to processors and then calculate the hash of the data point along with the produced error. Finally he has to require from the processors to submit which are these random data points. There is no other way for the processor than to do the actual training with all data points until he picks the ones the user has selected. In the case where the user cannot conduct any processing then an alternative approach would be to have partial overlapping of data points distributed to users and utilize the results received from other nodes to test the integrity of each processing node.

Data processors are obliged to transfer to the contract a stake equal to their maximum earnings according to their bid. If they do not act according to the protocol then the stake is lost, so economic loss is a strong incentive for nodes, to perform a valid training procedure.

5 RELATED WORK

There are some recent research efforts in the literature that combine federated learning with either blockchain or privacy preserving methods.

In (Weng et al., 2018) a privacy preserving deep learning distributed network is presented, where blockchain is used for the orchestration of the process. The participating parties do computations locally without exposing their data and then upload the local gradients to a central node to in order to combine those. Blockchain is utilized to offer a direct incentive to users to be behave trustfully and non-maliciously. The system is tested in a local Corda network and interesting results are obtained.

Also in (Mendis et al., 2018) a theoretical decentralized model is presented, in which there are three kinds of nodes. First one is the initiator who proposes the deep learning model and the profit that each node will earn. Apart from that there is the contributor which does the computations and gets tokens as a reward and the verificator who verifies the correctness of the results. No practical implementation is presented.

In (Kurtulmus and Daniel, 2018) a trust less machine learning distributed network is presented where the orchestration of the whole process is done exclusively through Ethereum smart contracts. Nodes compute the models and upload the results to smart contracts for verification. Smart contracts automatically verify the results and pay the required rewards. That creates a machine learning market where users utilize available GPU power to solve machine learning tasks and get profit instead of mining cryptocurrencies.

Our approach, while still in early stages, combines privacy preserving techniques, blockchain technology and integrity checking mechanisms, in order to provide a flexible infrastructure that will enable all users to engage with modern machine learning. We envision an open system where end users can either train machine learning models or offer processing power and data to others. Such a system is not described in any of the previous research efforts.

6 FUTURE WORK

Regarding future work, a detailed privacy and security threat model has to be defined for federated learning schemes. The main actors that may interfere with personal data privacy and the procedure's security are going to identified along with the probable attacks that may be executed by each one of them.

According to this threat model the general architecture of the proposed system will be designed. This will include the definition of nodes taxonomies the collaboration protocol between such nodes along with the main building blocks of the proposed system.

Homomorphic encryption techniques performance will be tested with regards to machine learning models training requirements. Also a large scale federated learning network incorporating blockchain will be developed to test if any scalability issues may arise.

Blockchain technology will also be tested in the context of the proposed system's requirements. It must be seamlessly integrated to the system in order not to hinder the adoption of required users due to usability issues or limited familiarity with blockchain technology.

Lastly integrity violations is a great threat to such an ecosystem. A detailed analysis of the attack vectors is going to be applied, in order to research on all probable issues and design specific counter-measures that can be enforced.

7 CONCLUSIONS

In conclusion, federated learning is a powerful tool for deploying machine learning in distributed networks. Meanwhile, combining blockchain in this architecture makes it even more robust and creates an ecosystem where users can share their data or processing power for monetary reward, which is a great incentive for engaging more users. Also security and privacy measures need to be applied because during the training sensitive and personal data are shared.

In general this is a promising approach that could make recent machine learning technology advancements available to more users. In order for it to be effective, a lot of effort is required for designing and implementing all the required building blocks.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Aono, Y., Hayashi, T., Wang, L., Moriai, S., et al. (2017). Privacy-preserving deep learning: Revisited and enhanced. In *International Conference on Applications and Techniques in Information Security*, pages 100–110. Springer.

Backes, M., Berrang, P., Hecksteden, A., Humbert, M., Keller, A., and Meyer, T. (2016). Privacy in epigenetics: Temporal linkability of microrna expression profiles. In *USENIX Security Symposium*, pages 1223–1240.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., and Shmatikov, V. (2018). How to backdoor federated learning. *arXiv preprint arXiv:1807.00459*.

Bennett, J., Lanning, S., et al. (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA.

Chamatidis, I., Katsika, A., and Spathoulas, G. (2017). Using deep learning neural networks for ecg based authentication. In *Security Technology (ICCST), 2017 International Carnahan Conference on*, pages 1–6. IEEE.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Deng, L., Hinton, G., and Kingsbury, B. (2013). New types of deep neural network learning for speech recognition and related applications: An overview. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8599–8603. IEEE.

Deng, L., Yu, D., et al. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387.

Hitaj, B., Ateniese, G., and Perez-Cruz, F. (2017). Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 603–618. ACM.

Jing, Y., Bian, Y., Hu, Z., Wang, L., and Xie, X.-Q. S. (2018). Deep learning for drug design: An artificial intelligence paradigm for drug discovery in the big data era. *The AAPS journal*, 20(3):58.

Kurtulmus, A. B. and Daniel, K. (2018). Trustless machine learning contracts; evaluating and exchanging machine learning models on the ethereum blockchain. *arXiv preprint arXiv:1802.10185*.

Li, P., Li, J., Huang, Z., Li, T., Gao, C.-Z., Yiu, S.-M., and Chen, K. (2017). Multi-key privacy-preserving deep learning in cloud computing. *Future Generation Computer Systems*, 74:76–85.

Liu, F., Liu, B., Sun, C., Liu, M., and Wang, X. (2013). Deep learning approaches for link prediction in social network services. In *International Conference on Neural Information Processing*, pages 425–432. Springer.

Melis, L., Song, C., De Cristofaro, E., and Shmatikov, V. (2018). Inference attacks against collaborative learning. *arXiv preprint arXiv:1805.04049*.

Mendis, G. J., Sabounchi, M., Wei, J., and Roche, R. (2018). Blockchain as a service: An autonomous, privacy preserving, decentralized architecture for deep learning. *arXiv preprint arXiv:1807.02515*.

Min, S., Lee, B., and Yoon, S. (2017). Deep learning in bioinformatics. *Briefings in bioinformatics*, 18(5):851–869.

Na, S.-H. (2015). Deep learning for natural language processing and machine translation.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Narayanan, A. and Shmatikov, V. (2008). Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE.

Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H. G., and Ogata, T. (2015). Audio-visual speech recognition using deep learning. *Applied Intelligence*, 42(4):722–737.

Orekondy, T., Oh, S. J., Schiele, B., and Fritz, M. (2018). Understanding and controlling user linkability in decentralized learning. *arXiv preprint arXiv:1805.05838*.

Rouhani, B. D., Riazi, M. S., and Koushanfar, F. (2018). Deepsecure: Scalable provably-secure deep learning. In *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE.

Takabi, H., Hesamifard, E., and Ghasemi, M. (2016). Privacy preserving multi-party machine learning with homomorphic encryption. In *29th Annual Conference on Neural Information Processing Systems (NIPS)*.

Weng, J., Weng, J., Li, M., Zhang, Y., and Luo, W. (2018). Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. Technical report, Cryptology ePrint Archive, Report 2018/679. 2018. Available online: https ....

Zeiler, M. D. (2013). *Hierarchical convolutional deep learning in computer vision*. PhD thesis, New York University.

Zhou, Y., Wilkinson, D., Schreiber, R., and Pan, R. (2008). Large-scale parallel collaborative filtering for the netflix prize. In *International Conference on Algorithmic Applications in Management*, pages 337–348. Springer.