Long-Range Attack detection on Permissionless Blockchains using Deep Learning

Olanrewaju Sanda[1], Michalis Pavlidis[1], Saeed Seraj[1], Nikolaos Polatidis[1]*
[1]School of Architecture, Technology and Engineering, University of Brighton, BN2 4GJ, Brighton, U.K

Email addresses: O.Sanda@Brighton.ac.uk, M.Pavlidis@Brighton.ac.uk, S.Seraj@Brighton.ac.uk, N.Polatidis@Brighton.ac.uk

*Corresponding author

Abstract

Blockchain has been viewed as a breakthrough and an innovative technology due to its privacy, security, immutability, and data integrity characteristics. The consensus layer of the blockchain is the backbone and the most important layer of the blockchain architecture because it acts as the performance and security manager of the blockchain. The detection of Long-Range Attacks (LRA) on the Proof-of-Stake (PoS) blockchain is a complex task. Earlier studies have shown various challenges in detecting long-range attacks and monitoring the activities of validator nodes on the blockchain network. Thus, this paper proposes a novel dataset for node classification on a proof-of-stake permissionless blockchain and proposes a Deep Learning method that can be used to classify nodes into malicious or non-malicious nodes to mitigate long-range attacks with high accuracy. The performance metrics for the model are compared and measured which suggest the developed performance of the proposed model. The proposed solution can serve as a guide on how future researchers and blockchain developers can simulate and curate proof-of-stake datasets and goes further to demonstrate that artificial intelligence models can be used as a mitigating checkpoint for long-range attacks. The dataset in the paper is publicly available and can be used by other researchers to detect other activities and behaviors on a permissionless blockchain. These techniques can further enhance security, performance and create fairness on the proof-of-stake consensus.

Keywords: Blockchain, Proof-of-Stake, Block Validation, Long-Range Attack detection, Dataset, Deep Learning.

1. Introduction

Blockchain was built with the intention of creating a robust distributed database that eliminates the need for a central authority. It has become one of the emerging technologies of the decade and has been used in healthcare, voting, supply chains, finance, internet of vehicles (IoV), art and many more sectors (Gemeliarana et al., 2018). The data on the blockchain database is shared by all participants which are called nodes but cannot be modified by any node (Deirmentzoglou et al., 2019). The validation and authentication of data by the nodes on a blockchain is controlled by an engine called the consensus mechanism. Consensus in a broader context refers to a generally accepted interest, values, opinion, or ideas among a group of people in a community that focuses on eliminating corruption and create data integrity among participants (Nguyen et al., 2019).

In this paper we focus on the Proof-of-Stake (PoS) consensus where blocks and transactions are validated based on the staked-coins of participating nodes as opposed to the Proof-of-Work (PoW) where blocks are validated based on the mining power of participating nodes. PoS consensus was developed as a

solution to the extensive energy consumption, hardware requirements, block generation speed required by PoW blockchain based systems (Gazi et al., 2018). The PoS consensus was developed as a fair system that considers the resources (tokens) of participants to be block validators and earn a stake-reward as opposed to their computational power for block validation as in the PoW consensus. The coins (tokens) for the PoS are created at the initial stage of developing the blockchain and participants can be penalized for bad behavior by deducting their staked coins. Many blockchain applications are now adopting the PoS consensus because of its unique sustainability credentials and low energy consumption. For example, Ethereum which is the second largest cryptocurrency has recently made complete shift from the PoW to the PoS consensus mechanism which will cut energy usage by almost 99% (Akbar et al., 2021).

This has set the stage for the mass adoption of PoS consensus in many other sectors like healthcare and supply-chain management. The principle behind the PoS consensus mechanism is simple; the right and reward for producing a block will not depend on the computational power of nodes but on the majority stake (native tokens) owned by them. This replaces the "honest majority hashing power" model for PoW with the "honest majority stake" model for PoS (Nguyen et al., 2019). The weight on real-world resources such as the energy required to create a block and add this to the chain of existing blocks will be greatly reduced (Houy, 2014). In the PoS consensus, blocks are said to be minted, validated, or forged not mined as in the PoW.

However, despite the efficiency of the PoS to prevent security breaches and reduce energy consumption, it is still prone to a series of attacks that hinder the growth of PoS network. The transaction fees and stake reward system of the PoS are strongly coupled with the security properties of the PoS protocols (Larimer, 2013). These has given rise to a host of attack vectors targeting the PoS consensus based on their incentive mechanism. For this reason, we have developed a novel dataset to evaluate the security and performance of the PoS based on the block validation features like coin-stake, coin-age, stake distribution rate, stake reward, etc. The dataset is a semi-synthetic dataset which means that it is an artificial dataset that was created using generated and real features that mirror the statistical properties of the original data but has no information regarding real people. We have chosen to use a semi-synthetic data because there is a lack of real-world data for PoS consensus based blockchain. This semi-synthetic dataset was created to train an AI model in other to detect a Long-Range Attack (LRA) in the PoS consensus protocol. Machine Learning (ML) was chosen as an efficient solution because of its ability to train from small datasets, learn patterns from previous data and apply this knowledge effectively to make informed decisions. The features were selected using a feature selection method to eliminate redundant features and features that have no significant impact on the performance of the model. ML is efficient for systems that lack context and background in decision making. The ML model will take into consideration all the inputted data for the PoS block creation and validation process to determine the best accuracy for our solution.

PoS protocols lack the theoretical background and formal definitions to support their security (Bhardwaj, 2021). For instance, attacks like double spending, bribery attacks, grinding attacks, 51% attacks, and long-range attacks are set-off because of attempts to penetrate the blockchain network (Irannezhad, 2020). A comprehensive study of the effects of stake-pool parameters on the network stake, block-generation rate, effects of block rewards and the total network stake can create a more secure PoS consensus mechanism for future blockchain networks. This paper will give an overview on the security of the PoS consensus with a focus on LRA and propose a solution to mitigate it.

Our motivation for this work is because there is limited research on Long-Range Attacks on PoS consensus mechanism and how this is a major threat to the core of the consensus protocol. With the rise in PoS blockchain use-cases, an attack like LRA can have major consequences. Another motivation for our research is the lack of publicly available Proof-of-Stake blockchain datasets for ML experiments. The combination of ML and blockchain can enhance the security blockchains and help blockchain developers make informed decisions.

In other to mitigate the major attacks plaguing the PoS, some studies have proposed hybrid solutions, fines, special node validators and voting systems. While some of these approaches give more knowledge on the incentive mechanism behind the PoS; there are still some shortcomings on how this makes the PoS more secure and improves performance (Kaur et al., 2021). Motivated by this, we investigate

the Long-Range Attack on the PoS consensus protocol. Long-Range Attacks (LRA) can be described as a fork in the blockchain which occurs from the genesis block. The unique characteristics of Weak-Subjectivity and Costless-Simulations of the PoS protocols increases the attack surface for a Long-Range attack to take place, this is discussed in more detail later in the paper.

A Long-Range attack if successfully implemented will allow a malicious node to rewrite the entire history of all the transactions stored on the blockchain thereby challenging the core advantage of blockchains which is immutability (Gazi et al., 2018). This attack only occurs in PoS protocols because of the relationship between the coin-age selection and random-block selection method that determine the output of the block-reward distribution and block-generation rate which is not found in the PoW consensus because of its cryptographic architecture (Qu et al., 2018).

PoS consensus became attractive because of the low energy consumption rate required for block validation. The PoS consensus was first adopted by the cryptocurrency called Peercoin (Puthal et al., 2018). Participating nodes deposit a sum of token (Stake) into the stake-pool, and this earns the nodes voting rights to elect validators to forge/validate blocks (Chalaemwongwan & Kurutach, 2018). The PoS consensus is susceptible to malicious attackers because of the Nothing-at-Stake or Costless Simulation architecture. This means that it costs a malicious node nothing to attack the blockchain. The attacker will expend no real-world resources in creating a malicious attack on the PoS blockchain. Nodes who want to participate in the staking pool must buy tokens and then make a minimum stake (deposit) set by the PoS network protocols. The more stake a node has, the more chance of being selected as the validator for that round or epoch (Jin & Xia, 2022). Forged blocks are validated and added to the chain.

A long-range attack phenomenon is considered successful if a malicious validator was able to create a secret or alternative chain starting from the genesis block and is identical to the real or main chain. Some solutions have been proposed to detect and mitigate long-range attacks such as software updates and checkpoint publishing for all nodes. These solutions have remained conceptual and are difficult to implement without leading to centralization of the blockchain system. Long-range attack is possible because validators can sell their old keys to a malicious actor putting the entire system at risk. This observation has led us to propose this novel approach that can detect malicious validators by studying the block creation and validation process to prevent long-range attacks on the proof-of-stake consensus.

The main novelty of this paper is the creation of a novel dataset for Proof-of-Stake (PoS) blockchains. The next main aspect can be found in the application of machine learning algorithms to detect malicious nodes, actors, and validators on the PoS blockchain. Most of the research and studies on Long-Range Attacks (LRA) have proposed some mitigating solutions but none of this mitigating solution has been implemented and tested on a PoS blockchain dataset. This paper has proactively demonstrated that artificial intelligence models can be used as a mitigating checkpoint for LRAs. Although our solution can be applied to other consensus mechanisms, it is mainly applicable to proof-of-stake blockchains. Our solution monitors malicious validators creating a long-range attack by freely replicating transactions to rival the main-chain. The proposed solution can also give protection against attacks like replay attack. Finally, we simulate our proposed solution on our novel PoS dataset and evaluate the performance.

The important contributions of our approach are summarized as follows:

- We developed a new dataset to detect malicious nodes and node classification on a permissionless blockchain.
- We identify and introduce features for validator nodes based on research from previous attacks on PoS based blockchains.
- We developed a Deep Learning algorithm that can be applied to the above dataset and is both practical and effective

The rest of this paper is as follows: In the next section we discuss background studies on PoS consensus and analysis of long-range attacks. Then in Section 3, we discuss related work on LRA and PoS based attacks. In Section 4 we discuss our dataset context, creation, content, and classification of nodes. In Section 5 and 6 we discuss our proposed method, solution, and experimental results. Finally, in Section 7 we end with conclusion and future research.

2. Background

The PoS consensus has its unique system of validating transactions on the blockchain. In this section we will discuss the PoS consensus in more detail and explore two major characteristics of the PoS that make it susceptible to attack vectors.

2.1 Validating Transaction in the PoS Consensus Mechanism

The blockchain network allows the agreement of nodes to operate via the system of the consensus mechanism. PoS is described as a consensus mechanism that operates through a staking system where validator nodes validate transactions to be included in the blockchain ledger. The PoS consensus is a relatively new consensus mechanism and was first developed in 2011 and has gone through numerous iterations to increase the efficiency (Wu et al., 2019). One of the fundamental building blocks of the PoS consensus is "the ability of nodes to decide between conflicting valid chains" (Deirmentzoglou et al., 2019). This characteristic presents a unique strength but also presents a weakness that can be exploited by malicious attackers and lead to an attack such as the Long-Range Attack (LRA). Since the inception of the PoS, researchers and developers have deliberated on the security in comparison to the PoW. The PoS adopts a stake-pool process as opposed to the mining process in the PoW. In the PoS consensus miners are called "Validators" or "Forgers" and they are not required to solve any complex mathematical puzzle for a block to be published (Hazari & Mahmoud, 2019). These group of validators deposit a stake to be selected as the next validator and receive a reward for successfully broadcasting these transactions on the blockchain network. The more tokens a validator is willing to deposit into the stake-pool, the greater their chances of validating the next block.

PoS allows anyone to become a participating node without having a high computing power setup. This process of selecting or electing a validator with the largest stake is managed through a pseudo-random generator to ensure fairness and more decentralization of the PoS consensus (Sankar et al., 2017). Once a validator has been elected, they will select transactions from the transaction pool to be added to their block, the block is then published or sent to other validators for approval depending on the type of PoS (Zheng et al., 2017). A fork-choice protocol is embedded into the PoS consensus and the chain with the most accumulated stake will be accepted as the main-chain; that is the chain where the validators have the most votes, bets, and tokens. Figure 1 shows how a block is generated for validation on the PoS and the fields included in the block at that stage. Let it be noted that the output of validator nodes is validated block transactions.

```
// isBlockValid makes sure block is valid by checking pointer
// and comparing the hash of the previous block
func isBlockValid(newBlock, oldBlock Block) bool {
        if oldBlock.Pointer+1 != newBlock.Pointer {

                return false
        }

        if oldBlock.Hash != newBlock.PrevHash {

                return false
```

```
}

if calculateBlockHash(newBlock) != newBlock.Hash {

        return false
}

return true
```

Figure 1: Code for Block Generation in PoS (Kaur et al., 2021).

The level of security of this staking process has been up for debate because of some major characteristics that define the PoS consensus but also limit it. The first series of PoS implementations were established on a naive foundation that brought in a wave of attacks on existing PoS applications. The PoS is predicated on the fact that it requires very low-computational power to run the network, but this also makes it simple for attackers to launch an attack. This concept is referred to as Costless Simulation. Another flaw in the PoS design is termed as Weak Subjectivity where nodes that have been offline then come online and cannot decipher which is the main chain. We discuss more about the concepts of weak subjectivity and costless simulation in the next section.

2.2 The Concepts of Weak Subjectivity and Costless Simulation

As we briefly discussed in the introductory section of the paper; one of the major attacks facing the PoS consensus is the long-range attack where a malicious attacker starts creating a different branch or moves to fork the genuine blockchain with an intention of overtaking it (Xiao et al., 2020). This attack vector is intensified because of the concepts of "weak subjectivity" and "costless simulation" of the PoS consensus.

2.2.1 Weak Subjectivity: The concept of weak subjectivity predominantly affects nodes that have offline for a long time and new nodes that have just joined the PoS (Fanti & Kogan, 2021). The problem of weak subjectivity is when a new or offline node is presented with many branches on the blockchain which may be the same length or not but are all in the race to becoming the main-chain; they are unable to distinguish between these branches and may be deceived into accepting a malicious branch as the main-chain (Bouraga, 2021). The problem of weak subjectivity does not affect online nodes because they are aware of the genuine branch from monitoring the blockchain in real-time.

2.2.2 Costless Simulation: Costless simulation problems in the PoS consensus stems from the core advantage of the PoS, which is that it requires no complex mathematical puzzle therefore there is little to no energy consumption in generating blocks on the blockchain. This means that it will cost an attacker no resource to attack the blockchain (Kaur et al., 2021). Nodes can create a different branch to fork the blockchain without spending any resource and this affects the stability and security of the main chain. This type of attack will be impossible on a PoW blockchain due to the high energy and computational power required when generating a new block (Shahaab, 2019).

2.3 The Relationship Between Coin-Age Selection and Randomized Block Selection

Selecting the next validator on the PoS is dependent on the coin-age and block selection process. If a validator is selected simply by the number of coins that they possess this will make the PoS system favor the rich and will lead to centralization. Therefore, the coin-age selection method and randomized block selection method ensure fairness in the PoS consensus.

2.3.1 Coin-Age Selection: this is calculated by multiplying the stake (coin) of a validator by the number of days the coins have been held in the staking pool (Ometov et al., 2020). There is a universal rule in the PoS consensus that coins must be held for a minimum of 30 days before they can compete to be selected as a

validator. Once a validator successfully publishes a block, the coin-age is reset back to zero and they must wait for at least 30 days before they compete again in the staking pool. Validators are paid a stake reward anytime they successfully publish a block. The coin-age selection method removes monopoly on the PoS consensus. Here is an example of a few lines of pseudocode for coin age selection is PoS:

1. Function coin_age (Node 1)
2. Var n = number_of_coins_staked (Node 1)
3. Var multiplication time = number_of_days_coins_staked (Node 1)
4. Coin_age = n * multiplication_time

2.3.2 Randomized Block Selection: This method of random block selection is encoded by a formula that combines the use of selecting a validator with the lowest hash or hit value with the total staked coins in the stake pool to select them as a validator (Zarin et al., 2021). A validator with the lowest hash value not exceeding the target value set by the network is chosen to forge the next block. This unique combination of hit value and staked coins creates an additional level of security and fairness into the PoS consensus mechanism.

2.3.3 Simple Workflow of the Proof-of-Stake Consensus Mechanism

1. All transactions are put in the transaction pool via the proof-of-stake algorithm.
2. Competing validator nodes deposit a minimum stake defined by the network into the staking pool.
3. The PoS algorithm uses the coin-age and randomized block selection method to select a validator.
4. Validators will then verify and publish block while their stake is held by the network. This is until other nodes verify the published block.
5. If successful, the validator gets a stake reward for publishing a new block and their original stake is returned.
6. The coin-age for the validator stake is reset to 0.
7. If unsuccessful, the validator will lose their stake.

2.4 Advantages of Proof-of-Stake (PoS)

As stated in the section 1 (Introduction), the PoS was created as a solution to the problems associated with the PoW consensus. Below are the major advantages of the PoS consensus:

- One of the most citied advantages of the PoS is in the low energy consumption and environmentally friendly nature as opposed to the high energy consumption required in the PoW.
- Participants in the PoS have a stake in the consensus and overall performance and well-being of the network. Making the interest of both delegators and validators align in a unique manner as opposed to the PoW where miners can sell off their coins after collecting their rewards and are not tied to the consensus.
- The PoS elevates the decentralization of the network by eliminating the mining process and hashing power method found in other consensus mechanisms.

2.5 Analysis of Long-Range Attack in Proof-of-Stake (PoS) Blockchains

Long-Range Attack (LRA) is a potential threat to the PoS consensus. LRA in PoS is when a malicious attacker starts creating a false branch with the intention of forking the blockchain (Xiao et al., 2020). The LRA is successful if the malicious attacker's branch becomes longer than the original or main-chain. This attack forks the main-chain with the hope of attaching forged blocks with different transactions (Lunardi et al., 2019, Aste et al., 2017). It is very important to note that the LRA starts from the Genesis Block which is the first block in the chain.

Let us consider an example of a simple Long-Range attack where there are four validators that we call Validator A, Validator G, Validator Z, and Validator R where Validator Z is the adversary trying to perform

a long-range attack. For the sake of simplicity, we consider that every validator owns the same portion of the systems stake which will be 25% each. Once Validator Z initiates the attack, she forks the blockchain from the genesis block and starts minting a secret branch that will contain different transactions and blocks from the original or main-chain as shown in the figure 2 below.

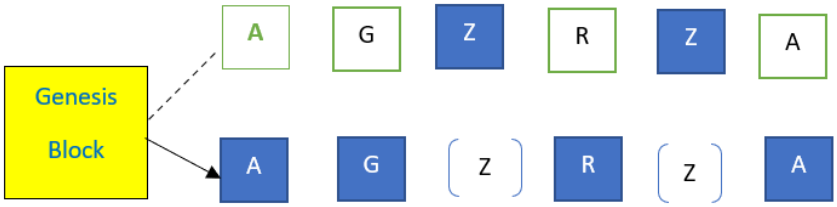

Figure 2: Example of Long-Range Attack

Some solutions to LRA like checkpointing and longest-chain rule have been proposed but have proved insufficient in defending the PoS against an LRA. We discuss some of these studies and their proposed solutions in the next section.

3. Related Work

In this section we give a summary on the related works of attacks on PoS based blockchains and studies on LRA.

The lack of standards in developing blockchains and their consensus make it vulnerable to attack vectors. The popularity of the PoS consensus is growing daily and attacks are becoming more imminent. Stake-pools are a source of income for participating nodes which can be an incentive for attacks to be launched on the PoS based applications. Past studies have explored attack vectors on the PoS consensus and attacks on blockchains. The LRA is documented as one of the most critical attacks that can invade the PoS consensus. There are many studies of attacks on blockchains but only a limited number of studies and research were focused on LRA. Extensive analysis on all blockchain attacks have not been caried out in this paper and the focus of this research is predominantly on the long-range attacks. Some studies have proposed some solutions to defeating LRA, but this has been based on surveys while others proposed hybrid consensus that involves some form of PoW mining and consumes energy.

In (Akbar et al., 2021), they propose a distributed hybrid solution of PoW and PoS, but their solution was limited by scalability and more centralization of the consensus. The PoW layer of the solution prevents block generation time from being consistent. The PoS validates the block without voting rights. In (Nguyen et al., 2019) they investigate the PoS consensus using a performance analysis. Their research employed data from the past studies and survey of PoS protocols. They analyze the stake pools to show the result between block rewards and total network stake. This model will have a high impact on the decentralization of the network which is counter-intuitive.

In Deirmentzoglou et al., (2019), they provide a survey on LRA and propose some mitigating solutions. Their approach covers only components of PoS block creation and PoS block validation process, and their analysis is limited due to a lack of data and implementation process to test their solution.

In (Wang et al., 2020), they propose a PoS protocol for LRA using a secure randomness beacon that will be generated by tamper-proof hardware, but their solution does not prevent dormant participants from launching a costless simulation attack. Their solution is based on bootstrapping via a social consensus, but the data is not publicly available to explore making it difficult to ascertain the accuracy of the solution. They make some challenging assumptions like “At any round, the malicious nodes control less than one-third of the total money in the pool” for the sanity of the PoS network which is not feasible in a real-world PoS implementation.

In (Azouvi et al. 2020), they propose the use of checkpoints, but this relies on some type of central authority. In their follow up work (Azouvi & Vulkolic 2022), use a new taproot upgrade to Bitcoins

blockchain that allows efficient threshold signatures and checkpointing but this solution lacks scalability. The checkpointing mechanism still makes use of some form of PoW which will limit the scalability of the solution due to limited real-world resources.

Babylon proposes a solution to use Bitcoins hashing power to enhance the security of PoS based blockchains (Tas et al., 2022). The drawback of this solution is that it requires miners (Bitcoin) to have the Babylon software. BMS proposes a smart contract solution to mitigate LRA but has some fundamental issues like the security for the underlying PoS blockchain (Steinhoff et. al., 2021).

In (Kuznetsov & Tolkih, 2020), they propose a long-range attacks solution by using secure digital signatures, but this solution is challenged by an assumption where most nodes disobey the assumption and hold on to their old private keys. Our solution is a refinement of this studies as it evaluates the block validation and creation process by nodes (participants) on the PoS based blockchain; it also protects the stake of honest validators and creates fair distribution of stake rewards after blocks are validated. The limitation of this study is in the availability of the type of data collected and used in their studies. This makes reproducibility and enhancement of their solution difficult.

In (Agarwal et. al. 2021), they study the behavior of accounts on a permissionless blockchain using temporal graph properties. They generated features from previous cryptocurrency attacks and a systematic review of transaction data from Ethereum. However, their solution is limited to studying the behavior of malicious accounts and this was focused on transactional data and not block generation and validation data. Their dataset introduces features that change overtime which create inconsistencies in ML model performance. Features used fail to capture validator and block generation behavior.

The existing studies either use large or small data sets with a high-class imbalance that makes analysis inconsistent. The ML algorithms are applied to dataset with features that fail to express the desired solution or features that fail to capture the proposed classes. Our study addresses these limitations, and we focus on collecting new features by studying and comparing performance metrics, effects, and relationships of PoS pool parameters, timestamping, longest chain rule and identifying the right ML classifiers to address the issue of long-range attacks on the PoS consensus. The features introduced to the dataset capture the classes of malicious and non-malicious nodes that we want to predict and help to detect the launching of a long-range attack. Applying these features to our ML model gave the best accuracy for detecting long-range. This makes our study novel and unique compared to other related studies because proposed method and solution avoids the introduction of any form of centralization into the PoS consensus. Our solution is scalable compared to the other solutions above; it shows elements of reproducibility, can be enhanced, and effectively criticized for future research endeavors. The availability of the dataset allows experimentation into patterns of PoS coin-staking and block validation process. This will enhance the body of knowledge in the blockchain ecosystem.

There is not a publicly available PoS dataset and most permissionless blockchain datasets that are available are financial transaction that do not capture block creation and validation data. This makes our studies and dataset novel in the knowledge contribution, proof-of-stake data collection and a low-cost ML prediction method that maximizes performance.

4. Proposed Dataset

PoS consensus blockchains involve participants (nodes) locking in their stake (tokens) to become validators for them to validate transactions. Long-range attacks threaten the purity of this validation process by taking over the main-chain completely or partially for the purpose of rewriting the transaction history that are stored on the chain.

4.1 Dataset Creation

In this Dataset, we initially developed a semi-synthetic dataset for 303 entries based on the node (validator) and block (transaction) outputs. This dataset was then augmented using python libraries to simulate additional entries to create an augmented dataset with 10,000 entries. Some of our data was extracted from the Etherscan API from Ethereum accounts/nodes with tags such as Phishing and Hacks. Note that to get

the ground truth, we manually browsed through the tags on Etherscan and marked such nodes as malicious that had the tags of 'Phishing' and 'Hack' in their tags. Etherscan in May 2020 labelled 4708 nodes as malicious, and these have been tagged (Newsbtc, 2018). These tags represent nodes that have been engaged in any form of hacking and phishing on the Ethereum network. Some of these nodes can also be prime suspects for launching an LRA in the future because of their roles in previous nefarious behaviors on the network (Wang et. al., 2020). The output of malicious nodes are malicious transactions; we explain more on this in (section 4.2). Our dataset is split between malicious nodes and non-malicious nodes. There is some disparity in the ratio of malicious nodes to non-malicious nodes. The features of this dataset that were collected include: TxHash, block height, UnixTimestamp, TxFee (ETH), TxFee (Binary), Status (Tag). The other features identified from past studies include, block explorer APIs and synthesis of past research, and they include: block-generation rate, coin stake, stake distribution rate, stake reward, coin age, coin days, block density, block score, coin-day weight, transaction size and label

4.1.1 Feature Selection

The core reason for including a feature selection step was to minimize the number of features and ensure a rigorous feature selection process is included in our model without sacrificing the performance of the model. For this paper we adopted the Backward Feature selection method where the major features for the PoS block creation and validation process are run to find the best features that will suit our model. The insignificant features are removed after thorough combination of features. For this dataset features such as "coinstake Kernerl", "Transaction Amount ($Tx)", "TransactionTimeStamp", "Block Proposer" "Transaction Hash", "Senders Address" and "Receivers Address" were removed. It was computationally impossible, time-consuming, and expensive to look at all the combination of features that embody block creation, block generation, block validation and the PoS network protocols. For the size of our dataset, it was inexpensive to run a backward feature selection to optimize our model. The features that were selected to best optimize the model are discussed in the following section.

4.1.2 Data Glossary (Column-Wise)

- **TxHash:** This is the transaction hash associated with the validated block.
- **Block-generation rate:** The rate at which blocks are created every epoch, starting from the Genesis Block. This feature was used in the studies of (Akbar et. al, 2021).
- **Coin stake:** The number of coins or token a node must stake to be selected as a validator. The minimum stake for our scenario is 30 tokens. Any node that stakes less than 30 tokens will not be selected as a validator and those that stake more than 30 have a higher probability of being selected as validators. This feature was used in the studies of (Akbar et. al, 2021).
- **Stake distribution rate:** This the process of actively participating in the transaction validation on the PoS network. This feature was used in the studies of (Akbar et. al, 2021).
- **Stake reward:** This is the coin reward a validator gets for validating a transaction and adding it to the block to create a valid block. The reward will be the same for every valid block created. This has been represented as: Stake Reward = 1 and No stake Reward = 0
- **Transaction fees:** Transaction fee vary depending on the size and speed of the transaction. For this data set we will populate this as: **Transaction Fee = 1** and **No Transaction Fee = 0**
- **Coin days:** This will be represented in periods based on the PoS protocols and rules. This will be represented as**: Null = 0, 0 – 30 days= 1, 30 – 60 days = 2, 60 – 90 days = 3**. This feature was used in the studies of (Li et. al., 2020).
- **Block height:** This is calculated by an addition of Epoch + Block generation rate +stake distribution + stake reward.
- **Coin-day weight:** The parameters for this are:

Coin Age	60 days	90 days	120 days
Coin day weight	30	60	90

- **Transaction size:** Number of transactions in a block.
- **Coin Age:** This is the Total coins staked X No. of days (period) coins is held. This feature was used in the studies of (Li et. al., 2020).
- **Label:** The labels are Malicious Nodes and Non-Malicious Nodes

The figure 3 shows a sample of an Ethereum transaction with some very important features like block height and block (stake reward).

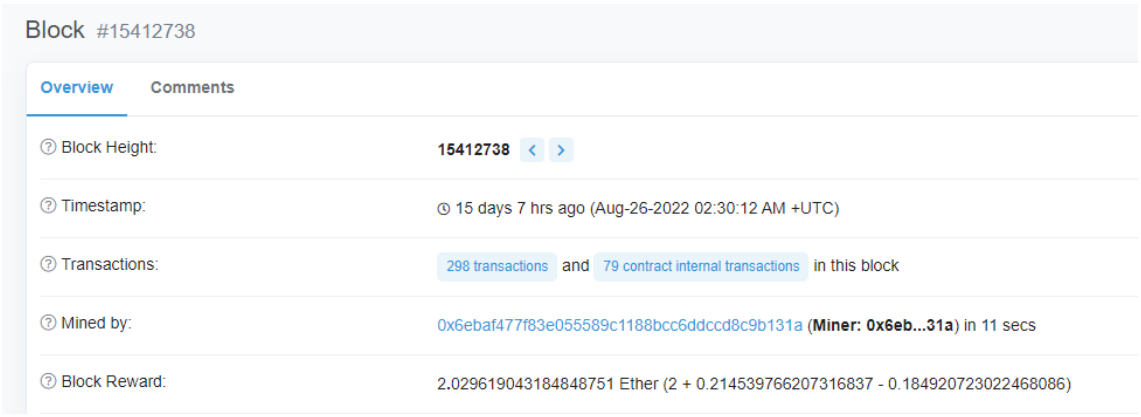

Figure 3: Sample of Ethereum Transaction Block

The figure 4 shows a sample of proposed blocks for validation on a PoS network with features like epoch, slot, status, coinage, and block proposer, which are time variant. This is how we can get a view of the content of a block from a stake pool where we can view the chained blocks of transaction with unique block heights even within the same block and a unique stake pool ID for each round.

Figure 4: Sample of PoS Transactions Blocks for Validation

4.2 Node Classification

We classify accounts set up on a permissionless blockchain as nodes. Therefore, some nodes (accounts) can be honest or malicious. Transactions among these nodes are done by interactions and communication between nodes. These transactions are then inputted on the blockchain ledger to form blocks on the blockchain. Motivated by this we define honest nodes, honest transactions, malicious nodes, and malicious transactions as follows:

- **Honest Nodes:** These are nodes that follow and adhere to the rules and protocols set by the PoS staking-pool and have no intentions on cheating or conspiring together to create a fork or secret-chain on the blockchain.
- **Honest Transactions:** These are created from the interactions among honest nodes to produce records that will be validated and inputted on the blockchain ledger. The output of these interactions is termed as honest transactions.
- **Malicious Nodes:** These are nodes or a node that is trying or has successfully partitioned the whole network by separating themselves from the other nodes. These nodes conspire with other nodes to create a fork in the blockchain and irregularly validate transactions.
- **Malicious Transactions:** This is when a malicious node proposes a candidate block to become a validator and they have conspired and succeeded in stealing or buying the private key of offline nodes to have enough staking power to continuously verify block transactions at every round (Wang et. al., 2020). The transactions produced from the interactions among malicious nodes are termed as malicious transactions.

It should also be noted that malicious nodes produce blocks at a 2X rate and tend to interact with other nodes at a faster rate to produce transactions (Wang et. al., 2020). Honest nodes operate at network stability and interact at a much lower rate. Since nodes can go from being malicious to honest and honest to malicious there is a continuous need to analyze block generation timescales for prediction. Malicious nodes as seen on the (Etherscan.io) can spew out both honest and malicious transactions.

We therefore collected and organized our features based on previous studies on attacks on blockchains and proposed some new features for our semi-synthetic PoS consensus dataset such as block generation rate, coin age, coin-day weight, stake distribution rate and coin days.

4.3 Structure of the Dataset

The figure 5 shows the first four entries of our dataset and the corresponding columns and features of the dataset.

	BlockHeight	UnixTimestamp	TxnFee(ETH)	TxnFee (Binary)	Status (Tags)	Block Generation Rate	Stake Reward	Coin Stake	Stake Distribution Rate	Txnsize	Coin Days	Coin Age	Block Density (%)	Block Score	Coin Day Weight	Node Label
0	15450240	1659043486	0.515018	1	1	1	1	92	2699	75	1	112	2939	6658	51	1
1	6379281	1660912130	0.000000	1	1	0	1	53	2561	47	1	115	1740	4447	54	1
2	5468684	1524145611	0.000000	1	0	0	0	62	1551	86	2	101	1913	2224	0	0
3	15450240	1645535000	0.511077	1	1	1	1	88	1448	72	1	75	2107	4790	55	1
4	5740662	1524995972	0.000000	1	0	0	1	39	890	79	1	98	1495	3534	58	0

Figure 5: Snapshot of the First Four (4) Entries of the Dataset

https://www.kaggle.com/datasets/a9910rut/proofofstake-blockchain-dataset **(Dataset Link)**

5. Proposed Method

In this section we propose an all-round approach for detecting malicious nodes to mitigate the threat of launching a Long-Range Attack (LRA) on the PoS permissionless blockchain.

Machine Learning (ML) is the ability of a machine to keep improving the performance and accuracy without any human interference and influence. ML has become very popular because of large datasets being collected by both small, medium, and large organizations in other to predict patterns and make informed decisions. ML approach splits the datasets into training and testing datasets and with the use of novel algorithms the machine can cluster or classify data for accurate decision-making. ML allows to write algorithms that learn by itself by giving it many examples of the problem. Algorithms such as k-means, decision trees, and logistic regression can improve their performance with an increasing amount of data. The choice of algorithm used in ML is determined by the task and solution to be proposed. For example, our paper proposes a Deep Learning classifier and examines the use of ML classifiers such as the K-nearest neighbors (KNN), Decision Tree (DT) and Multi-Layer Perceptron (MLP), as a tool to discover patterns in PoS block creation and validation process that are not inherently apparent.

One of the major advantages of ML is in its ability to produce good results with small datasets simply by identifying the input, extracting the appropriate features, classification and providing the output. This is our rationale for selecting this as a suitable model for node prediction and classification. Our Business Problem is to use ML as a tool to mitigate long-range attacks in PoS based blockchains because of its ability to consider the selected features of the PoS block creation and validation process.

We assume a proof-of-stake implementation where the consensus protocol has just finished a round and all the validators (Nodes) are active and offline nodes are inactive (Agarwal et. al. 2021). Malicious nodes are trying to fork the blockchain by creating a separate partitioned chain to separate themselves from the other Validators (Nodes). From the Non-Malicious nodes viewpoint these nodes are offline (Wang et al., 2020). We then assume that our system knows the true activity status of each node and that all malicious nodes are active and are trying to fork the blockchain to perform an LRA (Wang et al., 2020). A significant number of the malicious nodes tagged with “Phishing” and “Hack” have been highlighted to be involved in more than one malicious behavior on the Ethereum network (Newsbtc, 2018). A long-range attack is performed by malicious nodes, and we propose the analogy that such nodes that have been involved in previous malicious behaviors have a higher probability of forking the PoS protocol to perform a long-range attack.

Our solution is a Deep Learning Convolutional Neural Network (CNN) as shown in figure 6 below. The proposed method has been developed using the Python programming language and the Keras library with the following settings:

- Sequential architecture
- Bias: True
- Optimizer: Nadam
- Learning rate: 0.03
- Batch size: 32
- Epochs: 20

Subsequently we used the KNN, DT and MLP algorithms to determine which classifier best suits our dataset and will make a good prediction between malicious and non-malicious nodes.
The characteristics below determine the probability of successfully initiating a fork on the PoS blockchain by malicious nodes. They include but are not limited to:

- We propose that the PoS based blockchain is secure and working efficiently.
- The total amount of tokens for both active and inactive nodes at each round is known.
- The amount operated by active or participating nodes and the amount operated by inactive nodes is known.
- Active participants are tagged as honest if they keep the secrecy of their private key and not sell them to the malicious nodes.

- The block-height of the PoS based chain will be a differentiator between malicious nodes and non-malicious nodes. The height on the honest chain must differ from the secret chain (Azouvi & Vukolić, 2022).

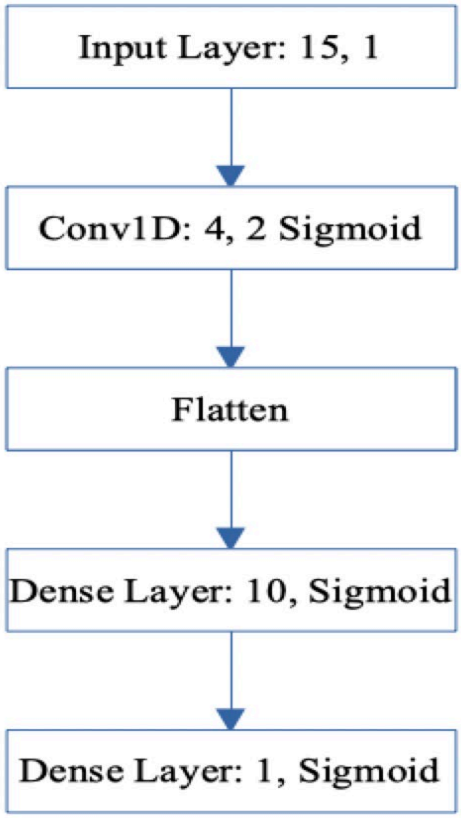

Figure 6. Proposed method

6. Experimental Evaluation

In this section we present the evaluation metrics for this study, result comparison and discussion of our experiment. The evaluation metrics is a very crucial part of measuring the quality and performance of our solution. Our evaluation metrics was done using the Python development interface and corresponding python libraries. Our experimental set up to run the dataset on the python development interface using two standalone systems with 64-bit (Windows 10) operating systems with Intel Pentium CPU 6495u @ 2.40 GHz, with 8 GB (RAM) and we compared the results based on the parameters in the equations below. A cross validation of 5 segments using the Scikit learn library was used in running the experiment.

We initially created a dataset for 303 entries as discussed in section 4 of this paper. This dataset was then optimized to create 10,000 entries from the CSV file of the original dataset (303 entries) using the *Numpy, Python SDK* and *Scikit-learn* libraries. *Numpy* offers a random module with multiple ways of generating random entries from a fixed set of parameters. Our synthetic model operated by learning about our original little dataset to mimic the inputted data closely and maintain characteristics and correlations of the original dataset. This was achieved by loading the .csv file with 303 entries into the Pandas Data Frame to train the model and recommended at least 12,500 entries. After computation this gave an output of 10,050 entries where we selected 10,000 entries for the experiment.

Moreover, since other similar data collections and machine learning or deep learning research aren't available, we tested various classifiers on the novel dataset that we developed for specifically closing this research gap by creating a data collection which will allow further research taking place in this area.

We ran two separate experiments for our dataset of (original) 303 entries and that of (augmented)10,000 entries with four (4) executions for accuracy, precision, recall and F1 measure which are the evaluation metrics adopted. The evaluation metrics include the Accuracy, Precision, Recall and F1 as shown in the equation 1,2,3 and 4 below. The *Accuracy* will assign a representation for the performance of our model; *Precision* is the correct prediction of classes in our dataset, *Recall* the average between the correctly predicted class and total of the whole class and *F1-measure* is termed as the mean or weighted average between recall and the precision. For all experiments we did a 5-Fold cross validation to train the model.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1}$$

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

The components of the evaluation metrics are TP for True Positive, TN for True Negative, FP for False Positive, FN for False Negative. K Nearest Neighbor (KNN), Decision Tree and Random Forest have been compared using the following evaluation metrics: accuracy, recall, precision and F1. The results for our dataset with 300 entries after four executions are shown in tables 1, 2, 3, and 4 below:

Algorithm	**1st Execution**	**2nd Execution**	**3rd Execution**	**4th Execution**
KNN	98.0 %	99.0 %	98.0%	100%
Decision Tree	98.0%	99.0%	100%	100%
MLP	67.8%	65.2%	68.1%	65.5%
Proposed method	100%	100%	100%	100%

Table 1: Accuracy Results for 303 Data Entries

Algorithm	**1st Execution**	**2nd Execution**	**3rd Execution**	**4th Execution**
KNN	98.0 %	100 %	98.0%	100%
Decision Tree	98.0%	100%	100%	100%
MLP	35.4%	34.5%	35.4%	34%
Proposed method	100%	100%	100%	100%

Table 2: Precision Results for 303 Data Entries

Algorithm	1st Execution	2nd Execution	3rd Execution	4th Execution
KNN	99.0 %	100 %	97.0%	100%
Decision Tree	98.0%	100%	100%	100%
MLP	50%	50%	50%	50%
Proposed method	100%	100%	100%	100%

Table 3: Recall Results for 303 Data Entries

Algorithm	1st Execution	2nd Execution	3rd Execution	4th Execution
KNN	98.0 %	100 %	98.0%	100%
Decision Tree	98.0%	100%	100%	100%
MLP	41%	41%	41%	40%
Proposed method	100%	100%	100%	100%

Table 4: F-1 Results for 303 Data Entries

Below we display the results for 10,000 data entries and they are as shown in tables 5, 6, 7, and 8:

Algorithm	1st Execution	2nd Execution	3rd Execution	4th Execution
KNN	75.0 %	77.0 %	76.0%	76.0%
Decision Tree	80.0%	80.0%	80.0%	79.0%
MLP	57.1%	57.1%	57.6%	57.7%
Proposed method	87.11%	86.65%	87.54%	86.91%

Table 5: Accuracy Results for 10,000 Data Entries

Algorithm	1st Execution	2nd Execution	3rd Execution	4th Execution
KNN	76.0 %	77.0 %	77.0%	78.0%
Decision Tree	80.0%	80.0%	79.0%	79.0%
MLP	29%	28.9%	28.7%	28.7%
Proposed method	85.20%	87.68%	87.56%	84.86%

Table 6: Precision Results for 10,000 Data Entries

Algorithm	1st Execution	2nd Execution	3rd Execution	4th Execution
KNN	75.0 %	77.0 %	76.0%	74.0%
Decision Tree	80.0%	80.0%	79.0%	79.0%
MLP	50%	50%	50%	50%
Proposed method	84.45%	79.62%	82.31%	84.48%

Table 7: Recall Results for 10,000 Data Entries

Algorithm	1st Execution	2nd Execution	3rd Execution	4th Execution
KNN	73.0 %	76.0 %	75.0%	74.0%
Decision Tree	80.0%	80.0%	79.0%	79.0%

MLP	37%	37%	36%	36%
Proposed method	85%	83%	85%	85%

Table 8: F-1 Results for 10,000 Data Entries

As shown in tables 5,6,7 and 8 above we have performed the experiment with KNN, Decision Tree and MLP classifiers as baselines. The performance of the proposed method is much higher than KNN, DT and MLP. There is significant improvement in the proposed method in all sets of execution compared to KNN and DT. In general, the proposed method outperforms the baselines in both the original and augmented datasets.

There were several limitations during the data collection process for blockchain block features and validator features due to a lack of publicly available dataset for PoS based blockchains. Our dataset was developed and complied in a semi-synthetic approach. Creating additional features was accomplished by synthesizing data from previous research on PoS blockchain consensus and manually browsing through several block explorer APIs on the web. After rigorous refinement and multiple iterations, we eventually produced a dataset that was applicable to investigating our solution. Our original dataset with 303 entries was then augmented to 10,000 entries using the python libraries to create a more diverse dataset with a direct correlation to the original dataset. This gave the opportunity to compare results from both the original and augmented dataset after training our model.

The classifiers used were a Deep Learning CNN architecture, the KNN, DT and MLP methods. These classifiers were selected due to the size and type of dataset available. KNN and DT are easy to compute and refine hence the reason why we chose them, while they give very good results. The most accurate classifier was the proposed method. Node classification based on malicious behaviors is a way of studying the pattern of malicious nodes when producing blocks with the intention of outpacing the main chain to perform an LRA. This solution is envisioned to run on the backend every 90 days after the PoS stake-pool is recalibrated back to zero tokens and a new round of staking is about to begin on the consensus protocol. The application admin can run this through newly produced blocks to trace anomalous block generation patterns which can indicate malicious nodes validating blocks at an irregular pace. The speed of running our model to detect malicious and non-malicious nodes was moderate and can be enhanced by running data directly from the cloud.

7. Conclusions and future work

Blockchain is an innovative technology that has disrupted many industries over the past decade. There is a growing adoption of proof-of-stake based blockchains into healthcare, finance, voting, supply chain, education, and mining. Proof-of-stake consensus mechanism is based on validator nodes and is scaled up faster than proof-of-work consensus. This creates an attack surface of malicious validators to enact a long-range attack. Many existing applications are currently making the shift from the Proof-of-Work (PoW) consensus to the Proof-of-Stake (PoS) consensus because of their low energy consumption and scalability without expending real-world resources which will help the environment. The long-range attack is one of the largest threats facing the PoS consensus mechanisms as it threatens the core fundamental and building block of the blockchain which is immutability. The malicious nodes in a long-range attack can completely rewrite the history of the blockchain and the transactions stored by that PoS blockchain. In this paper we present a novel dataset with unique features of the block and validators. Our solution and proposed method can be a good countermeasure to detecting long-range attacks on PoS based blockchains and can be extended to further analyze anomalous behaviors of validator nodes on the PoS consensus. Our proposed solution provides an approach to monitor the block creation and validation process of validators. We show that machine learning models can predict the actions of validators trying to fork the blockchain to create a fake-chain. We first conducted a study into the PoS stake pool and how blocks are validated, then we developed our dataset and a Deep Learning classifier. We report an accuracy of 87% which shows that the

proposed method performed best out of all the other models with a high accuracy. Our solution and proposed method and dataset can be a good countermeasure to detecting long-range attacks on PoS based blockchains and can be extended to further analyze anomalous behaviors of validator nodes on the PoS consensus.

This research successfully contributed to the knowledge of curating proof-of-stake datasets, the application of machine learning models to detect malicious nodes to mitigate long-range attacks and a study into the coin-age, block creation and block validation process. This solution is however only applicable to permissionless blockchains and the lack of available datasets on PoS protocols is a limitation. A future research direction can be to use reinforced learning to study the behavior of malicious nodes and different types of malicious activities conducted by validator nodes on the PoS blockchain.

References:

Agarwal, R., Barve, S., & Shukla, S. K. (2021). Detecting malicious accounts in permissionless blockchains using temporal graph properties. Applied Network Science, 6(1). https://doi.org/10.1007/s41109-020-00338-3

Akbar, N. A., Muneer, A., Elhakim, N., & Fati, S. M. (2021). Distributed hybrid double-spending attack prevention mechanism for proof-of-work and proof-of-stake blockchain consensuses. Future Internet, 13(11). https://doi.org/10.3390/fi13110285

Aste, T., Tasca, P., & Di Matteo, T. (2017). Blockchain Technologies: The Foreseeable Impact on Society and Industry. Computer, 50(9), 18–28. https://doi.org/10.1109/MC.2017.3571064

Azouvi S., Danezis G., & Nikolaenko V. (2020). Winkle: Foiling Long-Range Attacks in Proof-of-Stake Systems. In Proceedings of the 2nd ACM Confrence on Advances in Financial Technologies, Pages 189-201.

Azouvi, S., & Vukolić, M. (2022). Pikachu: Securing PoS Blockchains from Long-Range Attacks by Checkpointing into Bitcoin PoW using Taproot. In Proceedings of ACM Conference (Conference'17) (Vol. 1, Issue 1). Association for Computing Machinery. http://arxiv.org/abs/2208.05408

Bhardwaj, S. (2021). Blockchain's consensus algorithm: A review. ACADEMICIA: An International Multidisciplinary Research Journal, 11(11), 1022–1029. https://doi.org/10.5958/2249-7137.2021.02562.3

Bouraga, S. (2021). A taxonomy of blockchain consensus protocols: A survey and classification framework. Expert Systems with Applications, 168(June 2020), 114384. https://doi.org/10.1016/j.eswa.2020.114384

Chalaemwongwan, N., & Kurutach, W. (2018). Notice of Removal: State of the art and challenges facing consensus protocols on blockchain. International Conference on Information Networking, 2018-January (April), 957–962. https://doi.org/10.1109/ICOIN.2018.8343266

Deirmentzoglou, E., Papakyriakopoulos, G., & Patsakis, C. (2019). A survey on long-range attacks for proof of stake protocols. IEEE Access, 7, 28712–28725. https://doi.org/10.1109/ACCESS.2019.2901858

Etherscan (2022). Ethereum Developers API. https://etherscan.io/blocks

Fanti, G., & Kogan, L. (2021). Economics of Proof-of-Stake Payment Systems. November 2018, 1–24.

Gazi, P., Kiayias, A., & Russell, A. (2018). Stake-bleeding attacks on proof-of-stake blockchains. Proceedings - 2018 Crypto Valley Conference on Blockchain Technology, CVCBT 2018, 85–92. https://doi.org/10.1109/CVCBT.2018.0001510

Gemeliarana, I. G. A. K., & Sari, R. F. (2018). Evaluation of proof of work (POW) blockchains security network on selfish mining. 2018 International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2018, February, 126–130. https://doi.org/10.1109/ISRITI.2018.8864381

Hazari, S. S., & Mahmoud, Q. H. (2019). Comparative evaluation of consensus mechanisms in cryptocurrencies. Internet Technology Letters, 2(3), e100. https://doi.org/10.1002/itl2.100

Houy, N. (2014). 'It will cost you nothing to "kill" a proof-of-stake crypto-currency. Economics Bulletin, 34(2), 1038–1044. https://doi.org/10.2139/ssrn.2393940

Irannezhad, E. (2020). The Architectural Design Requirements of a Blockchain-Based Port Community System. Logistics, 4(4), 30. https://doi.org/10.3390/logistics4040030

Ismail, L., & Materwala, H. (2019). A review of blockchain architecture and consensus protocols: Use cases, challenges, and solutions. Symmetry, 11(10). https://doi.org/10.3390/sym11101198

Jain, A., & Jat, D. S. (2022). A Review on Consensus Protocol of Blockchain Technology. Lecture Notes in Networks and Systems, 334, 813–829. https://doi.org/10.1007/978-981-16-6369-7_72

Jin, S. Y., & Xia, Y. (2022). CEV Framework: A Central Bank Digital Currency Evaluation and Verification Framework With a Focus on Consensus Algorithms and Operating Architectures. IEEE Access, 10, 63698–63714. https://doi.org/10.1109/access.2022.3183092

Kaur, S., Chaturvedi, S., Sharma, A., & Kar, J. (2021). A Research Survey on Applications of Consensus Protocols in Blockchain. Security and Communication Networks, 2021. https://doi.org/10.1155/2021/6693731

Kaur, M., Khan, M. Z., Gupta, S., Noorwali, A., Chakraborty, C., & Pani, S. K. (2021). MBCP: Performance Analysis of Large-Scale Mainstream Blockchain Consensus Protocols. IEEE Access, 9, 80931–80944. https://doi.org/10.1109/ACCESS.2021.3085187

Kuznetsov, P., & Tonkikh, A. (2020). Asynchronous reconfiguration with byzantine failures. Leibniz International Proceedings in Informatics, LIPIcs, 179(27), 1–17. https://doi.org/10.4230/LIPIcs.DISC.2020.27

Larimer, D. (2013). Transactions as proof-of-stake. Cryptochainuni.Com, 1–8. https://cryptochainuni.com/wp-content/uploads/Invictus-Innovations-Transactions-As-Proof-Of-Stake.pdf

Lunardi, R. C., Michelin, R. A., Neu, C. V., Nunes, H. C., Zorzo, A. F., & Kanhere, S. S. (2019). Impact of consensus on appendable-block blockchain for IoT. ACM International Conference Proceeding Series, 228–237. https://doi.org/10.1145/3360774.3360798

Nwesbtc (2018). Over 3,000 Ethereum Smart Contracts Contain Major Security Flaws. https://www.newsbtc.com/news/over-3000-ethereum-smart-contracts-contain-major-security-flaws/

Nguyen, C. T., Hoang, D. T., Nguyen, D. N., Niyato, D., Nguyen, H. T., & Dutkiewicz, E. (2019). Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities. IEEE Access, 7, 85727–85745. https://doi.org/10.1109/ACCESS.2019.2925010

Ometov, A., Bardinova, Y., Bardinova, Y., Afanasyeva, A., Masek, P., Zhidanov, K., Vanurin, S., Sayfullin, M., Shubina, V., Komarov, M., & Bezzateev, S. (2020). An Overview on Blockchain for Smartphones: State-of-the-Art, Consensus, Implementation, Challenges and Future Trends. IEEE Access, 8, 103994–104015. https://doi.org/10.1109/ACCESS.2020.2998951

Puthal, D., Malik, N., Mohanty, S. P., Kougianos, E., & Yang, C. (2018). a Decentralized. IEEE Consumer Electronics Magazine, 7(March), 18–21.

Qu, Q., Xu, R., Chen, Y., Blasch, E., & Aved, A. (2021). Enable Fair Proof-of-Work (PoW) Consensus for Blockchains in IoT by Miner Twins (MinT). Future Internet, 13(11), 1–17. https://doi.org/10.3390/FI13110291

Sankar, L. S., Sindhu, M., & Sethumadhavan, M. (2017). Survey of consensus protocols on blockchain applications. 2017 4th International Conference on Advanced Computing and Communication Systems, ICACCS 2017. https://doi.org/10.1109/ICACCS.2017.8014672

Shahaab, A., Lidgey, B., Hewage, C., & Khan, I. (2019). Applicability and Appropriateness of Distributed Ledgers Consensus Protocols in Public and Private Sectors: A Systematic Review. IEEE Access, 7, 43622–43636. https://doi.org/10.1109/ACCESS.2019.2904181

Steinhoff, S., Stathakopoulou, C., Pavlovic, M., & Vukolić, M. (2021). BMS: Secure Decentralized Reconfiguration for Blockchain and BFT Systems. http://arxiv.org/abs/2109.03913

Tas, E. N., Tse, D., Yu, F., & Kannan, S. (2022). Babylon: Reusing Bitcoin Mining to Enhance Proof-of-Stake Security. In Proceedings of ACM Conference (Conference'17) (Vol. 1, Issue 1). Association for Computing Machinery. http://arxiv.org/abs/2201.07946

Wang, Y., Sun, J., Wang, X., Wei, Y., Wu, H., Yu, Z., & Chu, G. (2020). Sperax: An approach to defeat long range attacks in blockchains. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2020, 574–579. https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9163036

Wu, H., Cao, J., Yang, Y., Tung, C. L., Jiang, S., Tang, B., Liu, Y., Wang, X., & Deng, Y. (2019). Data management in supply chain using blockchain: challenges and a case study. Proceedings - International Conference on Computer Communications and Networks, ICCCN, 2019-July (January 2020), 1–8. https://doi.org/10.1109/ICCCN.2019.8846964

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). Modeling the Impact of Network Connectivity on Consensus Security of Proof-of-Work Blockchain. Proceedings - IEEE INFOCOM, 2020-July, 1648–1657. https://doi.org/10.1109/INFOCOM41043.2020.9155451

Xiao, Y., Zhang, N., Lou, W., & Hou, Y. T. (2020). A Survey of Distributed Consensus Protocols for Blockchain Networks. IEEE Communications Surveys and Tutorials, 22(2), 1432–1465. https://doi.org/10.1109/COMST.2020.2969706

Zarrin, J., Wen Phang, H., Babu Saheer, L., & Zarrin, B. (2021). Blockchain for decentralization of internet: prospects, trends, and challenges. Cluster Computing, 24(4), 2841–2866. https://doi.org/10.1007/s10586-021-03301-8

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017, 557–564. https://doi.org/10.1109/BigDataCongress.2017.85