Smarter Contracts to Predict using Deep-Learning Algorithms

Syed Badruddoja, Ram Dantu, Yanyan He, Mark Thompson, Abiola Salau, Kritagya Upadhyay
Dept. of Computer Science & Engineering
University of North Texas
Denton, TX, 76207, USA
E-mail: syedbadruddoja@my.unt.edu, ram.dantu@unt.edu, yanyan.he@unt.edu,
mark.thompson2@unt.edu, abiolasalau@my.unt.edu, kritagyaupadhyay@my.unt.edu

2022 Fourth International Conference on Blockchain Computing and Applications (BCCA) | 978-1-6654-9958-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/BCCA55292.2022.9922240

***Abstract*—Deep learning techniques can predict cognitive intelligence from large datasets involving complex computations with activation functions. However, the prediction output needs verification for trust and reliability. Moreover, these algorithms suffer from the model's provenance to keep track of model updates and developments. Blockchain smart contracts provide a trustable ledger with consensus-based decisions that assure integrity and verifiability. In addition, the immutability feature of blockchain also supports the provenance of data that can help deep learning algorithms. Nevertheless, smart contract languages cannot predict due to the absence of floating-point operations required by activation functions of neural networks. In this paper, we derive a novel method using the Taylor series expansion to compute the floating-point equivalent output for activation functions. We train the deep learning model off-chain using a standard Python programming language. Moreover, we store models and predict on-chain with blockchain smart contracts to produce a trusted forecast. Our experiment and analysis achieved an accuracy (99%) similar to popular Keras Python library models for the MNIST dataset. Furthermore, any blockchain platform can reproduce the activation function using our derived method. Last but not least, other deep learning algorithms can reuse the mathematical model to predict on-chain.**

***Index Terms*—Deep-learning security, Blockchain, Smart Contract, Immutability, DApp, On-chain Prediction**

I. Introduction & Motivation

Trustworthy AI: Artificial Intelligence (AI) applications face challenges with data, models, and predictions with poisoning attacks [1]–[4]. The performance of AI applications may not be reliable under such exposure when left unprotected. Moreover, a tampered dataset may produce an incorrect model, and an incorrect model will result in wrong predictions. For instance, deep learning algorithms work with large datasets and build a multi-layer neural network to perform image classification, speech recognition, and many other prediction applications that require trustworthy models and predictions. The training from deep learning algorithms can be unreliable due to the damaged data. Furthermore, attackers can manipulate a model on a central server, producing a fake prediction. Therefore, trust in deep learning applications can be assured with the integrity of data, models, and algorithms.

Data & Model Provenance of AI With Blockchain: Blockchain integration with AI provides provenance of data and model [1], [5], [6]. Provenance refers to the tracking of unaltered data and models that can be relied upon for further development of applications. AI applications require the provenance of information to create explainable AI that ensures the audit-ability of model training. A trained model with provenance maintains accountability that helps identify any underlying learning problems with the algorithm. Finally, the model predicting the unknown data can provide the nature of predictions, explaining the classification. Consequently, AI model provenance with immutability features [7], [8] can facilitate reproducibility to debug a prediction problem.

Automation & Decentralized Voting: The absence of third parties in blockchain applications increases the reliability and trustworthiness of the resulting decisions [1], [9]. Additionally, the automation feature of blockchain allows anyone to train a model and predict without changing the smart contracts defined for the particular AI algorithm. Besides, the Blockchain decentralized platform has a consensus-based decision-making mechanism that does not depend on a single server computation but on multi-server output validation. For instance, swarm robots and deep reinforcement learning are agent-based decision-making systems that borrow the features of automation and decentralization from blockchain [10]–[12].

Model Ownership & Incentivization: One of the significant aspects of current research in AI is the ownership of data and training model [13]–[15]. The data owners can share the data with a cooperative blockchain and retain their credibility in the blockchain. In addition, the model trainers can train their models and be incentivized based on the usage of the models.

Smart Contract Limitations: Smart contracts in blockchain cannot execute floating-point mathematical computations [16], [19]. Moreover, the smart contracts do not support signed exponents, which again limits many exponential operations required by the deep learning algorithms of AI. For example, deep learning algorithms often involve floating-point mathematical computations [20] which cannot train and predict with smart contracts. Additionally, blockchain is facing challenges in scaling applications because of block size limits, delays in output and expensive transactions over time [17], [18]. Furthermore, with increasing demands of decentralized finance requests and complex applications, the verification

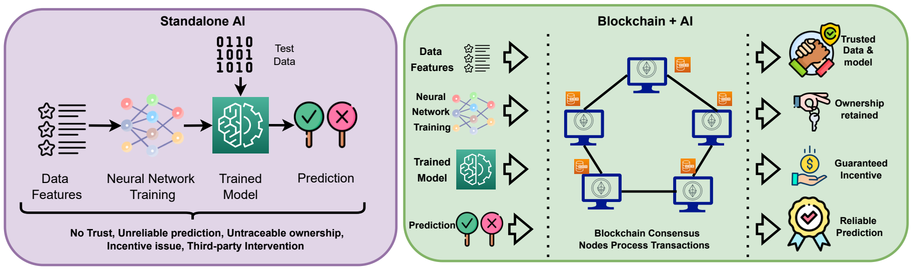

Fig. 1. Comparison of standalone AI and Blockchain integrated AI for deep learning systems

costs and networking costs are rising.

AI Marketplace: AI model marketplace sells models to users to avoid the laborious process of training [21]. Anybody can request a model without training by outsourcing the arduous job. This training requires the provenance of information to maintain trust in the model. For instance, amazon web service (AWS), Genesis AI, and SingularityNet [22] are some of the marketplaces currently available that provide AI marketplace services. However, AWS and Genesis AI marketplace solutions lack a trusted platform to provide model provenance. SingularityNet uses a blockchain platform for storing models but fails to predict on-chain. Figure 1 shows the disadvantages of standalone AI applications and the advantages of blockchain-based AI applications.

II. Problem Definition

Deep learning algorithms are data-hungry approaches to prediction where a slight change in data can impact the model's performance, resulting in unreliable predictions. Besides, third-party involvement in artificial intelligence can cause such manipulation to produce superficial and untrustworthy projections. Moreover, deep learning model development ownership requires a trusted platform to reward owners. Blockchain is one such platform that offers immutability, provenance, and incentivization to help deep learning algorithms with smart contracts. However, there are challenges with floating-point arithmetic operations in smart contracts, which limit the ability of prediction. Moreover, the activation functions involved in deep learning algorithms require the calculation of signed exponents such as sigmoid and softmax, which are not supported by smart contracts. Therefore, deep learning algorithms cannot develop a decentralized application to predict with blockchain smart contracts as per current literature.

III. Our Contributions

- We propose a novel, platform-agnostic, and reusable mathematical method to estimate activation functions output for prediction using Taylor series expansion using smart contracts. The activation function outputs are shown in **Figure 5 and 6**.
- We have tested our smart contract-based prediction for the MNIST digit recognition dataset and achieved almost the same accuracy as the built-in library function available for deep learning applications. See **Figure 7**.
- We have shown that the on-chain prediction is scalable with linearly increasing gas consumption for an increasing number of features and neurons. See **Figure 8 and 9 and 10**.
- We have shown the analysis of the cost of prediction for the MNIST digit recognition dataset in different blockchain platforms. See **Table V**.

IV. Relevant Literature

Trusted Model & Security: Table I shows the recent literature on the requirements of trustworthy AI by National Institute of Standard and Technology (NIST) and the National Artificial Intelligence Institute(NAII). The literature mainly focuses on the failure of AI to hold faith in the intelligence system. Blockchain with distributed ledger technology can make AI applications tamper-proof [1], [6], [24] and trustworthy. Moreover, one of the main contributions of [23] is to create trustworthy machine learning contracts where evaluations of machine learning contracts require an exchange of models. However, the trustworthy application requires blockchain smart contracts that cannot learn and predict with AI algorithms, which limits the efficacy of a trustworthy AI system.

Programming Language Barrier: Neural networks in deep learning algorithms require computations of activation functions that involve exponential operations and divisions. However, as per the latest release of Solidity programming language version 0.8.13 [25], the fixed-point variable cannot compute floating-point mathematical calculations and signed exponents. Fixidity [26], ABDK [27], Decimalmath [32] and PRBMath [31] are some of the recent libraries developed to provide additional floating-point mathematical operations. Nevertheless, these libraries fail to produce exponent computations, thereby limiting the smart contracts to predict with neural networks. Table II provides the summary of the latest fixed-point non-standard libraries. Table III shows the blockchain platforms along with the programming languages that do not allow floating-point computations.

Existing Applications: Many applications are trying to secure AI with blockchain primitives and help blockchain with AI intelligence. In **SingularityNet** [39], AI developers are given a platform to monetize their creations with different

Literature	Year	Author	Purpose
Evaluate trust in AI [28]	2021	NIST	Propose method to evaluate user trust in AI
Trust and AI [29]	2021	NIST	AI system trustworthiness concerns
Trustworthy AI [30]	2021	NAII	AI system trustworthiness concerns

TABLE I
THE TABLE SHOWS THE RECENT LITERATURE ARTICLE FROM THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY, NATIONAL AI INITIATIVE ON TRUSTWORTHY AI SYSTEMS.

Library	Year	Required Mathematical Operation	Function Limitation
PRBMath [31]	2021	Sigmoid/Softmax	Exponents require off-chain computation
Decimalmath [32]	2020	Sigmoid/Softmax	Exponent not supported
ABDK [26]	2019	Sigmoid/Softmax	Negative exponents not supported
Fixidity [27]	2019	Sigmoid/Softmax	Exponents not supported

TABLE II
LIMITATIONS OF NON-STANDARDIZED LIBRARIES TO COMPUTE FIXED POINT CALCULATIONS, NOT SUITABLE FOR COMPLEX DEEP LEARNING PREDICTION

Blockchain	Language	Float Support
Ethereum	Solidity [33]	No
Algorand	TEAL [34]	No
Cardano	Marlowe [35]	No
Polkadot	Solidity [36]	No
NEO	C# [37]	No
Binance	Solidity [33]	No

TABLE III
TABLE SHOWING THE ABSENCE OF FLOATING-POINT COMPUTATION SUPPORT IN POPULAR BLOCKCHAIN SMART CONTRACT LANGUAGES.

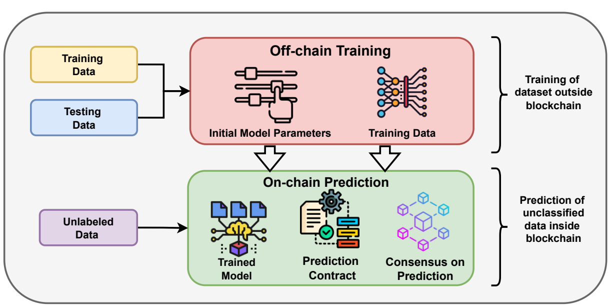

Fig. 2. AI models developed off-chain and stored on-chain. A smart contract with the model can predict on-chain with sigmoid and softmax activation functions through our solutions

AI techniques. For instance, an AI developer can create a node on the blockchain platform and start offering and receiving tasks for monetizing their work. **Coin.AI** [38] is another theoretical work that emphasizes validating a deep learning model on exceeding a performance threshold with a proof-of-useful-work consensus mechanism. Moreover, **Raven protocol** [40] introduces the incentivization benefits of anyone willing to share their computing resources to train compute-intensive deep learning tasks. Furthermore, **Cortex** [41] is an AI-Blockchain platform allowing storage of models on-chain for inference and incentivizing the model creators in the process. Anytime the model needs training or retraining, the model updates with off-chain training in the proprietary cortex machines called CVM (Cortex Virtual Machine). Additionally, the Cortex AI stores models to incentivize the model owners. However, these contributions fail to establish a trusted prediction system using smart contracts.

V. METHODOLOGY

Design Overview: Our design prepares blockchain to predict with neural network algorithms to offer trusted prediction with reliable data. Data is stored in a distributed files system that assures data integrity by creating a hash value. We deploy an interplanetary file system(IPFS) to record hashes of data sets. The trustworthy AI design consists of a training and prediction phase. First, the AI developer trains data outside blockchain with a neural network algorithm and produces a model with reasonable accuracy. Secondly, we prepare a smart contract to compute the sigmoid and softmax activation function where a user request prediction to classify objects. Since smart contracts can only use integer operations, we derive a numerical method with Taylor's series expansion to produce floating-point equivalent output for prediction. The smart contracts ensure the integrity of the model and prediction output in the prediction phase. Moreover, our design also offers provenance of the model for future development and incentivization for prediction on the blockchain network. Figure 2 shows a high-level plan of off-chain training and an on-chain prediction scheme. Figure 3 shows the event flow between the model developer and prediction requester for our solution.

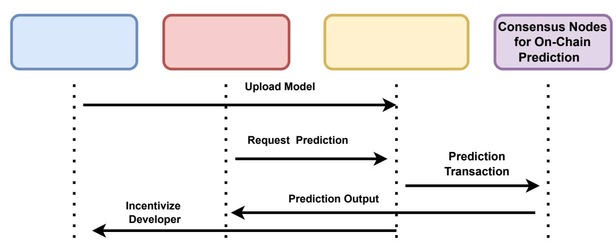

Fig. 3. Event flow between a model developer and prediction requester for on-chain prediction requests.

Neural Network: We consider a basic neural network design with an input layer, an output layer, and a hidden layer [42]. The input layer consists of data input, while the hidden layer consists of neurons with sigmoid activation functions and the output layer consists of the same number of neurons as the required output classes with softmax activation function (10 neurons for digit recognition). A neural network is trained off-chain, and a final set of weights and biases are stored on-chain through the smart contract. These weights and biases with forward propagation of our neural network predict the

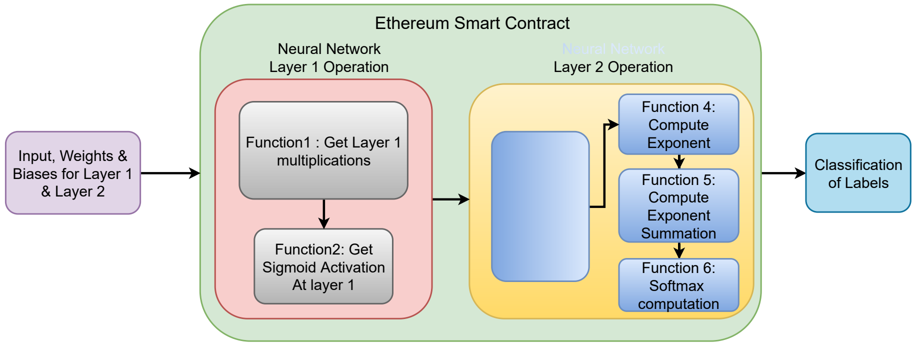

Fig. 4. Block diagram representing the flow of data in smart contract function where layer one operations calculate hidden layer multiplication with sigmoid activation and layer two operations calculate output layer multiplication with softmax activation.

Algorithm	Function	Reusability
Convolutional Neural Network	Sigmoid/Softmax	Yes
Recurrent Neural Network	Sigmoid	Yes
Reinforcement Learning	Softmax	Yes
Naive Bayes	Gaussian Probability	Yes

TABLE IV
RE-USABILITY MATRIX SHOWING THAT OUR DERIVED METHOD CAN BE USED FOR FURTHER PREDICTION SERVICES FOR FUTURE DEVELOPMENTS IN OTHER ALGORITHMS.

class of AI task in the smart contract. For this, we derived a novel method to compute sigmoid and softmax functions without floating-point computations, which are discussed later. The forward propagation in a smart contract uses the linear combination of weighted features (with weights *W1*), adds biases (*B1*), and computes sigmoid output at the hidden layer (Layer one in Figure 4). Later, the hidden layer output multiplies the second set of weights *W2*, adds a second set of biases (*B2*), and computes softmax (Layer two Figure 4). Finally, argmax function (maximum of an array) returns final outcome of prediction [44]. Figure 4 shows the layer one process that obtains the first set of sigmoid output, and layer two functions compute softmax activation and predict the final classification of labels.

Consensus Protocol:**Proof of Stake(PoS)** [45] consensus protocol PoS save energy and still provide consensus with security for blockchain networks. With PoS, the consensus protocol is more robust in security, ensuring persistence, liveness, safety, and openness. **Persistence** defines the immutability of blocks. **Liveness** means that honest validators in a blockchain can create a block given that the participants are honest. **Safety** refers to the fact that all honest participants have the same data. Finally, **Openness** states that anyone can participate in the consensus process.

Sigmoid Activation Function For sigmoid activation function [43], we used Taylor series expansion to approximate sigmoid output and obtain the sigmoid output in fraction format. Equation 1 shows the formula for sigmoid activation. Algorithm 1 shows the pseudo-code for sigmoid activation.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

For $x >= 0$, we consider $\frac{e^x}{1+e^x}$; while for $x < 0$, we calculate $\frac{1}{1+e^{|x|}}$. Therefore, the major calculation of the sigmoid function is the calculation of the exponential of a positive number. Rewrite $|x|$ to be a summation of an integer q and a fraction r/b with an absolute value smaller than one as $|x| = a/b = q + r/b$. Then the exponent of $|x|$ can be written as $e^{|x|} = e^q e^{r/b}$. For the term with integer exponent, we approximate the value of *e* with 19/7, and consequently, $e^q \approx 19^q/7^q$. For the terms with fraction exponent, we approximate it using five-term Taylor expansion as follows:

$$e^{r/b} = 1 + \frac{r}{b} + \frac{(r/b)^2}{2!} + \frac{(r/b)^3}{3!} + \frac{(r/b)^4}{4!}. \tag{2}$$

With simple arithmetic operations, $e^{|x|}$ can be written as a fraction formula depending on q, r, b with both numerator and denominator requiring only integer operations.

$$e^{|x|} = \frac{19^q(24b^4 + 24rb^3 + 12r^2b^2 + 4r^3b + r^4)}{24 * 7^q b^4} \tag{3}$$

Consequently, the sigmoid function can be represented as a fraction supporting integer operations for both numeration and denominator, with its output in fraction format. Algorithm 1 provides the logic of sigmoid activation function.

Softmax Activation Function Algorithm 2 provides the logic of softmax computation. Softmax is an activation function [43] used to classify labels when there is a multi-class prediction system. Softmax defines the probability of each class and predicts the most probable class to be the predicted outcome. Equation 4 shows the formula for softmax activation.

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{k} e^{x_j}} \tag{4}$$

For $x_i >= 0$, as shown in Section V, with the help of Taylor expansion, the exponential function of x_i (and consequently the softmax function) can be written as a fraction formula depending on q, r, b with both the numerator and denominator requiring only integer operations. In algorithm 2, first we

Algorithm 1 Sigmoid Activation

function SIGMOID (a,b) ▷ getting input in fraction with x as numerator and y as denominator
  if $(a/b \geq 1)$ **then**
    $q \leftarrow a/b$
    $r \leftarrow a\%b$
    $num \leftarrow 19^q * 24b^4$
    $den \leftarrow 19^q * 24b^4 + 7^q * (24b^4 - 24rb^3 + 12r^2b^2 - 4br^3 + r^4)$
  else if $(a/b == 0) \wedge (a * b > 0)$ **then**
    $num \leftarrow 24b^4$
    $den \leftarrow 48b^4 - 24rb^3 + 12r^2b^2 - 4br^3 + r^4$
  else if $(a/b = 0) \wedge (a * b < 0)$ **then**
    $num \leftarrow 24b^4$
    $den \leftarrow 48b^4 + 24rb^3 + 12r^2b^2 + 4br^3 + r^4$
  else
    $q \leftarrow a/b$
    $r \leftarrow a\%b$
    $num \leftarrow 7^q * 24b^4$
    $den \leftarrow 7^q * 24b^4 + 19^q * (24b^4 + 24rb^3 + 12r^2b^2 + 4br^3 + r^4)$
  end if
  return num, den
end function

compute the exponents(e^x) with $EXPONENT$ function for all input values of array x represented in fraction form of array a as numerator values and array d as denominator values. Next, we compute the sum of all the exponents with the $SUM\_EXPONENT$ function. Finally, we compute the $Softmax$ with the $GET\_SOFTMAX$ function and predict the class.

Expected Performance Analysis According to our proposed design, we have performed prediction tests on unclassified data sets to compare the performance accuracy of neural network prediction with smart contracts concerning the existing Keras [65] library function (A Python library for deep learning algorithms). Furthermore, we have compared the outputs of sigmoid and softmax activation to check our proposed solution implemented within smart contracts. Apart from these, we have studied the cost analysis of the Ethereum smart contract concerning computations performed for on-chain prediction.

VI. Experimental Setup

Dataset: We have chosen the MNIST digit recognition dataset for our experiment [46]. The MNIST dataset has 784 features and 60000 samples with ten classes for digit classification problems. The training parameters of the neural network model (i.e., weights and biases) are converted to integer format with scalar multiplication. Dataset is divided into 50,000 training samples and 10,000 testing samples. Moreover, for testing our method, 101 random samples between 5.0 to

Algorithm 2 Softmax Activation

function EXPONENT (Array a,Array b) ▷ array "x" is represented as fraction with array "a" as numerator and array "b" as denominator where x[i] = a[i]/b[i]
  $N \leftarrow length(a)$
  for $i \leftarrow 1$ to N **do**
    $q \leftarrow a[i]/b[i]$
    $r \leftarrow a[i]\%b[i]$
    if $(q \leq 1)$ **then**
      $num \leftarrow 7^q * 24b[i]^4$
      $den \leftarrow 19^q * (24b[i]^4 + 24rb[i]^3 + 12r^2b[i]^2 + 4b[i]r^3 + r^4)$
    else if $(q == 0) \wedge (a[i] * b[i] > 0)$ **then**
      $num \leftarrow 24b[i]^4 - 24rb[i]^3 + 12r^2b[i]^2 - 4b[i]r^3 + r^4$
      $den \leftarrow 24b[i]^4$
    else if $(q == 0) \wedge (a[i] * b[i] < 0)$ **then**
      $num \leftarrow 24b[i]^4$
      $den \leftarrow 24b[i]^4 + 24rb[i]^3 + 12r^2b[i]^2 + 4b[i]r^3 + r^4$
    else
      $num \leftarrow 19^q * (24b[i]^4 + 24rb[i]^3 + 12r^2b[i]^2 + 4b[i]r^3 + r^4)$
      $den \leftarrow 7^q * 24b[i]^4$
    end if
    $z[i] = num/den$ ▷ array z store output of this function
  end for
  return z
end function
function SUM_EXPONENT (Array z) ▷ This function adds exponents which takes z as an array of exponents
  $N \leftarrow length(z)$
  for $i \leftarrow 1$ to N **do**
    $sum \leftarrow sum + z[i]$
  end for
  return sum
end function
function GET_SOFTMAX (Array z, sum) ▷ This function returns the classification output with z as an array of exponents and sum as the summation of all exponents
  $N \leftarrow length(z)$
  $maxn \leftarrow length(z)$ ▷ Numerator of softmax
  $maxd \leftarrow length(z)$ ▷ Denominator of softmax
  for $i \leftarrow 1$ to N **do**
    $maxn[i] \leftarrow z[i]$
    $maxd[i] \leftarrow sum$
  end for
  $predicted\_class \leftarrow argmax(maxn, maxd)$ ▷ output index of greatest of fractions maxn[item]/maxd[item]
  return $predicted\_class$
end function

-5.0 for sigmoid activation tests and 41 samples between +20 to -20 for softmax activation tests are generated.

Software Framework: The software framework of our experiment involves a blockchain network, a Metamask plug-in, ethers from the Ethereum faucets, Ethereum smart contracts with required functions, and a Python framework for handling data flow between the components. We have developed six smart contract functions for our experiment: one for sigmoid activation, three for softmax activation, and two for neural network layer-wise multiplication-addition. The three smart contract functions in softmax computation involve the computation of exponential values, the summation of exponents, and finding the final value of softmax output.

Blockchain Network: To test our hypothesis, we have considered Ethereum local blockchain Ganache-cli and the public test net Ropsten test network to conduct experiments. The local blockchain Ganache-cli is for testing and stabilizing our transactions with the fair output accuracy of our proposed work. Moreover, smart contracts are deployed on Binance smart chain [49] and Polygon [48] to generalize the cost of predictions. Our project implementation is available in GitHub [47].

VII. Performance Results

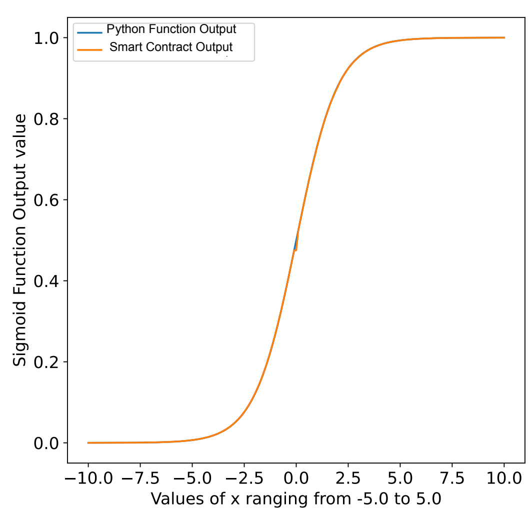

Fig. 5. Output of Sigmoid Activation Function for 101 values ranging from +5.0 to -5.0 with intervals of 0.1 inside Smart Contract shows similar values for Python function and smart contract function

Smart Contract Activation Function Output: From figure 5, we can see that the derived sigmoid activation function with the smart contract has almost the same output as the Python library-based computations. Moreover, the softmax activation function in smart contracts is even more accurate and produces the same output as non-smart contract library functions, as shown in figure 6. The curve forms an exponential behavior as expected by the softmax activation function. The blue curve of the library function is not seen due to the exact values produced by smart contract output.

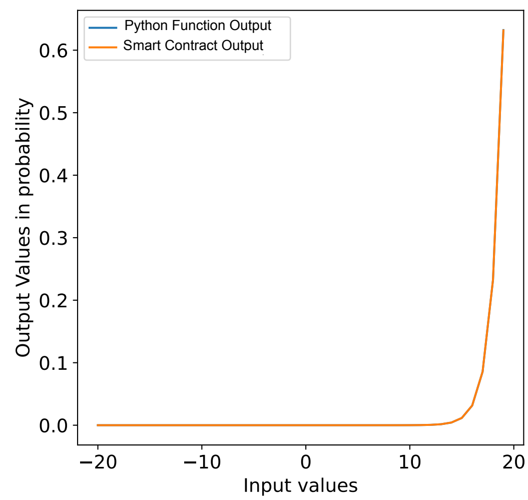

Fig. 6. Output of softmax activation function for 41 values ranging between -20 to +20 of inside smart contracts produce similar values to Python function

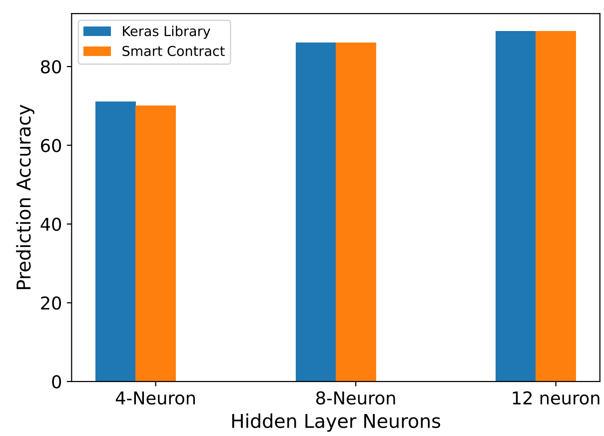

Fig. 7. Prediction accuracy of Keras library-based prediction and on-chain smart contract prediction for MNIST dataset for 10,000 samples

On-Chain Prediction Accuracy: One of the major contributions of this work is to produce reasonable accuracy for predicting a neural network-based prediction task. Figure 7 shows the prediction accuracy of smart contract-based functions that produced 71 percent accuracy for the 4-neuron trained model, 86 percent accuracy for the 8-neuron trained model, and 89 percent accuracy for the 12-neuron trained model. Our model produces an accuracy of 99 percent compared to the Python Keras library.

Prediction Gas Usage : Figure 8 shows the rise of gas consumption with the number of input features for multiplying inputs with weights and adding biases. The line equation given in the graph estimates the gas consumption for more components in this setup. Moreover, feature scalability depends on the block gas limit to create a block. For the Ethereum Ropsten test network, the creation of a block is restricted to a gas consumption of 30,000,000 Gwei [51]. Provided we do not exceed the gas limit, the layer one multiplication would run

Smart Contract Function	Ethereum Mainnet	Ethereum Testnet	Binance Testnet	Polygon Testnet
Layer 1 Computation	0.005263	0.00473	0.06282	0.004739
Sigmoid Activation	0.00241	0.000167	0.00062	0.000167
Layer 2 Computation	0.00037	0.00068	0.00309	0.000687
Softmax Activation	0.00077	0.00069	0.00151	0.000401
Total Cost	0.023413	0.010687	0.085868	0.010725

TABLE V
COST OF PREDICTION WITH SMART CONTRACTS CONSIDERING 4 NEURONS FOR THE HIDDEN LAYER AND 10 NEURONS FOR THE OUTPUT LAYER FOR MNIST DIGIT CLASSIFICATION DATASET WITH 784 FEATURES

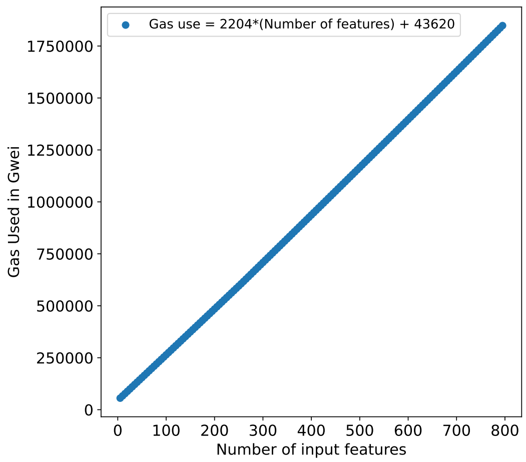

Fig. 8. Gas consumption of layer one operation (Hidden layer computations) with an increasing number of features and fixed number of neurons(4 neurons).

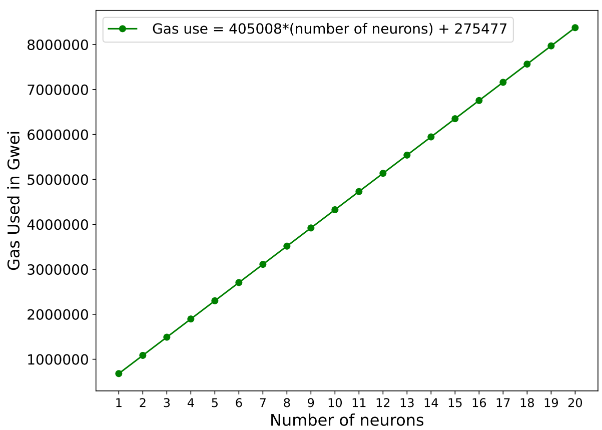

Fig. 9. Gas consumption of neural network with a rising number of neurons (Hidden layer computations) and fixed number of features(784 features)

without any obstruction. Figure 9 shows the gas consumption with a rising number of neurons. The gas consumption is linear, and the equation provides estimated gas for many features.

We tested our hypothesis for scalability with the multi-hidden-layer neural network with two hidden layers. Figure 10 shows a 3D plot of the rising gas consumption of this architecture with an increasing number of neurons. When the number of neurons grows on both the x and y-axis, multiplication complexity rises, resulting in higher gas consumption along the z-axis. This trend is expected to follow and grow with more hidden layers as per our study.

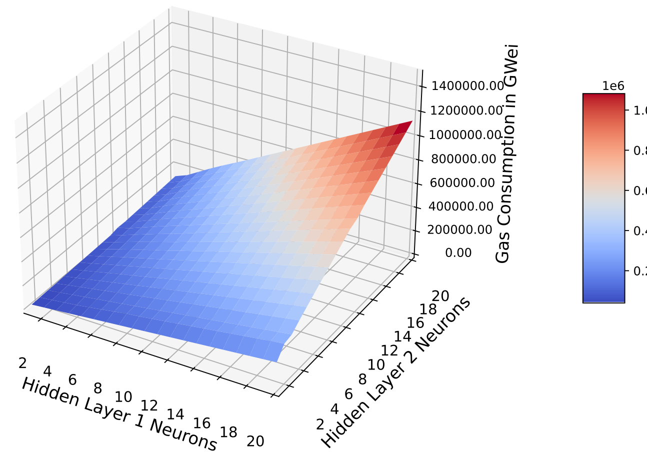

Fig. 10. 3D plot showing gas consumption for multi-layer computations concerning different units of neurons for two hidden layers

Cost Analysis: Table V shows the cost of prediction for a single classification inside a smart contract with different blockchain platforms. While Ethereum Ropsten and polygon networks have an average price of 0.010 $ether$ and $matic$, the main network has 0.023 $ether$. However, the Binance smart chain has an average cost of 0.085868 bnb, which is higher than the rest of the test networks.

VIII. LIMITATION AND CHALLENGES

Integer Overflow: Since we used fraction equivalence of exponent values, which is of the form 19/7, the exponent of this fraction will be a very high number depending on the power value. An Ethereum smart contract using Solidity language has a limitation of integer representation with 256 bits for a signed and unsigned integer [50]. That means the value of a signed integer will range from $-2^{256}/2$ to $+2^{256}/2$. An unsigned integer value ranges from 0 to about 2^{256}. Any value that goes beyond this range cannot be used for measuring softmax computations.

Block Gas Limit: The block gas limit of the Ethereum Ropsten test network limits the number of computations that can be performed in a single block. As of the current Etherscan website, report [52], the Ropsten test network can create a block with a gas limit of 30,000,000. Anything over this gas consumption needs to be transacted in a different block.

Multiple Block Creation: With our current deployment and smart contract functions, each of the predictions inside the blockchain smart contracts will require 6 blocks of information: one for layer one multiplication input, one for layer two multiplication, 1 for sigmoid activation, and 3 for softmax multiplication. With an increase in the prediction count, the ledger will increase with 6 blocks each time as each of these transactions are part of individual blocks.

IX. Conclusion

Deep learning prediction services lack trust, provenance, immutability, accountability, and security. The blockchain-based deep learning prediction can provide a solution for untampered prediction, ownership retainment, and consensus-based transactions that can address the security concerns of deep learning. However, the smart contracts programming languages in blockchain technology are not designed to handle complex activation functions required by deep learning algorithms. We derived a platform-agnostic novel method to compute the activation function with Taylor series expansion to estimate the classification of an unknown dataset. Our derived method produced an excellent accuracy and can be reused in other deep learning algorithms for predictions.

X. Future Work

Off-chain Model Validation: Off-chain training needs to be secured under a blockchain platform for better trust and security. For off-chain training, we need to explore scalable blockchain solutions with faster transactions and cheaper computational costs. There are many scalability platforms such as sharding [53], optimistic and zero-knowledge proof roll-ups [54], plasma [55], and validium [50] that can provide the training of our algorithm at a little cost and time. We plan to implement the off-chain training component on these scalable solutions in our future work.

Other Deep Learning Techniques: Our current work studies the specific requirements concerning individual layers and activation functions of deep learning models. In our subsequent works, we aim to study different deep learning models such as convolutional neural networks, recurrent neural networks, and reinforcement learning. We expect these models would involve more intense implementation than our current work.

XI. Acknowledgments

We thank National Security Agency for the partial support through grants H98230-20-1-0329, H98230-20-1-0403, H98230-20-1-0414, and H98230-21-1-0262.

References

[1] R. Shinde et al., “Blockchain for securing AI applications and open innovations,” Journal of Open Innovation: Technology, Market, and Complexity, vol. 7, no. 3, p. 189, 2021.

[2] Shafay et al. (2022). Blockchain for deep learning: review and open challenges. Cluster Computing, 1-25.

[3] Tariq et al., (2020). A review of deep learning security and defensive privacy techniques. Mobile Information Systems, 2020.

[4] Comiter, M. Attacking Artificial Intelligence AI’s Security Vulnerability and What Policymakers Can Do About It. 2019. Available online: https://www.belfercenter.org/publication/AttackingAI

[5] P. by Jerry Cuomo, “How blockchain adds trust to AI and IoT,” 2020. [Online]. Available: https://www.ibm.com/blogs/blockchain/2020/08/howblockchain-adds-trust-to-ai-and-iot/

[6] K. Sarpatwar, R. Vaculin, H. Min, G. Su, T. Heath, G. Ganapavarapu, and D. Dillenberger, “Towards enabling trusted artificial intelligence via blockchain,” in Policy-based autonomic data governance. Springer, 2019, pp. 137–153

[7] F. P. Hjalmarsson et al., ”Blockchain-Based E-Voting System,” 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), 2018, pp. 983-986, DOI: 10.1109/CLOUD.2018.00151.

[8] T. Alladi et al., ”Blockchain Applications for Industry 4.0 and Industrial IoT: A Review,” in IEEE Access, vol. 7, pp. 176935-176951, 2019, DOI: 10.1109/ACCESS.2019.2956748.

[9] D. Campbell, “Combining ai and blockchain to push frontiers in healthcare,” Nov 2018. [Online]. Available: https://www.macadamian.com/learn/combining-ai-and-blockchain-in-healthcare/

[10] S. Janson et al., “A decentralization approach for swarm intelligence algorithms in networks applied to multi Swarm PSO,” International Journal of Intelligent Computing and Cybernetics, vol. 1, no. 1, pp. 25–45, Jan 2008. [Online]. Available: https://doi.org/10.1108/17563780810857112

[11] K. Hassan et. al., “On relative-output feedback approach for group consensus of clusters of multiagent systems,” IEEE Transactions on Cybernetics, pp. 1–12, 2021.

[12] D. Magazzeni, P. McBurney, and W. Nash, “Validation and verification of smart contracts: A research agenda,” Computer, vol. 50, no. 9, pp. 50–57,2017.

[13] Sriraman, A., Bragg, J., & Kulkarni, A. (2017, February). Worker-owned cooperative models for training artificial intelligence. In Companion of the 2017 ACM Conference on computer supported cooperative work and social computing (pp. 311-314).

[14] N. Baranwal Somy et al., ”Ownership Preserving AI Market Places Using Blockchain,” 2019 IEEE International Conference on Blockchain (Blockchain), 2019, pp. 156-165, DOI: 10.1109/Blockchain.2019.00029.

[15] Salau, A., Dantu, R., & Upadhyay, K. (2021, May). Data Cooperatives for Neighborhood Watch. In 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 1-9). IEEE.

[16] J. D. Harris and B. Waggoner, ”Decentralized and Collaborative AI on Blockchain,” 2019 IEEE International Conference on Blockchain (Blockchain), 2019, pp. 368-375, DOI: 10.1109/Blockchain.2019.00057.

[17] Catalini, C., & Gans, J. S. (2020). Some simple economics of the blockchain. Communications of the ACM, 63(7), 80-90.

[18] W. Gao, W. G. Hatcher and W. Yu, ”A Survey of Blockchain: Techniques, Applications, and Challenges,” 2018 27th International Conference on Computer Communication and Networks (ICCCN), 2018, pp. 1-11, DOI: 10.1109/ICCCN.2018.8487348.

[19] Battah, A., Iraqi, Y., & Damiani, E. (2021). Blockchain-based reputation systems: Implementation challenges and mitigation. Electronics, 10(3), 289.

[20] S. Marchese, ”AI Chips Must get The Floating-Point Math Right”, https://semiengineering.com/artificial-intelligence-chips-must-get-the-floating-point-math-right/#: :text=Floating%2Dpoint%20representations%20of%20real,of%20values%20without%20losing%20precision., Retrieved March 2022

[21] Sarpatwar, K., Vaculin, R., Min, H., Su, G., Heath, T., Ganapavarapu, G., Dillenberger, D. (2019). Towards enabling trusted artificial intelligence via blockchain. In Policy-based autonomic data governance (pp. 137-153). Springer, Cham.

[22] Sandro Luck, ”3 AI Marketplaces Everyone Has To Know [One Will Define The Century]”, https://towardsdatascience.com/3-ai-marketplaces-everyone-has-to-know-one-will-define-the-century-a4295d4f0229

[23] Kurtulmus, A. B., & Daniel, K. (2018). Trustless machine learning contracts; evaluating and exchanging machine learning models on the Ethereum blockchain. arXiv preprint arXiv:1802.10185.

[24] Wang, T., Wu, X., & He, T. (2019). Trustable and automated machine learning running with blockchain and its applications. arXiv preprint arXiv:1908.05725.

[25] ”Solidity Language Documentation”, https://docs.soliditylang.org/en/v0.8.13/, Retrieved March 2022

[26] ”Fixidity Library” , https://github.com/CementDAO/Fixidity, Retrieved March 2022

[27] "ABDK Libraries For Solidity", https://github.com/abdk-consulting/abdk-libraries-solidity, Retrieved March 2022
[28] NIST, https://www.nist.gov/news-events/news/2021/05/nist-proposes-method-evaluating-user-trust-artificial-intelligence-systems, 2021
[29] NISTIR, https://nvlpubs.nist.gov/nistpubs/ir/2021/NIST.IR.8332-draft.pdf, 2021
[30] NAAI, https://www.ai.gov/strategic-pillars/advancing-trustworthy-ai/: :text=To %20be%20trustworthy%2C%20AI%20technologies,ensure %20that%20bias%20is%20mitigated.
[31] "PRBMath library" , https://github.com/paulrberg/prb-math, Accessed July 2022
[32] "Decimalmath",https://github.com/alcueca/DecimalMath, Accessed July 2022
[33] "Solidity Smart Contract language Documentation", https://docs.Soliditylang.org/en/v0.8.14/
[34] "TEAL Smart Contract Language Documentation", https://developer.algorand.org/docs/get-details/dapps/avm/teal /specification/
[35] "Marlowe Smart Contract Language Documentation" https://plutus-apps.readthedocs.io/en/latest/marlowe/tutorials/marlowe-data.htmlmarlowe
[36] "Polkadot Smart Contract Language Documentation", https://wiki.polkadot.network/docs/build-smart-contracts
[37] "Neo Smart Contract Language Documentation", https://docs.neo.org/docs/en-us/develop/write/basics.html
[38] Baldominos, A., & Saez, Y. (2019). Coin. AI: A proof-of-useful-work scheme for blockchain-based distributed deep learning. Entropy, 21(8), 723.
[39] Singularitynet, White paper , https://public.singularitynet.io/ whitepaper.pdf
[40] Raven Protocol, White Paper, https://drive.google.com/file/d/1FA aVKkg_CjxMj-n1yHZc6ufcVDtOU1Ct/view
[41] Cortex- AI on Blockchain, Whitepaper, https://cryptorating.eu/whitepapers /Cortex/Cortex_AI_on_Blockchain_EN.pdf
[42] M.Ardi, Simple Neural Network on MNIST Handwritten Digit Dataset", https://becominghuman.ai/simple-neural-network-on-mnist-handwritten-digit-dataset-61e47702ed25, Retrieved March 2022
[43] S. Polamuri, "Difference Between Softmax Function And Sigmoid Function", https://dataaspirant.com/difference-between-softmax-function-and-sigmoid-function/, Retrieved March 2022
[44] Agarwal, S., "Argmax and softmax", https://medium.com/@s_hash_wat/argmax-and-softmax-496714956aab, Retrieved April 2022
[45] "Ethereum whitepaper", https://ethereum.org/en/developers /docs/consensus-mechanisms/pos/: :text=on%20consensus %20mechanisms.-,What%20is%20proof%2Dof%2Dstake%20(PoS) %3F,a%20smart%20contract %20on%20Ethereum.
[46] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6), 141–142.
[47] Implementation code, https://github.com/syber2020/Python-NN-SC.git
[48] Polygon lightpaper, https://polygon.technology/lightpaper-polygon.pdf, Retrieved March 2022
[49] Binance Whitepaper, https://polygon.technology/lightpaper-polygon.pdf, Retrieved March 2022
[50] Ethereum White Paper, https://ethereum.org/en/developers/docs/ scaling/state-channels/,Retrieved March 2022
[51] "Etherscan latest block gas limits", https://ropsten.etherscan.io/blocks, Retrieved April 2022.
[52] Ropsten Etherscan transactions, https://ropsten.etherscan.io/blocks, Retrieved April 2022.
[53] Zamani, M., Movahedi, M., & Raykova, M. (2018, October). Rapidchain: Scaling blockchain via full sharding. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (pp. 931-948).
[54] Schaffner, T. (2021). Scaling Public Blockchains. A comprehensive analysis of optimistic and zero-knowledge rollups. University of Basel.
[55] Poon, J., & Buterin, V. (2017). Plasma: Scalable autonomous smart contracts. White paper, 1-47.
[56] NIST, https://www.nist.gov/blogs/taking-measure/zero-trust-cybersecurity-never-trust-always-verify, 2020
[57] Badruddoja, S., Dantu, R., He, Y., Upadhayay, K., Thompson, M. (2021, May). Making smart contracts smarter. In 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 1-3). IEEE.
[58] Badruddoja, S., Dantu, R., Widick, L., Zaccagni, Z., Upadhyay, K. (2020, May). Integrating DOTS With Blockchain Can Secure Massive IoT Sensors. In 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) (pp. 937-946). IEEE.
[59] Upadhyay, K., Dantu, R., Zaccagni, Z., Badruddoja, S. (2020, November). Is your legal contract ambiguous? Convert to a smart legal contract. In 2020 IEEE International Conference on Blockchain (Blockchain) (pp. 273-280). IEEE.
[60] Upadhyay, K., Dantu, R., He, Y., Salau, A., Badruddoja, S. (2021, December). Make Consumers Happy by Defuzzifying the Service Level Agreements. In 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA) (pp. 98-105). IEEE.
[61] K. Upadhyay, R. Dantu, Y. He, A. Salau and S. Badruddoja, "Paradigm Shift from Paper Contracts to Smart Contracts," 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), 2021, pp. 261-268, doi: 10.1109/TPS-ISA52974.2021.00029.
[62] K. Upadhyay, R. Dantu, Y. He, S. Badruddoja and A. Salau, "Can't Understand SLAs? Use the Smart Contract," 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), 2021, pp. 129-136, doi: 10.1109/TPS-ISA52974.2021.00015.
[63] A. Salau, R. Dantu, K. Morozov, K. Upadhyay, S. Badruddoja, "Multi-Tier Reputation for Data Cooperatives", The 3rd International Conference on Mathematical Research for Blockchain Economy, 2022
[64] A. Salau, R. Dantu, K. Morozov, K. Upadhyay, and S. Badruddoja (2022). Towards a Threat Model and Security Analysis for Data Cooperatives. In Proceedings of the 19th International Conference on Security and Cryptography - SECRYPT, ISBN 978-989-758-590-6; ISSN 2184-7711, pages 707-713. DOI: 10.5220/0011328700003283
[65] Gulli, A., Pal, S. (2017). Deep learning with Keras. Packt Publishing Ltd
[66] Mark Robins, Publish Date: May 27, "The difference between Artificial intelligence, Machine learning and Deep learning." [Online]. Available: https://www.intel.la/content/www/xl/es/artificial-intelligence/posts/difference-between-ai-machine-learning-deep-learning.html
[67] S. Gupta, Publish Date: January 16, 2018, "Deep Learning Performance breakthrough" [Online]. Available: https://www.ibm.com/blogs/systems/deep-learning-performance-breakthrough/
[68] K. Salah et al., "Blockchain for AI: Review and open research challenges," IEEE Access, vol. 7, pp. 10 127–10 149, 2019.
[69] Zheng et al., (2021). Agatha: Smart Contract for DNN Computation. arXiv preprint arXiv:2105.04919.
[70] Liu et al.,(2021). Proof of Learning (PoLe): Empowering neural network training with consensus building on blockchains. Computer Networks, 201, 108594.
[71] Zapotochny. A, "101 On Deep Learning + Blockchain [A Brief Introduction]", https://blockgeeks.com/101-on-deep-learning-blockchain-a-brief-introduction/, Accessed March 2022
[72] da Cruz et al.,, (2020, May). Blockchain-based Traceability of Carbon Footprint: A Solidity Smart Contract for Ethereum. In ICEIS (2) (pp. 258-268).
[73] Otoum, S., Al Ridhawi, I., & Mouftah, H. T. (2020, December). Blockchain-supported federated learning for trustworthy vehicular networks. In GLOBECOM 2020-2020 IEEE Global Communications Conference (pp. 1-6). IEEE.
[74] Zhang et al., (2020). Blockchain-based federated learning for device failure detection in industrial IoT. IEEE Internet of Things Journal, 8(7), 5926-5937.
[75] ur Rehman et al.,(2020, July). Towards blockchain-based reputation-aware federated learning. In IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 183-188). IEEE.
[76] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," 2017 IEEE International Congress on Big Data (BigData Congress), 2017, pp. 557-564, DOI: 10.1109/BigDataCongress.2017.85.