International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

ANN Based Block Chain Security Threat Mechanism

Pranab Kumar Bharimalla, Souratendu Praharaj, Satya Ranjan Dash

Abstract*: Blockchain was first introduced to the world in a form of crypto currency. Since then it's grabbing popularity day by day and has already started changing lifestyle and business process in some areas. The Blockchain is a data structure, and consists of time-stamped list of blocks, used to create a digitally secured, transactional ledger that, facilitates the process of recording transactions and tracking assets in a distributed business network instead of resting with a single provider. It brings a paradigm shift from the conventional way of storing transactional data, exchanging value, assets decentralizing and sharing data across a large network of untrusted participants without intervention of any centralized trusted agency, but still building a trusted system without compromising data integrity, security and reliability. This paper is based on survey of different consensus mechanisms already available so far and an introduce our own approach to tack one of the security threats by introducing a Neural Network to the consensus mechanism of Blockchain.

***Index Terms*: BlockChain, Cryptocurrency, Security Threats, Double Spending, Deep Learning**

I. INTRODUCTION

BlockChain as its name suggests is a chain of blocks containing information. This technique was first originally described by Stuart Haber and W. Scott Stornettaback in 1991. The original purpose of BlockChain was intended to timestamp digital documents so it would not be possible to tamper with them. This concept of BlockChain went by completely unused until an alias named "Satoshi Nakamoto" implemented it for creating a digital cryptocurrency named Bitcoin in 2009. In 2014, BlockChain went from just being used to implement digital cryptocurrencies to implement more complex financial offerings like stocks, bonds and contracts, thus ushering in the era of BlockChain 2.0. BlockChain 2.0 introduced smart contracts which are code snippets that can also be added to the blocks. Recently BlockChain has expanded from financial applications to diverse fields like Healthcare, charity, et al with the introduction of building of Decentralized applications using BlockChain called Dapps.

BlockChain is a Distributed Ledger that is completely open to anyone in the network. One of the most important properties of a BlockChain is that once a block had been added to the BlockChain, it is very difficult to change the data in it. A Block consists of a number of valid transactions which are hashed and encoded into a merkle tree and added to the block. It also contains the hash of the previous block. This way the BlockChain can be traced back all the way to its Genesis Block (First Block of the BlockChain) using the previous hash. The blockchain always aims to maintain the highest possible length of the blockchain. So, if one block is modified then the hash of the BlockChain changes thus making the subsequent block contain an invalid hash, thus making all the subsequent blocks invalid which would reduce the size of the blockchain. As the blockchain aims to maintain the longest possible chain it would revert back to the last stable state.

The transactions in a blockchain network happen in a Peer to Peer setting. There is no reliance on the third party involved. BlockChain does this by having each node in the network maintain a copy of the ledger, i.e. the entire blockchain. To start a transaction, the sender broadcasts a message to all the nodes called miners with link to it all previous transactions of the sender. The nodes verify if the sender has enough balance and then compete to solve a complex mathematical problem to add the new transaction into the blockchain. The mathematical problem involves finding a random number called nonce which satisfies a predefined condition for the hash generated for the transaction. The solution can easily be verified by other nodes as the nonce has already been found. Once the transaction is verified by at least 51% of the miners, it is added to the blockchain as a block.

Let us look at the most important features of a blockchain:

1. SHA256

Blockchain Technology uses a key Hash algorithm named SHA256. This algorithm ensures that once a piece of data is encrypted it cannot be decrypted back. Another key feature of this algorithm is that it produces a hash of the same size for chunks of different size or different types of data. Even a slightest change will alter the complete hash generated hence it is impossible to reverse engineer a generated hash. Let's have a look at the below example (Fig 1) for a better understanding. In the above example, the boxes on the left side are very similar texts and the boxes on the right side are hashes generated by SHA256 Hash Function from the texts in the left side boxes as the source. As you can see from the first and the second example, the first letter of the word was just changed from upper case to lower case, still the hash generated from both those texts vary completely. This hash function is an integral part of the consensus mechanisms used in blockchain which will be explained few sections later.

Manuscript published on 30 August 2019.
*Correspondence Author(s)
Pranab Kumar Bharimalla, School of Computer Science and Engineering, KIIT, Deemed to be University, Bhubaneswar, INDIA.
Souratendu Praharaj, Infosys Limited, Bhubaneswar, INDIA
Satya Ranjan Dash, School of Computer Applications, KIIT, Deemed to be University, Bhubaneswar, INDIA

Fig 1 : SHA256 Hashing

2. Public Key – Private Key

Each user in the blockchain network has a Public Key and a Private Key. The public key can be considered as an identity for the user that is shared with everyone. The private key is a kept only by the respective user. When data is transferred from person A to person B, the miner has to first verify if it is a valid transaction. Before sending the data to the network, person A encrypts the hash of the data using the private key. The miner has to verify the transaction by following two steps:

a. Get all the unencrypted data (e.g. "Hello" in the below figure) and feed it to the hash function to generate a hash.
b. Use the public key to decrypt the encrypted data and get a hash value.

If both the hashes generated by both of the above mentioned steps are same, the transaction can be considered as valid. Have a look into the below figure for better understanding.

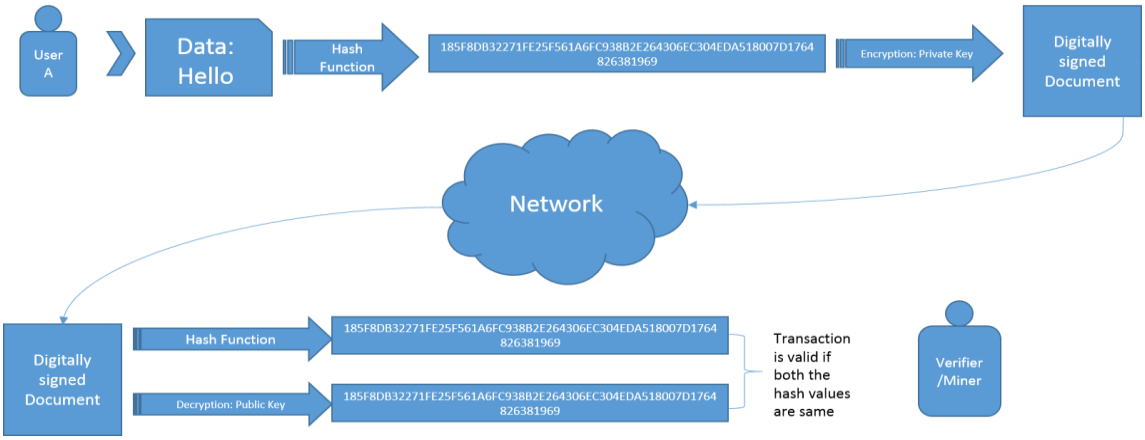

Fig 2: Public key private key : An illustration

3. Consensus Mechanisms

The base of the blockchain network is that it is a trustless network and a third party is not required to verify the transactions. But, there has to be some sort of consensus among all the nodes in a network so that a transaction can be considered valid and added to the chain. This task is performed by miners. The miners verify blocks of transactions which are then again verified the rest of the nodes and if at least 51% of the nodes agree then the transaction is added to the chain. There are many consensus mechanisms available in the market today, we are going to throw some light on the two most popular ones.

a. Proof of Work

Proof of work involves solving a complex mathematical puzzle. Finding a solution of the puzzle is very difficult but verifying the solution is comparatively easy. Solving the puzzle requires a lot of computational effort and once the puzzle is solved the miner broadcasts the results to the rest of the network where it can be verified by the other miners. Once the solution is verified by at least 51% of the miners, the new block is added to the blockchain in the entire network.

The mathematical puzzle involves finding a specific number called *nonce* which can produce a predetermined number of trailing zeros for a hash value. The source for this hash value involves hash of the previous block, hash of the data in the current block, and the nonce. The miner keeps looping through until the nonce value which satisfies the predetermined number of trailing zeros is satisfied. Currently it takes around 10 minutes to solve a given puzzle. The value of the number of trailing zeros is adjusted as the computing power of nodes increases. This process ensures that better computing power does not mean that the miner has to do less work.

Proof of work is the oldest consensus mechanism since the advent of blockchain technology. Bitcoin uses proof of work as its consensus mechanism.

b. Proof of Stake

To complete one given transaction using POW consensus mechanism it takes at least 10 minutes of time. This feature of a POW makes micro-transactions very expensive. One of the faster alternatives to Proof of Work is Proof of Stake. In Proof of Stake, the validity of a user's work is not verified by computational power but rather by the amount of stake held by the particular node.

Each node has to deposit a cryptocurrency amount in a separate wallet, the node which gets to add a new block is decided randomly. But, the more the stake held by a node the more chances the node has of getting selected. The node receives a small transaction fees as incentive.

Retrieval Number: J94420881019/19©BEIESP
DOI: 10.35940/ijitee.J9442.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

There are two variations of proof of stake available in the market.

i.Leased Proof of Stake

In proof of stake if only the nodes with more stake have the best chances of adding a block and receiving incentives, the rest of the users with lesser amounts of stake to join the network. As an incentive, the small stake holders can lease their stakes to large stake holders. The Large stake holders have complete control of the stakes leased and after every successful addition of the blocks, the small stake holders receive a proportionate amount of incentives.

ii. Delegated Proof of Stake

In this variation of Proof of stake, instead of leasing, each node gets vote who gets to add a block in the blockchain. This allows each node to influence and have a greater control of the network.

II. SECURITY THREATS IN BLOCKCHAIN

in two-column format, including figures and tables.

a. Vulnerability

All the major consensus mechanisms used in Blockchain have a risk of 51% vulnerability. In a Proof of Work based Blockchain when a single miner has computation power which is equal to or more than the computational power of the entire network then 51% vulnerability attack can be launched. The miner can take control of the new entire chain, manipulate transactions, double spend the resources, etc. With the control of the entire network, the dominant miner can also steal assets from others.

In January 2014, Ghash.io which is a major mining pool for bitcoin reached 42% computational power of the entire network [1]. Many of the miners started dropping from the mining pool and in 24 hours the computational power of the mining pool dropped to 38% of the total power. In October 2016, the Ghash.io was closed. Currently mining pools like BTC.com and AntPool have the highest mining hash rate, having 23.9% and 15% network hash rate respectively [2].

Similarly, in a Proof of Stake based Blockchain when a single miner/verifier owns more than 50% percent of the total number of coins present in the network.

b. Eclipse attack

In contrast to the above section, if you think that a proof of concept based blockchain is secure as long as 51 percent of mining power is honest, then it's not correct always. This assumption holds well as long as all parties see valid transactional blocks.

A blockchain is a distributed network and relies on its own peer-to-peer network to deliver its information. Consequently, if you control the peer-to-peer network, you control the information flow, and subsequently, can control the blockchain. Hence an attacker can lunch 51 percent attack even with less computing power. To eclipse a node [14], an attacker can manipulate the node so that all its outgoing connections are to attacker IPs. This can be achieved simply by filling the node's peer tables with attacker IPs, force the node to restart and then loses its current outgoing connections. Finally, the node makes new connections only to the attacker IPs.

In a block chain network if β is the fraction of the mining power that is eclipsed then, eventually there is a decrease of total mining power of the chain and increase on attackers mining power. It can be depicted in below equation [15].

$$\alpha' = \frac{\alpha}{1-\beta}$$

α' = Attacker's mining power after eclipse attack

α = Attacker's original mining power

c. Loss or Theft of Private Key

Blockchain provides security to any user's personal information, but the security of the user's identity depends upon the safekeeping of the Private Key of the user. Once the Private Key of a user is lost or stolen, no third party can recover it. If stolen by anyone with malicious intent, the private key can be used to tamper with the Blockchain information. As the Blockchain is maintained as a decentralized system, once stolen it would be difficult to track down the activities performed using the private key.

In December 2014, a Data Scientist who goes by the name "Johoe" managed to find a way to deduce a way to find a predict a private key of some users. Doing so, he managed to transfer funds in the wallets of the users whose private key "Johoe" had predicted to his own account [3].

Hartwig et al. [4] found a vulnerability in Elliptic Curve Digital Signature Algorithm by which an attacker can recover a user's private key stating that the algorithm did not generate enough randomness.

d. Double Spending

Double spending is a type of security loophole which can be easily exploited in most Proof of work based blockchains. Confirmation of one transaction can take some time in a Proof of work based Blockchain. During this period the attacker can initiate another transaction with the same coins. Before the second transaction can be verified as invalid, the attacker would have already received the resource for the first transaction thus leading to double spending.

Another popular technique of double spending is to mine a separate fork secretly. The attacker first makes a payment to an innocent payee, which is mined and added to the main chain. The user initiates another transaction to a second payee account which he has control of. The user mines the transaction secretly himself and does not broadcast it to the rest of the chain. The honest payee waits for N new blocks and sends the product. Once the attacker receives the product, he broadcasts the secret chain which is now longer than the original chain, thus making the original chain invalid. The attacker receives the gets his coins back and also receives the product without having to spend any coins.

Recently many techniques have been proposed to stop attackers to find ways to double spend. Xingjie Yu [5] et al proposes a technique of using Fair deposits in Bitcoin network, which an attacker will lose if any attempt to double spend is made.

III. RELATED WORK

1. Use of Hybrid blockchain with two consensus mechanisms – Proof of Stake was introduced to be used over Proof of Work to save energy and time. POS is a good consensus mechanism option for many type of applications.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

But in case of contract management, use of POS may not be that useful. The participants are more concerned about getting their contracts added to the chain than the value of the coins. In such cases, the attackers are likely to attack because their main objective is to edit or add new contracts in the chain and they have no concern about the value of the coins they have held as stake.

Watanabe et al. [6] propose the use of credibility instead of coins as stakes as the deciding factor if a miner is to be allowed to add new blocks. Credibility here can be defined as the number of number of valid contracts a person has with different people. The higher the number of contracts, the higher the credibility score.

Use of credibility still creates an issue when an attacker creates fake contracts with fictitious users which can drastically improve the credibility score of the attacker. Thus, an attacker with a very high fake credibility score can succeed in an attempt of 51% attack. By doing so the attacker can easily renew any other valid contracts illegally.

Watanabe et al. [6] further propose the use of a hybrid blockchain where Proof of Stake and Proof of Credibility are both used as consensus mechanisms. Each new block added to the chain are alternatively verified by Proof of Stake and Proof of Credibility. If some miner adds a block to the chain using Proof of Stake, the next block must be added using a high credibility score and vice versa. Thus Proof of stake requires that coins are stored which can be then used as stakes, whereas Proof of credibility requires use of coins for creating various contracts. Using and Storing coins are two contradictory ideas, hence it is very difficult to increase both. Thus, 51% attack becomes much more difficult while using this new hybrid consensus mechanism.

Hawk – Smart contracts over decentralized systems allow distrustful parties to perform transactions without the need of any trusted third party. In case of any contractual breaches or aborts, the decentralized system ensures that that the honest party receive proper compensation and dishonest parties are fined. However, most such systems lack transactional privacy. Even though the participants in the transaction can use pseudonyms to maintain their privacy, still the amount transacted in the contracts are still visible to the public. Cryptocurrencies like Zerocash [7] have made some development on preserving privacy, but they do so by trading off programmability.

Kosba et al. [8] propose *Hawk*, a decentralized smart contract system that does not store financial transactions in the clear on the blockchain, thus retaining transactional privacy from the public's view. A Hawk programmer can write a private smart contract in an intuitive manner without having to implement cryptography, and the Hawk compiler automatically generates an efficient cryptographic protocol where contractual parties interact with the blockchain, using cryptographic primitives such as zero-knowledge proofs.

A Hawk program will contain two parts:

i. *Private part:* The private part of the Hawk program takes input from the users as well as currency units. It also performs computations and manages the payment to the respective users as dictated in the smart contract code.

ii. *Public part:* The public part of the Hawk program does not touch the private data or the currency units.

The Hawk program compiles the smart contract into three executables:

i. The blockchain part, which must be executed by all the nodes

ii. The users part, which is executed by all the users taking part in the transaction.

iii. astly, a part that must be executed by a special facilitating party called the manager.

Hawk covers two aspects of security:

i. On-Chain Privacy: For maintaining contractual fairness, the users in a Hawk program do interact with the blockchain. However, the input data and transaction amounts that are shared with the blockchain are encrypted. The blockchain relies on methodologies like zero knowledge proofs to maintain contractual fairness. Thus, the user's data and transactional history is kept as public unless the users themselves voluntarily disclose the information.

ii. Contractual security: The Hawk compiler assumes that each contractual participant is bound to act selfishly. Hence, when any case of cheating or aborting prematurely occurs, the Hawk program penalizes the cheating participant and distributes the penalty among the Honest participants left in the contract.

Interactive Incontestable Signature for Transactions Confirmation in Bitcoin Blockchain – One of the biggest issues prevalent in the current blockchain technology is block conflict. There is a probability that when couple of blocks are published simultaneously, it might lead to the creation of a fork, i.e. the chain starts to grow with two branches. In the current blockchain, this issue is resolved by keeping the branch that is the longest with the blockchain. When one transaction ends up in the side which is not the longest, it takes lot of time to get added to the chain again. This leads to delay in transaction confirmation. It is assumed in blockchain that a transaction can be said to be confirmed after six new blocks have been added to the chain. Even though this mechanism ensures transaction confirmation, it makes the blockchain prone to double spending attack. Waiting for six new blocks to be added can also take a lot of time as it might take around one hour for six new blocks to be added to the blockchain.Yan-Zhu et al. [9] proposes a new system for exact confirmation of transactions in a block. Replacing original signature, a new Interactive Incontestable Signature (IIS) scheme is used between dealer and owner to confirm a transaction. By this signature, the dealer can assure the owner that a transaction will be included into blockchain in a non-repudiation way. The scheme has been proved to be secure for owner's unforgeability and dealer's incontestability. The above-mentioned proposal aims to build a system with exact confirmations and have faster performance than any existing blockchain systems. To achieve so, the transaction needs to get feedback from the block generator once it has been verified. The block generator/verifier cannot deny the transaction once it has been added to the blockchain. Their system is built on the existing bitcoin blockchain with few modifications. In one accounting cycle (time interval between the generation of two new blocks) in this new proposed system, only one dealer is elected to generate a block.

Another modification is that once a block is added to the blockchain, it cannot be reorganized. This new system also adds a new part in each block called *Witness.* It contains the identity of the dealer. Once the transaction is verified a tag is generated and attached with the *Witness* to the transaction which can be verified by any node. Thus, by comparing the tag and the *Witness*, incontestability of the dealer is ensured.

2. Bluewallet – With the growth in popularity of bitcoin, there has been a rapid growth of malicious attacks to steal Bitcoins from other users. A study by Litke and Stewart [10] shows that the amount of cryptocurrency-stealing malware has increased with the popularity of Bitcoin. To execute transactions each user is allocated with a Bitcoin address. Each address is associated with a private key and public key. The users should always be in possession of their respective private keys to perform any sort of transactions. Even though Bitoin itself is protected by strong cryptographic algorithms, attackers have managed to steal Bitcoins by gaining access to the victim's private key. The private key is stored in the user's computer or mobile device, or any other device the user performs Bitcoin transactions with. As the device is often connected to internet, it is prone to malware and spyware attacks.

Tobias Bamert et al. [11] created a Bitcoin hardware token: BlueWallet. The device communicates using Bluetooth Low Energy and can securely sign Bitcoin transactions. The device can also be used as an electronic wallet in combination with a point of sale and serves as an alternative to cash and credit cards.

Bluewallet stores the private key of the user and can be connected to any transaction performing hardware by Bluetooth. The private key cannot be accessed from the Bluewallet token until the correct PIN is entered. The computer (or any other device) can create the unsigned transaction and it can be verified in the end using the private key stored inside Bluewallet. As the private key is not stored by the device connected to internet any more, it can never be stolen using any malware or spyware attacks.

Bluewallet has also been designed to be used in another application. The transaction will be created by an untrusted third party and BlueWallet acts as an electronic wallet. An example of this third party could be the point of sale in a store, a restaurant or any other place where one would normally pay with cash, debit or credit card. Even though the transaction will be created by an untrusted party, additional security measures have been implemented in BlueWallet to minimize the risk incurred by the user. In addition to the signing ability of BlueWallet, the user may review and authorize the transaction independently from the point of sale and BlueWallet has to ensure that only the authorized bitcoins are transferred.

3. Demystifying Incentives in the Consensus Computer - Next-generation cryptocurrencies such as Ethereum have introduced a Turing-complete script language which allows users to encode pieces of code into the blockchain and support a variety decentralized applications. The large number of miners on the cryptocurrency network, who both execute and verify computational tasks, reach agreement through an established consensus protocol. Therefore, these miners are collectively referred as verifiers, and the computation framework of scriptable cryptocurrencies are referred to as consensus computers. Miners have two separate functions in the consensus computer: checking that blocks are correctly constructed, or proof-of-work, and checking the validity of transactions in each block. While verifying correct block construction requires a relatively small amount of work (two SHA256 calculations), checking the validity of transactions contained in a block can take much more time for two reasons. First, the number of transactions per block may be large. Second, expressive transaction scripts in emerging cryptocurrencies such as Ethereum can require significant computational work to verify. These expressions create a new dilemma for miners — whether the miners should verify the validity of scripted transactions or accept them without verification. Miners are incentivized to verify all scripted transactions for the "common good" of the cryptocurrency so to speak. However, verifying scripts consumes computation resources and therefore delays honest miners in the race to mine the next block. We argue that this dilemma leaves open the possibility of attacks which result in unverified transactions on the blockchain. This means that some computation tasks outsourced to cryptocurrency-based consensus computers may not execute correctly.

Loi Luu et al. [12] have described various attacking techniques that can be used by attackers to exploit the vulnerability of verifiers mentioned above and use them for their own profit.

i. Resource Exhaustion attacks by problem givers- Honest miners will always verify all the transactions in new block that is broadcasted to them. Thus, if an attacker broadcasts his expensive transactions to the network, the honest miner will spend a lot of power and time to verify the transaction. Such attacks not only exhausts other miners' resource, but also gives the attacker some time ahead of other miners in the race to mine the next block.

ii. Incorrect transaction attack by provers – Due to the existence of the attack mentioned above. Honest miners feel they are more incentivized if the skip

iii. expensive transaction in the race to generate next blocks. In such a scenario, any malicious prover can include an expensive but wrong scripts in the contract which the Honest miners will chose to not verified but mark as verified. This makes the Blockchain system open to any mischievous activities like transferring coins from contracts wallet to one's own wallet and so on.

Loi Luu et al. [12] have further proposed a solution which can be used to prevent the above mentioned attacks. They have proposed a model which will incentivize miners to correctly verify all the transactions by limiting the amount of work required to verify all transactions in a block. They do so by providing an upper bound in the advantage that miners get by deviating from the honest protocol. This means honest miners are also guaranteed not to run long and expensive scripts while verifying the transactions.

IV. OUR APPROACH

Consensus mechanisms are used to prove the reliability of a miner/verifier, so that he/she is allowed to add a new block to the chain.

In real world scenarios such tasks are accomplished by are decision by particular person or a group of persons. Initially when none of the users are known, users with maximum stakes are chosen. But, as time passes users can be chosen based on their reliability. Reliability depends on many factors like number of valid transactions, period of usage of the particular service, etc. We can build on the above mentioned real world mechanism by using machine learning algorithms to learn of Blockchain transactions which uses Leased proof of stake initially but gradually uses a hybrid model based on leased POS and deep learning. We can build a neural network which can take various parameters of a particular miner/verifier as an input and predict the probability of a miner/verifier to add a valid block to the Blockchain. The higher this probability, the higher the probability the miner/verifier gets a chance to add a block to the chain.

The small investors can still lease their stakes to a bigger investor (not just with a higher stake but also with higher reliability) thus on successful transaction they will still receive respective proportion of the transaction fees. The neural network has to be maintained by all the nodes and before any new block is added each investor should share a detail of investment amount, amount leased and the ID of the user leasing that amount. Once a new block is added all the nodes can be updated with the properties like total leased amount, total successful leased amounts which got returns, total investments, successful investments and so on. May other factors will also influence the input layer of the artificial neural network built. Here is a snapshot of the proposed ANN.

The initial days of the Blockchain would be used just to train the ANN, after a certain period the testing phase of the Blockchain will start. In this phase, with each new block added the results would be compared with the probability and the error is back propagated to adjust the weight. Once global minima for the error is reached, the third phase can start. In this phase, the higher the probability predicted by the Artificial Neural Network, the greater the chance for the node to add a new block. The node with the highest probability will not be selected as this will result in a particular node getting to add a block every time. Instead top nodes with highest probabilities are grouped and one of them is selected using random selection. Thus the nodes can involve in various profitable activities for the Blockchain and increase their probability and compete.

Various factors also need to be added as inputs to the ANN (Artificial Neural Network) such that a node that has attained a high probability loses points if it sits idle for long. This can be done by adding a parameter named last successful block verified. This way, each node has to stay active and are rewarded for successful transactions. Any malicious user with intent to use more stake will have to spend a lot of time and resources to get a chance to add a block.

We have chosen the deep learning algorithms above all other machine learning algorithms as it has been proven to outperform every other algorithm in the long run. The issue with the above mentioned approach is that, one has to begin with a traditional leased POS consensus mechanism and train the neural network with parameters of each successful and failed attempt to add a new block. As the neural network starts to predict outcomes more accurately, the consensus mechanism can then be switched to the hybrid Artificial Neural Network consensus mechanism.

Algorithm:

i. Phase 1(Training Phase):

a. Each competing node share details of stake, all the nodes who have leased to the investing node and their leased stake amounts with other competing nodes.
b. Higher the stake higher the chances to be selected to add a block.
c. Each node trains the ANN with initial weights. Back-propagate to adjust weights for every 10 new blocks.
d. Once block is added, Request to update properties for each node (competing and leasing). (Note: The properties need to be cryptographically stored in each node such that it cannot be changed by the node manually.)
e. Each node verifies the block and accepts the request to update its properties.

ii. Phase 2(Testing Phase)

a. Each competing node share details of stake, all the nodes who have leased to the investing node and their leased stake amounts with other competing nodes.
b. Higher the stake higher the chances to be selected to add a block.
c. Each node trains the ANN. Back-propagate to adjust weights for every 10 new blocks. Check error for each 100 new blocks and validate if global minima are maintained. Once the global minima for error is returned, the consensus mechanism is switched to deep learning model and the Blockchain moves to Phase 3.
d. Once block is added, Request to update properties for each node (competing and leasing). (Note: The properties need to be cryptographically stored in each node such that it cannot be changed by the node manually.)
e. Each node verifies the block and accepts the request to update its properties.

iii. Phase 3(Deep Learning Prediction phase)

a. Each competing node share details of stake, all the nodes who have leased to the investing node and their leased stake amounts with other competing nodes.
b. Higher the probability returned by the ANN higher the chances to be selected to add a block.
c. Each node trains the ANN. Back-propagate to adjust weights for every 10 new blocks. Check error for each 100 new blocks and validate if global minima are maintained. If any massive fluctuations in error is observed, switch the Blockchain to phase 2 into Leased POS consensus mechanism.
d. Once block is added, Request to update properties for each node (competing and leasing). (Note: The properties need to be cryptographically stored in each node such that it cannot be changed by the node manually.)
e. Each node verifies the block and accepts the request to update its properties.

Retrieval Number: J94420881019/19©BEIESP
DOI: 10.35940/ijitee.J9442.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

By using the above mentioned approach, we can gradually remove the security vulnerability of a POS Blockchain where a miner/verifier can invest more stake intentionally to add invalid block. In this approach, the miner/verifier has to maintain a high probability which is predicted by the neural network. As the network grows the neural network becomes more and more efficient thus reducing the chances of exploitation of above mentioned vulnerability of the POS Blockchain.

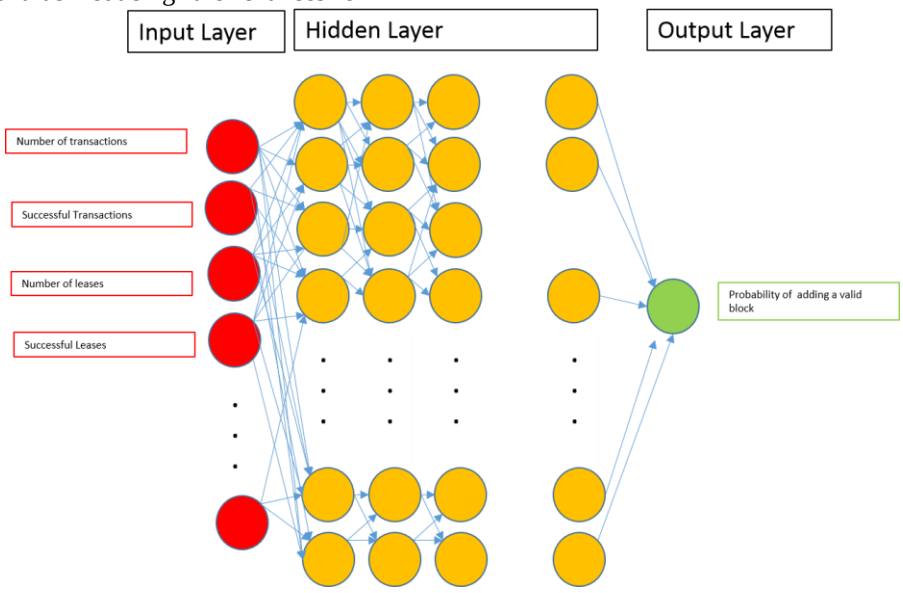

Fig 3: Proposed ANN Model

V. CONCLUSION

This paper started with a basic introduction to Blockchain and its many features. We gave a brief description to each feature of Blockchain. We proceeded by explaining various security vulnerabilities in Blockchain technology. We further explained few of the related works done on reducing the vulnerability of Blockchains to security threats. Then we proceeded to introduce our own approach to tack one of the security threats by introducing a Neural Network to the consensus mechanism of Blockchain. We further described our idea of implementing the Neural Network in a Blockchain such that it can be trained to predict if a node is likely to a valid node. We would further work to build a new Blockchain which can implement this aforementioned approach and be much more secure than the Blockchain's of the current time.

REFERENCES

1. Bitcoin Miners Ditch Ghash.io Pool Over Fears of 51% Attack. https://www.coindesk.com/bitcoin-miners-ditch-ghash-io-pool-51-attack/ .
2. Hashrate Distribution. https://blockchain.info/pools
3. Good Samaritan' Blockchain Hacker Who Returned 267 BTC Speaks Out. https://www.coindesk.com/good-samaritan-blockchain-hacker-returned-255-btc-speaks/
4. Mayer, H. (2016). ECDSA security in bitcoin and ethereum: a research survey. CoinFaabrik, June, 28.Yu, X., Shiwen, M. T., Li, Y., & Huijie, R. D. (2017, August). Fair deposits against double-spending for Bitcoin transactions. In Dependable and Secure Computing, 2017 IEEE Conference on (pp. 44-51). IEEE.
5. Yu, X., Shiwen, M. T., Li, Y., & Huijie, R. D. (2017, August). Fair deposits against double-spending for Bitcoin transactions. In Dependable and Secure Computing, 2017 IEEE Conference on (pp. 44-51). IEEE.
6. Watanabe, H., Fujimura, S., Nakadaira, A., Miyazaki, Y., Akutsu, A., & Kishigami, J. (2016, January). Blockchain contract: Securing a blockchain applied to smart contracts. In Consumer Electronics (ICCE), 2016 IEEE International Conference on (pp. 467-468). IEEE.
7. Sasson, E. B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., & Virza, M. (2014, May). Zerocash: Decentralized anonymous payments from bitcoin. In Security and Privacy (SP), 2014 IEEE Symposium on (pp. 459-474). IEEE.
8. Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016, May). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In Security and Privacy (SP), 2016 IEEE Symposium on (pp. 839-858). IEEE.
9. Zhu, Y., Guo, R., Gan, G., & Tsai, W. T. (2016, June). Interactive incontestable signature for transactions confirmation in bitcoin blockchain. In Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual (Vol. 1, pp. 443-448). IEEE.
10. Litke, P., & Stewart, J. (2014). Cryptocurrency-stealing malware landscape. Technical report.
11. Bamert, T., Decker, C., Wattenhofer, R., & Welten, S. (2014, September). Bluewallet: The secure bitcoin wallet. In International Workshop on Security and Trust Management(pp. 65-80). Springer, Cham. Luu, L., Teutsch, J., Kulkarni, R., & Saxena, P. (2015, October). Demystifying incentives in the consensus computer. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 706-719). ACM.
12. Luu, L., Teutsch, J., Kulkarni, R., & Saxena, P. (2015, October). Demystifying incentives in the consensus computer. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (pp. 706-719). ACM.
13. Heilman, E., Kendler, A., Zohar, A., & Goldberg, S. (2015, August). Eclipse Attacks on Bitcoin's Peer-to-Peer Network. In USENIX Security Symposium (pp. 129-144).
14. Nayak, K., Kumar, S., Miller, A., & Shi, E. (2016, March). Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In Security and Privacy (EuroS&P), 2016 IEEE European Symposium on (pp. 305-320). IEEE.
15. Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., & Capkun, S. (2016, October). On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 3-16). ACM.

AUTHORS PROFILE

Pranab Kumar Bharimalla is working at Infosys as Sr Project Manager and currently pursuing Ph.D at KIIT, Deemed to University, in Compute Science. He passed his master's degree from same institution. His research interests are cloud computing, block chain and security with application into healthcare.

Souratendu Praharaj is a master's student at University at Buffalo. He is focusing on Internet of Things as his major. He is also working as a research volunteer in Wireless Intelligent Network and Security lab at University at Buffalo. He got his Bachelor's from Biju Patnaik University of Technology in Computer Science and Engineering. He has 4+ years of experience as a web and application developer in Infosys Limited. Souratendu has developed multiple applications benefiting local startups. Souratendu has also been involved in freelancing projects for Indian Oil Corporation to build and Employee Information portal. His interests include Internet of Things, Blockchain and Machine Learning.

Satya Ranjan Dash is an Associate Professor in School of Computer Applications, KIIT Deemed to be University, Bhubaneswar, India. He received his MCA degree from Jorhat Engineering College, Dibrugarh University, Assam and M.Tech. degree in Computer Science from Utkal University, Odisha. He received his Ph.D. in Computer Science from Utkal University, Bhubaneswar, INDIA. His research interest includes Machine Learning, Biomedical Image processing and Cloud Computing. He is also working on Neural Machine Translation and Natural Language Processing.