Utilizing Deep Reinforcement Learning and Q-Learning algorithms for Improved Ethereum Cybersecurity

Professor Gabriel Kabanda
Adjunct Professor of Machine Learning
Woxsen School of Business, Woxsen University, Hyderabad, India
Email: gabrielkabanda@gmail.com / Gabriel.Kabanda@woxsen.edu.in
Dr. Colletor Tendeukai Chipfumbu
Department of Information and Marketing Sciences
Midlands State University Faculty of Business Sciences
Email: chipfumbuc@staff.msu.ac.zw
Tinashe Chingoriwo
DPhil (Information Technology) Candidate
Faculty of Technology, Zimbabwe Open University
Email:chingoriwot89@gmail.com

---ABSTRACT---

The purpose of the research is to explore and develop Deep Reinforcement Learning and Q-Learning algorithms in order to improve Ethereum cybersecurity in contract vulnerabilities, the smart contract market and research leadership in the area. Deep Reinforcement Learning (Deep RL) is gaining popularity among AI researchers due to its ability to handle complex, dynamic, and particularly high-dimensional cyber protection problems. The benchmark of RL is goal-oriented behavior that increases rewards and decreases penalties or losses, and enhances real-time interaction between an agent and its surroundings. The research paper examines the three major cryptocurrencies (Bitcoin, Litecoin and Ethereum) and the role played by cyber-attacks.The Design Science Research Paradigm as applied in Information Systems research was used in this research, as it is hinged on the idea that information and understanding of a design problem and its solution are attained in the crafting of an artefact. The proposed constructs were in the form of Deep Reinforcement Learning and Q-Learning algorithms designed to improve Ethereum cybersecurity. Smart contracts on the Ethereum blockchain can automatically enforce contracts made between two unknown parties. Blockchain (BC) and artificial intelligence (AI) are used together to strengthen one another's skills and complement one another. Consensus algorithms (CAs) of BC and deep reinforcement learning (DRL) in ETS were thoroughly reviewed. In order to integrate many DCRs and provide grid services, this article suggests an effective incentive-based autonomous DCR control and management framework. This framework simultaneously adjusts the grid's active power with accuracy, optimizes DCR allocations, and increases profits for all prosumers and system operators. The best incentives in a continuous action space to persuade prosumers to reduce their energy consumption were found using a model-free deep deterministic policy gradient-based strategy. Extensive experimental experiments were carried out utilizing real-world data to show the framework's efficacy.

Keywords - **Reinforcement Learning; DRL; Double Q-Learning; Blockchain; Ethereum blockchain; Cryptocurrencies; ECC; DNS.**

Date of Submission: March 28, 2023 Date of Acceptance: April 27,2023

I. INTRODUCTION

1.1 Background

A new era of artificial intelligence is currently being ushered in by recent advances in Machine Learning (ML) and neuroscience, as well as an increase in data volume and a new generation of computers (AI). At the moment, deep reinforcement learning, which encompasses a number of potent techniques, is gaining popularity among AI researchers (RL). Deep RL techniques have been effectively used in a variety of applications, from natural language processing to picture understanding. For instance, deep RL has been frequently suggested to combat cyberattacks against systems linked to the Internet. Cyber threats are complex and dynamic, therefore defenses must be quick to react, be flexible, and be powerfully efficient. Deep RL has demonstrated a strong ability to handle complicated, dynamic, and particularly high-dimensional cyber protection problems. Deep RL systems also outperformed humans at several Atari video games and the game Go. The fact that biological and artificial organisms both need to accomplish goals in order to survive and be useful is a factor in the success of deep RL. The benchmark of RL is this behavior that is goal-oriented. Such behavior is founded on developing behaviors that increase rewards and decrease penalties or losses. Real-time interaction between an agent and its surroundings is essential to RL. The agent must decide on its course of action based on a collection of inputs, where the inputs specify the environmental states. Over time, the agent aims to maximize these outcomes, which might be either rewards or penalties. In biological systems, this formulation is normal, but it has also shown to be quite effective for artificial agents. In actuality, deep RL is intrinsically interesting for a wide range of applications due to the mix of representation learning and goal-oriented behavior.

The use of cryptocurrencies will become more widespread as computer technology advances, and more newcomers will enter the market. Due to the need for a second reliable party, organized businesses adapt their business models. Smart contracts on the blockchain can automatically enforce contracts made between two unknown parties [1]. The second-largest blockchain platform, Ethereum, supported high-level programming languages for the first time and offers a runtime environment for virtually all Decentralized Financial applications [1]. Bitcoin also permits the development and execution of smart contracts, although it is affected by the nature of the programming language employed, and scarcely supports transactions except for confirming signatures. Certain security flaws can be quite expensive because smart contracts can allow a range of huge transactions. The topic of smart contracts for the Ethereum blockchain was extensively researched and surveyed by [1]. The study paper first analyzes some of the current or former contract vulnerabilities and their fixes, looks at the direction the smart contract market is headed in, and offers some recommendations for individuals conducting research in the area.

In energy trading systems, blockchain (BC) and artificial intelligence (AI) are frequently used individually (ETSs). When combined, these technologies can strengthen one another's skills and complement one another. Consensus algorithms (CAs) of BC and deep reinforcement learning (DRL) in ETS were thoroughly reviewed in [2]. While the immutability of prosumer transaction records is supported by distributed consensus, the flood of data generated opens the door for using AI algorithms for forecasting and addressing other data analytic-related challenges. Thus, the desire to create a secure and intelligent ETS by combining BC and AI. The study by [2] investigated the fundamental ideas, opportunities, models, ongoing research projects, and unresolved issues in the CA and DRL. Despite the current interest in each of these technologies, the review revealed that because of various unresolved challenges, little has been done to jointly leverage them in ETS. To fully utilize CA and DRL in ETS, fresh insights are therefore urgently needed.

Millions of grid-connected distributed controllable resources (DCR; for example, electric automobiles, controllable loads) are now able to deliver grid services like frequency management and demand response because of the quick development of the Internet of Things in smart grids[3]. These DCRs might combine to form a sizable virtual power plant network with a variety of features. This presents significant control and management difficulties, such as the cost of computing and communication, the complexity of optimization, the scalability constraint, the privacy of prosumers, etc. [3] proposed an efficient incentive-based autonomous DCR control and management framework to integrate a large number of DCRs to provide grid services, which simultaneously provides accurate active power adjustment to the grid. In addition, it optimizes DCR allocations, and maximizes the profits for all prosumers and system operators. The best incentives in a continuous action space to persuade prosumers to reduce their energy consumption were found by [3] using a model-free deep deterministic policy gradient-based strategy. The technique was included into the Hyperledger Fabric open-source blockchain technology, which enables controls and transaction management. Extensive experimental experiments were carried out utilizing real-world data by [3] to show the framework's efficacy.

1.2 Main Purpose

The purpose of the research is to explore and develop Deep Reinforcement Learning and Q-Learning algorithms in order to improve Ethereum cybersecurity in contract vulnerabilities, the smart contract market and research leadership in the area.

1.3 Research Objectives

The key research objectives of the research are to:

a) Ascertain how Deep Reinforcement Learning (Deep RL) can be used to handle complex, dynamic, and particularly high-dimensional cyber protection problems.
b) Investigate how Reinforcement Learning (RL) as a goal-oriented behavior that increases rewards and decreases penalties or losses, and real-time interaction between an agent and its surroundings, can be used to test a portfolio of cryptocurrencies.
c) Develop smart contracts on the Ethereum blockchain to automatically enforce contracts made between two unknown parties and improve Ethereum cybersecurity.

1.4 Research Questions

The research questions include the following:

a) How can complicated, dynamic, and especially high-dimensional cyber defense problems be handled using Deep Reinforcement Learning (Deep RL)?
b) How can a cryptocurrency portfolio be tested using Reinforcement Learning (RL) effectively?
c) How may smart contracts be developed for the Ethereum blockchain and used to enforce agreements that may enhance the security of Ethereum?

II. LITERATURE REVIEW

2.1 Deep Reinforcement Learning (DRL)

Deep reinforcement learning DRL is a rapidly growing field in machine learning that combines deep neural networks with reinforcement learning techniques to enable machines to learn from their environment and make decisions. Deep reinforcement learning (DRL) can be defined as a subfield of machine learning that involves the use of deep neural networks to solve reinforcement learning (RL) problems [4]. RL is a type of learning where an agent learns to make decisions by receiving feedback in the form of rewards or punishments. DRL is a promising approach to solving complex tasks that are difficult to program explicitly. In this literature review, we will examine the current state of research in the field of DRL, including the basic concepts, applications, challenges, and future directions.

2.1.1 Basic Concepts

DRL is a type of machine learning that uses neural networks with many layers to learn from data [5]. These networks are capable of learning complex representations of data, which can be used to make predictions or decisions. The foundation of DRL is built upon the principles of reinforcement learning, which involves an agent learning from its environment through trial and error to maximize its cumulative reward. DRL extends this approach by incorporating deep neural networks as function approximators to learn high-dimensional state-action value functions [6]. Deep neural networks consist of multiple layers of interconnected nodes, or neurons that transform input data into output data. Each layer learns a different representation of the input data, and the output of one layer serves as the input to the next layer. The final layer produces the output of the network. Deep reinforcement learning combines reinforcement learning with deep neural networks to learn policies for complex tasks. The use of deep neural networks allows the agent to learn complex representations of the state and action spaces, which can improve the quality of the learned policy [7]. The basic approach to deep reinforcement learning involves using a neural network to represent the policy, and training the network using a variant of the reinforcement-learning algorithm called the Q-learning algorithm [8]. The Q-learning algorithm uses a value function, called the Q-function, to estimate the expected cumulative reward for each state-action pair. The Q-function is learned using the Bellman equation.

2.1.2 Applications

In recent years, DRL has seen tremendous growth in terms of its applications across a wide range of domains, including robotics, gaming, healthcare, finance, autonomous driving, and natural language processing. DRL has been used to train robots to perform complex tasks such as grasping, object manipulation, and locomotion [4]. For instance, [8] used DRL to train a robot to grasp objects in cluttered environments, achieving success rates of over 90%. Similarly, [9] used DRL to train a simulated humanoid robot to walk and run, achieving results comparable to those of human experts. In addition, DRL has been used to train agents that can play complex games such as Go, chess, and poker at superhuman levels. In healthcare, DRL has been used to develop personalized treatment plans for patients with chronic diseases such as diabetes and hypertension. For example, [11] developed a DRL-based system that can automatically adjust insulin dosages for patients with diabetes, leading to improved glucose control compared to standard treatments. In finance, DRL has been used to develop trading algorithms that can learn to make profitable trades in complex and dynamic markets. For instance, [10] developed a DRL-based trading agent that learned to trade stocks and achieved higher returns than traditional trading strategies. DRL has also been applied to autonomous driving, where it has shown the potential to improve safety and reduce congestion [11]. Another important development in DRL is the use of meta-learning to improve the learning process. Meta-learning is the process of learning how to learn, and it has been applied to DRL to improve sample efficiency and generalization ([12]. One of the most successful meta-learning algorithms in DRL is Model-Agnostic Meta-Learning (MAML) introduced by [12]. MAML learns a good initialization of the network parameters that can be fine-tuned to adapt to new tasks quickly. With continued research and development, DRL is likely to find even more applications in the future.

2.1.3 Challenges

Despite its success in various applications, DRL faces several challenges that must be addressed to fully realize its potential. One of the primary challenges is the high computational cost required for training deep neural networks [14]. Training a DRL agent often requires large amounts of data and computation resources, which can make it impractical for some applications. This challenge is further compounded by the fact that DRL requires extensive trial and error, which can be time-consuming and resource-intensive. Another challenge is the need for more robust and reliable algorithms that can handle the non-stationarity of the environment and ensure stability during training [15]. In addition, the instability of the learning process is a problem in DRL. Due to the complex interactions between the deep neural network and the RL environment, the learning process can be sensitive to the choice of hyper parameters and the initialization of the network weights. To address this issue, researchers have proposed several techniques such as batch normalization, target networks, and prioritized experience replay. There is need for better exploration strategies. DRL agents often get stuck in local optima, where they fail to explore the entire state-action space, leading to suboptimal policies [11]. Several exploration strategies, such as epsilon-greedy exploration and adding noise to the action selection process have been proposed. Another challenge is the difficulty of transferring learned policies to new tasks or environments [16]. DRL agents are often trained on specific tasks or environments, and they may fail to generalize to new tasks or environments. Several transfer learning methods, such as fine-tuning and meta-learning are meant to solve this. DRL also faces challenges in terms of interpretability and safety [17]. Deep neural networks are often referred to as black boxes, which makes it difficult to understand how they make decisions. This lack of interpretability can be a significant challenge in applications such as healthcare and finance, where decisions made by DRL agents can have significant consequences. Ensuring the safety of DRL agents is another critical challenge, as they can learn to perform actions that may be unsafe or even dangerous in certain situations [18]. To address this issue, researchers have proposed several safety mechanisms, such as reward shaping and constraints on the actions taken by the agent. Overally, to address the challenges facing DRL, several promising directions for future research have been proposed. One direction is to develop more efficient algorithms that can reduce the computational cost of training deep neural networks. Another direction is to explore the use of transfer learning, where knowledge learned from one task can be transferred to another task to reduce the amount of training required. Finally, the development of more robust and reliable algorithms that can handle the non-stationarity of the environment and ensure stability during training is an important area for future research.

2.1.4 Future Directions

Despite the challenges, DRL is still a promising approach to solving complex RL problems using deep neural networks. It has shown remarkable success in a variety of domains, and there is still a lot of ongoing research to improve the performance and stability of DRL algorithms. DRL is expected to continue to grow and make significant contributions to the field of machine learning in the coming years. One of the key directions for future research is to improve the sample efficiency of DRL algorithms [19]. Sample efficiency refers to the ability of an algorithm to learn from limited amounts of data. Another important direction for future research is to improve the interpretability and safety of DRL agents [20]. Researchers are exploring several approaches to address these issues, such as developing more transparent neural network architectures, incorporating human-in-the-loop approaches, and developing techniques for verifying the safety of DRL agents. Finally, researchers are also exploring new applications of DRL in areas such as natural language processing, computer vision, and robotics.

2.2. Q-learning Algorithms

Q-Learning is a popular Reinforcement Learning (RL) algorithm that has been widely used in various applications, including robotics, game playing, and control systems. This literature review provides an overview of the Q-Learning algorithm and its variants, including their applications, strengths, and weaknesses. Q-Learning is a widely used reinforcement learning algorithm that enables an agent to learn an optimal policy by exploring its environment and receiving rewards for its actions. Q-Learning can handle environments with stochastic transitions and delayed rewards, making it a powerful algorithm for solving a wide range of reinforcement learning problems. Q-learning is a reinforcement learning algorithm that aims to find the optimal policy for a given Markov Decision Process (MDP). It is one of the most widely used algorithms in reinforcement learning, owing to its simplicity and effectiveness. Q-learning is a model-free algorithm, which means it does not require any prior knowledge of the MDP transition probabilities. In this paper, we will discuss the basic concepts of Q-learning, including its update rule, exploration-exploitation tradeoff, and convergence properties. The Q-learning algorithm is based on the Bellman equation, which expresses the optimal value function V*(s) in terms of the optimal action-value function Q*(s,a). The optimal action-value function Q*(s,a) is the expected sum of rewards obtained by taking action a in state s and then following the optimal policy thereafter. The Bellman equation is given by:

$$Q^*(s,a) = E[r + \gamma * \max\_a' Q^*(s',a') \mid s,a]$$

where r is the immediate reward obtained by taking action a in state s, s' is the next state, and γ is the discount factor that determines the importance of future rewards.

The Q-learning algorithm uses an iterative update rule to estimate the optimal action-value function Q*(s,a). At each time step t, the agent observes the current state s_t and takes an action a_t based on its current estimate of the action-value function Q. After taking the action, the agent observes the immediate reward r_t and the next state s_{t+1}. The update rule for Q is given by:

$$Q(s\_t,a\_t) = Q(s\_t,a\_t) + \alpha[r\_t + \gamma \max\_a' Q(s\_\{t+1\},a') - Q(s\_t,a\_t)]$$

where α is the learning rate that determines the rate at which the agent updates its estimates.

2.3. Ethereum Cybersecurity

A blockchain is essentially a digital ledger of transactions (DLT) that, upon duplication, is spread on the blockchain throughout the whole computer network. A blockchain is a data structure that keeps transactional information in the form of block chains and stored in many databases. The blockchain technology is a decentralized, rigid ledger that provides systems for recording transactions, managing resources, and storing records in a network where each node is linked to every other node and is referred to as a block [26]. Because the ledger is intended to enable transaction settlement and such settlement corresponds to all copies of the ledger documenting the transaction, the ledger's functionality depends on validators agreeing on its contents. Consensus among validators, also known as agreement, is therefore crucial for the sustainability of blockchain. Cryptography allows for the transmission and storage of data in a form that is only readable and usable by authorized parties. The process of turning well-known plain text into invisible content and vice versa is known as cryptography [26]. A digital payment system known as cryptocurrency was created utilizing Blockchain technology, allowing anyone to transfer and receive money digitally rather than physically [26]. Decentralized digital applications that enable users to engage in direct agreements and transactions to buy, sell, and trade products without the need for a middleman are run on an Ethereum Blockchain, which offers end-to-end security. The three most popular cryptocurrencies—Bitcoin, Litecoin, and Ethereum—as well as the part cyberattacks play are examined in the research report.

Ethereum is a platform that can be used to create organizations and apps, store assets, carry out transactions, and enable communications without the need for centralized oversight [27]. As the second-largest blockchain platform and the first to enable high-level programming languages for the implementation of smart contracts, Ethereum offers a runtime environment for virtually all decentralized financial applications. In addition to supporting the creation and execution of smart contracts, Bitcoin also mostly supports transactions, with the exception of verifying signatures [27]. Nevertheless, this support is influenced by the programming language employed. Certain security flaws can be quite expensive because smart contracts can allow a range of huge transactions. Smart contracts enable traceable and irreversible trusted transactions without the need for a third party. One of the most significant advantages of smart contracts over conventional contracts is the automatic execution of the outcomes upon fulfillment of the contract's requirements. Waiting for the results of human execution is useless. Alternatively, to put it another way, smart contracts do not require confidence. Traditional contracts are impacted by a variety of factors, such as automation dimensions, subjective and objective dimensions, cost dimensions, execution time dimensions, default penalty dimensions, scope of application dimensions, etc. The smart contract must consider and specify each possible event before placing a wager. After the incident, the operation will be carried out automatically and strictly in accordance with the pre-determined contract content [27]. Hence, it is possible to deal with the problems that come up with traditional contracts.

The Ethereum blockchain supports distributed, public, and immutable smart contracts. They do, however, have a lot of weaknesses that result from developers' coding mistakes. Between 2016 and 2018, seven cybersecurity incidents involving Ethereum smart contracts occurred, resulting in financial losses that are thought to have exceeded US$ 289 million [28]. Two of these catastrophes were brought on by reentrancy vulnerability, and the results extended far beyond monetary loss. There are a number of reentrancy countermeasures that can be deployed, based on predetermined patterns, to stop vulnerability exploitation before the deployment of a smart contract; however, these countermeasures have a number of drawbacks. By providing a solution that determines the discrepancy between a smart contract's contract balance and the total of all participants' balances both before and after any operation in a transaction that changes its state, the research study aims to help developers strengthen the security of smart contracts. This strategy can offer a detection and defensive mechanism against reentrancy assaults during the execution of any smart contract, according to proof-of-concept implementations [28]. With the launch of Ethereum smart contracts in 2015, there have been a number of instances where disputes or other problems have arisen as a result of the functioning of smart contracts that contained a certain quantity of ether [29]. Reentrancy vulnerability was the cause for these instances. Despite these events, smart contracts are becoming more and more common; yet, this also makes them more of a target for attackers.

Because they are much like any other executable apps that run on computers, smart contracts are one of the most often exploited attack vectors against Ethereum. However, as stated by [28], smart contracts are more vulnerable in terms of cybersecurity because the smart contracts work on top of an immutable blockchain and are connected to a digital fortune that may be worth millions of dollars. Consequently, even if the smart contracts have defects, once they are distributed on the blockchain it may be impossible and extremely difficult to change them (the "code is law" notion).

Since the contracts run on top of immutable technology, numerous countermeasures are intended to identify security issues during the development stage. Developers of smart contracts must take action to fix these issues during the execution phase rather than relying just on the development phase. The issue is how to protect a smart contract during execution despite the fact that it cannot be changed and has defects. One of the frequent dangers to the Ethereum blockchain, which is connected to the Solidity programming language, is reentrancy assaults. Attacks take place when a malicious party uses a smart contract's external call to trick it into running extra code by using a fallback function to call back to the contract [28]. Several techniques are used to guard against the reentrancy vulnerability in smart contracts. These proactive techniques, which are used prior to the implementation of the smart contracts, include security based on programming languages, security based on the creation of smart contracts, and vulnerability-detection tools for Ethereum smart contracts [28]. Reentrancy vulnerability can be found using a number of smart contract vulnerability detection tools. The introduction of several high-level programming languages enables the secure development of smart contracts. There are a few ways to improve the smart contract programming model to help developers reduce or eliminate the reentrancy risk. Ethereum smart contract best practices is one of them, which was first introduced by ConsenSys Diligence. This gives programmers of Solidity a foundational understanding of security issues [28]. Also, a number of suggestions are given to help Ethereum smart contract developers stay clear of code problems. All Ethereum smart contract development must follow protocol-specific advice to guard against reentrancy risk, such as avoiding state changes following external calls. The other suggestions are Solidity-specific, and they may be instructive for those creating smart contracts in other languages [28].

2.4. Cyber risks for Cryptocurrencies and Blockchain Technologies

In recent years, the use of cryptocurrencies and blockchain technology has exploded, with Bitcoin being the most well-known and popular one. The risk of cyberattacks has increased along with the adoption of these technologies, though. Over the past ten years, blockchain technology has become more prevalent. As a result, blockchain-based applications have impacted several industry sectors and have amassed substantial international user support. The most widely used blockchain applications are now cryptocurrencies [32].

Three types of blockchain technology exist: private, public or permissionless, and federated or consortium blockchain. Whether or not authenticated trusted participants are required depends on the number of companies involved in maintaining the digital ledger [34]. According to the perception of private and consortium blockchains as being permissioned, it is necessary for a specific management entity to offer access privileges to reliable and vetted members. The private blockchains Multichain, Monax, and Quorum are a few examples. A consortium blockchain is also governed by at least two different companies. The operation is entirely decentralized and ad hoc with a public blockchain, which allows anybody to access or write the data stored in the blockchain network without needing permission from any authority. Important examples are Bitcoin, Ether, and Monero. A consensus-based approach is used by public blockchains to choose which user will submit a block. Users with suspicions can cooperate in a blockchain network thanks to consensus procedures [34].

Cyberattacks that target cryptocurrencies and the fundamentals of blockchain technology have grown, costing consumers and companies millions of euros [32]. The following are some of the cyber dangers that cryptocurrencies and blockchain technologies are exposed to:

1. Phishing attacks35
2. Illegitimate trading platforms [35].
3. Use of third party applications [35].
4. Crypto-malware and mining malware [35].
5. Cryptocurrency account security [33].
6. Unregulated cryptocurrency exchanges [34].
7. User confusion [35]
8. Risk of reentrancy attacks in smart contracts [37].
9. Hacking [36].
10. Fraudulent ICOs [37].
11. 51% Attacks [37].
12. Software’s errors [34].
13. Risks associated with hash function [34]
14. Digital signature risks [34].

Mitigation Measures

Our understanding of money and transactions has changed dramatically as a result of the adoption of cryptocurrencies and blockchain technology. These technologies do, however, also pose important cyber threats that need to be handled. The dangers connected with cryptocurrencies and blockchain technologies can be reduced by putting in place the right security measures and carrying out regular audits, allowing these technologies to develop and expand. The following are some of the mitigation measures:

- *Multi-Factor Authentication:* Cryptocurrency wallets can benefit from an additional degree of protection provided by multi-factor authentication. This can involve the use of one-time passwords as well as biometric identification methods like fingerprint or face recognition.
- *Cold Storage:* Cold storage refers to storing cryptocurrency offline, such as on a hardware wallet or paper wallet. By doing this, money can be shielded from theft in the event of a hack or other cyber-attack.
- *Due Diligence:* Before investing in any blockchain startup or initial coin offering (ICO), investors should do their research. This may entail looking into the project's personnel, reading the white paper, and examining the token economics.
- *Consensus Mechanisms:* By making it difficult for any single entity to dominate the network, consensus mechanisms like proof-of-work or proof-of-stake can aid in preventing 51% attacks.

2.5. Proof-of-Stake Ethereum

The Proof-of-Stake (PoS) consensus algorithm is an alternative to the conventional Proof-of-Work (PoW) consensus algorithm employed by cryptocurrencies like Bitcoin [40]. Participants in a blockchain network can validate transactions and add new blocks to the chain using proof-of-stake (PoS). For a very long time, Ethereum, the second-largest cryptocurrency by market capitalization, has planned to switch from PoW to PoS. [41]. Validators in Ethereum PoS must make a minimum 32 ETH deposit in order to use the network. When they validate transactions and include new blocks to the blockchain, they get rewarded. A validator's chances of being chosen to validate transactions and receive rewards increase with the amount of ETH they stake. A validator, however, runs the danger of being fined a portion of their staked ETH if they act maliciously or fail to fulfill their duties [42]. Proof-of-stake (PoS) relies on validators staking their own cryptocurrency as collateral to participate in the network [43]. Thus, in PoS, validators are picked at random based on the amount of cryptocurrency they hold and are prepared to "stake" or lock up as collateral. The amount staked is a measure of the validator's commitment to the network, and the more they have staked, the more likely it is that they will be selected to validate transactions and add new blocks to the chain. Via a series of updates known as Ethereum 2.0, one of the biggest blockchain networks in the world, Ethereum, is in the process of switching from PoW to PoS [44]. This change is being made primarily to strengthen network security, decrease energy usage, and improve scalability. In Ethereum PoS, validators must stake a minimum of 32 ETH in order to access the network. This entails that they must secure this sum of ETH as collateral, which they run the risk of losing if they engage in malicious behavior or are negligent in their validator responsibilities. Transactions must be verified by validators, who also suggest new blocks and cast votes for or against other validators' proposed blocks. The more ETH a validator invests, the better their chances are of being chosen to carry out these duties. The rewards for validating transactions and adding new blocks to the chain are paid out in ETH [45]. PoS Ethereum is anticipated to increase network efficiency and accessibility by lowering transaction costs and speeding up transaction processing. The fact that PoS Ethereum uses a lot less energy than PoW is another advantage [46]. To participate in the PoS Ethereum network, which uses much less energy, validators merely need to stake their own ETH and host a node. PoS Ethereum also lowers the possibility of network centralization [47], which is another advantage. The risk of centralization and collusion is diminished in PoS Ethereum because validators are selected at random based on the amount of ETH they have pledged. Nevertheless, the move to Ethereum PoS is an exciting milestone for the cryptocurrency world because it is expected to significantly increase the network's scalability and durability.

III. RESEARCH METHODOLOGY: DESIGN SCIENCE RESEARCH

The Design Science Research Paradigm as applied in Information Systems research was used in this research. This paradigm is hinged on the idea that information and understanding of a design problem and its solution are attained in the crafting of an artefact [48]. In this research the main aim was to to explore and develop Deep Reinforcement Learning and Q-Learning algorithms in order to improve Ethereum cybersecurity in contract vulnerabilities, the smart contract market and research leadership in the area. This aim resonated well with the Design Science approach which seeks to provide an artefact in the form of a construct or a model [48]. In this case the proposed constructs were in the form of Deep Reinforcement Learning and Q-Learning algorithms designed to improve Ethereum cybersecurity. In this research the seven guidelines that the Design Science employs were followed throughout the study as detailed below according to [48].

Guideline 1: Design as an artefact

The product or output of Design-science research is a viable artefact that can take the form of a method, construct or model that will address an identified challenge in an organization [48]. In this study, the artefact was in the form of Deep Reinforcement Learning and Q-Learning algorithms meant to improve Ethereum cybersecurity in contract vulnerabilities, the smart contract market and research leadership in the area.

Guideline 2: Problem relevance

[48], indicate that the aim of design-science research is to come up with technology-based solutions that speak to identified business challenges. In this research, it was noted that smart contracts generated from the Ethereum platform were vulnerable to cyber-attacks. As such, there was need to come up with Deep Reinforcement Learning (Deep RL) algorithms capable of handling complex, dynamic, and particularly high-dimensional cyber protection problems.

Guideline 3: Design evaluation

According to [48], the design artefact's quality and effectiveness must be thoroughly proven by means of well-performed evaluation methods. In order to stick to this guideline, the researchers presented the Deep Reinforcement Learning and Q-Learning algorithms to subject matter experts and reviewers. Their feedback was then used for the purposes of fine tuning the algorithms for the security of the Ethereum.

Guideline 4: Research contributions

[48] recommend that real design-science research has to deliver clear and provable contributions in the areas of the design artefact, design foundations, or design methodologies. In this study, the researchers envisaged making a contribution through the development of the Deep Reinforcement Learning and Q-Learning algorithms tailored to improve Ethereum cybersecurity in smart contracts.

Guideline 5: Research rigor

Design-science research heavily depends on the use of rigorous methods in the building and appraisal of the design artefact [48]. In this study, research rigor was accomplished through comprehensive literature review and testing of the Deep Reinforcement Learning and Q-Learning algorithms under different situations. In addition to that, continuous review by subject matter experts in the domain of cybersecurity helped to attain the rigor that is expected when applying Design Science research.

Guideline 6: Design as a search process

Design is basically a search process that is meant to find out the best solution to an identified challenge using the available means whilst sticking to the applicable laws [48]. In this research the researchers stuck to research ethics in conducting the research.

Guideline 7: Dissemination of research

[48] recommends that Design-science research must be presented to technologists as well as management in order for them to reap the benefits. In order to fulfil this, the Deep Reinforcement Learning and Q-Learning algorithms, their usefulness and originality was presented to researchers and other appropriate audiences such as cybersecurity professionals. In addition, the study was also published in peer reviewed journals in order to reach the academia and research space.

Similar to JavaScript, Solidity is a language that lets you create contracts and compile them into EVM bytecode. Nowadays, it is Ethereum's most widely used and flagship language. The Proof of Stake consensus illustrated by Solidity Ethereum contract programming was used in this research study. In theory, consensus on a blockchain can be reached by designating a central authority to decide which transactions have completed settlement. A blockchain with such limitations is described by a protocol. A permissionless blockchain lacks a central authority, making it more difficult for such a blockchain to reach consensus. [39] asserts that the Proof-of-Work economic protocol will address the consensus problem in a permissionless blockchain (PoW). To update the blockchain, validators must compete in a PoW system. Proof-of-Stake (PoS) makes an effort to address the energy consumption issue brought on by PoW. As a result, PoS eliminates PoW's rivalry by choosing stakeholders at random to add to the blockchain.

There are many other types of consensus that can be considered but PoS was preferred for this study:

- ❖ (PoW) Proof of Work (Bitcoin, Ethereum, …)
- ❖ ***(PoS) Proof of Stake (Ethereum in future)***
- ❖ (PoI) Proof of Importance (used in NEM)
- ❖ (PBFT) Practical Byzantine Fault Tolerance (Hyperledger Fabric)

- ❖ (FBFT) Federated Byzantine Fault Tolerance (Ripple, Stellar)
- ❖ (DPoS) Delegated Proof of Stake
- ❖ (PoET) Proof of Elapsed Time (Hyperledger Sawtooth)

The Improved Delegated Proof of Stake (DPoS) Algorithm is illustrated below on Fig 1. We improve the conventional DPoS consensus algorithm and propose a reputation-based delegated proof of stake consensus algorithm, called Reputation-DPoS, in an effort to address the issues with the current DPoS (Delegated Proof of Stake) consensus algorithm, such as low voter enthusiasm and challenges dealing with malicious nodes.

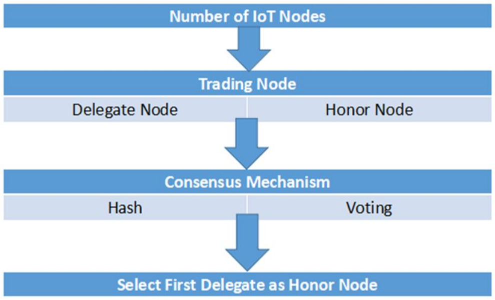

Figure 1: The Improved Delegated Proof of Stake (DPoS) Algorithm (Source: [49])

The algorithm for selection of honor delegate node is coded as follows:

```
Input: Voting results among N nodes, nonce
Output: honor delegates
Transmit (nonce, N) Ni : Hash (hash (BlockHead), nonce)
            while (Hash (hash (BlockHead), nonce) > delegates)
            Calculate the vote of each delegate and sort them
                if number of vote of first delegate ≥ second delegate
                                Select the first delegate as the honor delegate
                      If the votes of different delegates are same
                                Calculate the vote deviation percentage
                                Select small deviation percentage delegate as honor node
                      end if
                end if
```

IV. Data Analysis and Key Findings

Deep Reinforcement Learning (DRL) is a rapidly growing field of machine learning that combines deep neural networks with reinforcement learning techniques to enable machines to learn from their environment and make decisions. It involves using a neural network to represent the policy, and training the network using a variant of the reinforcement learning algorithm called the Q-learning algorithm. DRL has seen tremendous growth in recent years in terms of its applications across a wide range of domains, including robotics, gaming, healthcare, finance, autonomous driving, and natural language processing. Meta-learning is the process of learning how to learn, and has been applied to DRL to improve sample efficiency and generalization. However, it faces several challenges, such as the high computational cost required for training deep neural networks, the instability of the learning process, the need for better exploration strategies, the difficulty of transferring learned policies to new tasks or environments, and the lack of interpretability and safety. Reinforcement Learning (RL) is a promising approach to solving complex RL problems using deep neural networks, but there is still a lot of ongoing research to improve the performance and stability of DRL algorithms. To address this, researchers have proposed techniques such as batch normalization, target networks, and prioritized experience replay. Additionally, researchers are exploring new applications of DRL in areas such as natural language processing, computer vision, and robotics. Q-Learning is a popular reinforcement learning algorithm that enables an agent to learn an optimal policy by exploring its environment and receiving rewards for its actions. Q-Learning is a reinforcement learning algorithm that uses an epsilon-greedy exploration strategy to balance exploration and exploitation. It has several variants, including Deep Q-Network (DQN), Double Q-Learning, and Dueling Q-Networks. Blockchain technology is a decentralized, rigid ledger that provides systems for recording transactions, managing resources, and storing records. Cryptocurrency is a digital payment system that allows anyone to transfer and receive money digitally rather than physically. Ethereum is a platform for decentralized digital applications that enable users to engage in direct agreements and transactions without the need for a middleman. Cryptocurrencies such as Bitcoin, Litecoin, and Ethereum offer end-to-end security.

The research study discussed the limitations of existing techniques for identifying reentrancy vulnerabilities in smart contracts, such as relying on entire patterns and the unique characteristics of these patterns. It proposes a remedy to get around these restrictions by offering a prevention technique to safeguard the smart contract and a detection technique to catch the attacker. It also suggests that two values keep the funds in the smart contract in any smart contract that administers a fund for numerous participants: the protocol layer upholds the first value, which is the contract balance displayed as an address in Solidity as *address(this).balance*, while the application layer keeps track of the second value.

Cyberattacks that target cryptocurrencies and blockchain technology have grown, costing consumers and companies millions of euros. Crypto-malware and mining malware are two forms of dangerous software that can be used to harvest bitcoins from servers or computers belonging to other people. Crypto investors rely on third-party applications to manage their digital assets, but this increases their exposure to cybersecurity concerns. Cryptocurrencies are riskier than traditional ones because investors are solely responsible for safeguarding their private keys and making sure they are not exposed to hackers. Blockchain technology has tamper-evident and tamper-proof digital ledgers without any central authority, allowing users to use a shared ledger to keep transactions inside a community. However, there is no single agency, institution, or governing authority responsible and accountable for the manufacture, movement, or management of traditional currencies, leading to a fraudulent and hacker-friendly environment. Fraudulent ICOs are a well-liked strategy used by blockchain businesses to acquire capital, but there have been many instances of fraudulent ICOs. Smart contracts are more sensitive when it comes to cybersecurity because they operate on top of an unchallengeable blockchain, making it difficult or impossible to change them. Mitigation measures include multi-factor authentication, cold storage, due diligence, and consensus mechanisms.

A blockchain that enables smart contracts is Ethereum. Smart contracts, which are pieces of general-purpose computer code, have been used to implement cryptocurrency and crowdsourcing projects (ICOs). The security of smart contracts in Ethereum is a significant issue. Smart contracts are unchangeable after deployment, in contrast to conventional software development. Hence, smart contract flaws and vulnerabilities might cause enormous financial losses. Smart contract developers are urged to reuse code from reliable sources in order to eliminate the danger of producing flawed code.

The performance of various consensus algorithms are shown on Table 1 below.

TABLE 1: Performance of various consensus algorithm (Source: [49])

Consensus Mechanism	**Proof-of-Work (PoW)**	**Proof-of-stake (PoS)**	**Improved delegate PoS (DPoS)**
Mechanism for bock generation	Computing power	Stake	Stake votes
Security issues	Constant power	Inactive nodes	Malicious nodes
Energy consumption	Very High	High	Low
Average block generation time	10 min	65 sec	5 sec
Reliability	High	Low	Low
Robustness	High	High	High

The complete architecture for solving the Ethereum Cybersecurity problem is summarized by Fig 2 below, which is the significant contribution from the researchers.

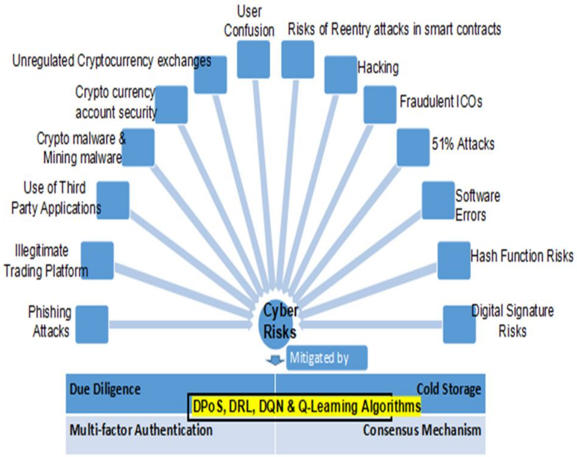
User Confusion
Risks of Reentry attacks in smart contracts
Unregulated Cryptocurrency exchanges
Hacking
Crypto currency account security
Fraudulent ICOs
Crypto malware & Mining malware
51% Attacks
Use of Third Party Applications
Software Errors
Illegitimate Trading Platform
Hash Function Risks
Phishing Attacks
Cyber Risks
Digital Signature Risks
Mitigated by
Due Diligence
Cold Storage
DPoS, DRL, DQN & Q-Learning Algorithms
Multi-factor Authentication
Consensus Mechanism

Figure 2: Ethereum Cybersecurity with DPoS Algorithm

The automatic execution of the terms agreed upon between two untrusted parties is made possible by the blockchain-based smart contract. The Ethereum blockchain-based smart contracts suffer from a number of security flaws, which occasionally cause them to behave improperly. Due to the fact that a smart contract may store millions of dollars in bitcoin, these security flaws could result in catastrophic losses. This study presents a thorough analysis of the security flaws in the Ethereum network. There are a number of blockchain smart contract development platforms available, each with a unique set of features, opportunities, and difficulties. Developers can choose from a variety of blockchain systems, with Ethereum being the most popular and well-known. Solidity-written functions, events, and state variables make up the majority of Ethereum smart contracts. The executable code, contract address, state made up of private storage, and balance in virtual currency make up an Ethereum Smart Contract account (Ether). With a contract invoking transactions to its unique address, a smart contract can be triggered, with some parameters such invocation data and payment in the form of ether as transaction fees (Gas). Due to the immutable nature of the blockchain, a smart contract cannot be altered after being deployed on it. Thus, when developing smart contracts, developers should take security flaws and recommended practices into account. Blockchain has a reputation issue since it is frequently associated with cryptocurrencies and perceived as a place where fraudsters and attackers thrive.

V. CONCLUSION

In conclusion, our study has shown how deep reinforcement learning and Q-learning algorithms can improve the Ethereum blockchain network's security. By employing these methods, cyberattacks may be detected and prevented with greater accuracy, enhancing network security as a whole. According to the findings of the research presented in this paper, many sorts of assaults, such as double-spending attacks and 51% attacks, can be successfully identified and mitigated using the suggested approach. In order to improve over time, the system may also be able to adapt to new assault scenarios and learn from its mistakes. As a result, using Q-learning and deep reinforcement learning algorithms offers a viable way to raise the security of blockchain networks like Ethereum. In addition, Q-learning algorithms can be utilized to create a successful plan for seeing and stopping possible assaults before they happen. Combining these two methods makes it possible to design a cybersecurity system that is both proactive and reactive. To further increase the resilience of blockchain networks against cyberattacks, further research in this field may examine the application of additional machine learning methods or the incorporation of extra security mechanisms. Overall, the findings of this study show that the Ethereum network's security may be greatly increased by the application of deep reinforcement learning and Q-learning algorithms. Even if there is still much to be done in this area, we think that this strategy offers a solid framework for future study and research in the topic of blockchain cybersecurity. Ethereum is a blockchain that enables smart contracts, which are used to implement cryptocurrency and crowdsourcing projects. Smart contracts are unchangeable after deployment, making them vulnerable to security flaws and vulnerabilities. Smart contract developers are urged to reuse code from reliable sources in order to eliminate the danger of producing flawed code. High-activity verified smart contracts typically include a limited amount of source code and have at least two subcontracts and libraries, demonstrating unique complexity features. This study presents a thorough analysis of the security flaws in the Ethereum network. Solidity-written functions, events, and state variables make up the majority of Ethereum smart contracts, and the executable code, contract address, state made up of private storage, and balance in virtual currency make up an Ethereum Smart Contract account (Ether).

REFERENCES

[1] Ma, T. (2023). Cybersecurity and Ethereum Security Vulnerabilities Analysis. *Highlights in Science, Engineering and Technology*, *34*, 375-381.

[2] Jogunola, O. et al., "Consensus Algorithms and Deep Reinforcement Learning in Energy Market: A Review," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4211-4227, 15 March15, 2021, doi: 10.1109/JIOT.2020.3032162.

[3] Ma R., Yi Z., Xiang Y., Shi D., Xu C., and Wu H., "A Blockchain-Enabled Demand Management and Control Framework Driven by Deep Reinforcement Learning," in *IEEE Transactions on Industrial Electronics*, vol. 70, no. 1, pp. 430-440, Jan. 2023, doi: 10.1109/TIE.2022.3146631.

[4] Liu, R., Nageotte, F., Zanne, P., de Mathelin, M., & Dresp-Langley, B. (2021). Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review. *Robotics*, *10*(1), 22.

[5] Wang, X., Wang, S., Liang, X., Zhao, D., Huang, J., Xu, X., ... & Miao, Q. (2022). Deep reinforcement learning: a survey. *IEEE Transactions on Neural Networks and Learning Systems*.

[6] Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, *34*(6), 26-38.

[7] Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., & Pérez, P. (2021). Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *23*(6), 4909-4926.

[8] Gu, S., Holly, E., Lillicrap, T., & Levine, S. (2017, May). Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In 2017 IEEE international conference on robotics and automation (ICRA) (pp. 3389-3396). IEEE.

[9] Gu, G. Y., Zhu, J., Zhu, L. M., & Zhu, X. (2017). A survey on dielectric elastomer actuators for soft robots. Bioinspiration & biomimetics, 12(1), 011003.

[10] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[11] Hong, Z. W., Shann, T. Y., Su, S. Y., Chang, Y. H., Fu, T. J., & Lee, C. Y. (2018). Diversity-driven exploration strategy for deep reinforcement learning. *Advances in neural information processing systems*, *31*.

[12] Finn, C., Abbeel, P., & Levine, S. (2017, July). Model-agnostic meta-learning for fast adaptation of deep networks. In International conference on machine learning (pp. 1126-1135). PMLR.

[13] Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015, June). Trust region policy optimization. In *International conference on machine learning* (pp. 1889-1897). PMLR.

[14] Sarwar, S. S., Ankit, A., & Roy, K. (2019). Incremental learning in deep convolutional neural

networks using partial network sharing. *IEEE Access*, *8*, 4615-4628.

[15] Gürtler, N., Büchler, D., & Martius, G. (2021). Hierarchical reinforcement learning with timed subgoals. *Advances in Neural Information Processing Systems*, *34*, 21732-21743.

[16] Zhu, Z., Lin, K., & Zhou, J. (2020). Transfer learning in deep reinforcement learning: A survey. *arXiv preprint arXiv:2009.07888*.

[17] Li, C., Zheng, P., Yin, Y., Wang, B., & Wang, L. (2023). Deep reinforcement learning in smart manufacturing: A review and prospects. CIRP Journal of Manufacturing Science and Technology, 40, 75-101.

[18] Pereira, A., & Thomas, C. (2020). Challenges of machine learning applied to safety-critical cyber-physical systems. *Machine Learning and Knowledge Extraction*, *2*(4), 579-602.

[19] Yang, T., Tang, H., Bai, C., Liu, J., Hao, J., Meng, Z., ... & Wang, Z. (2021). Exploration in deep reinforcement learning: a comprehensive survey. *arXiv preprint arXiv:2109.06668*.

[20] Whittlestone, J., Arulkumaran, K., & Crosby, M. (2021). The societal implications of deep reinforcement learning. *Journal of Artificial Intelligence Research*, *70*, 1003-1030.

[21] Kumar, V., & Webster, M. (2021). Importance Sampling based Exploration in Q Learning. *arXiv preprint arXiv:2107.00602*.

[22] Sharma, J., Andersen, P. A., Granmo, O. C., & Goodwin, M. (2020). Deep Q-learning with Q-matrix transfer learning for novel fire evacuation environment. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *51*(12), 7363-7381.

[23] Hase, H., Azampour, M. F., Tirindelli, M., Paschali, M., Simson, W., Fatemizadeh, E., & Navab, N. (2020, October). Ultrasound-guided robotic navigation with deep reinforcement learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5534-5541). IEEE.

[24] Wang, C., Wang, J., Shen, Y., & Zhang, X. (2019). Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach. *IEEE Transactions on Vehicular Technology*, *68*(3), 2124-2136.

[25] Wang, X., Ke, L., Qiao, Z., & Chai, X. (2020). Large-scale traffic signal control using a novel multiagent reinforcement learning. *IEEE transactions on cybernetics*, *51*(1), 174-187.

[26] Toqeer, A., Alghamdi, T., Nadeem, A., Perwej, Y., & Thabet, M. (2021). Cyber security intelligence and ethereum blockchain technology for e-commerce. *International Journal*, *9*(7).

[27] Ma, T. (2023). Cybersecurity and Ethereum Security Vulnerabilities Analysis. *Highlights in Science, Engineering and Technology*, *34*, 375–381. https://doi.org/10.54097/hset.v34i.5498.

[28] Alkhalifah, A., Ng, A., Watters, P. A., & Kayes, A. S. M. (2021). A mechanism to detect and prevent ethereum blockchain smart contract reentrancy attacks. *Frontiers in Computer Science*, *3*, 598780.

[29] Alkhalifah, A., Ng, A., Chowdhury, M. J. M., Kayes, A. S. M., and Watters, P. A. (2019). "An empirical analysis of blockchain cybersecurity incidents," in 2019 IEEE Asia-Pacific conference on computer science and data engineering (CSDE), Melbourne, Australia, December 9–11, 2019 (IEEE), 1–8. doi:10.1109/CSDE48274.2019.9162381.

[30] Alkhalifah, A., Ng, A., Kayes, A. S. M., Chowdhury, J., Alazab, M., and Watters, P. A. (2020). "A taxonomy of blockchain threats and vulnerabilities," in *Blockchain for cybersecurity and privacy: architectures, challenges and applications.* (Boca Raton, FL: CRC Press Taylor & Francis), Chap. 1, 1–27.

[31] Samreen, N. F., and Alalfi, M. H. (2020). "Reentrancy vulnerability identification in Ethereum smart contracts," The institute of electrical and electronics engineers, Inc.(IEEE) conference proceedings, London, ON, February 18, 2020 (IEEE), 22–29.

[32] Ramos, S., Lela, Melon, L., Ellul, J., (2022). Exploring Blockchains Cyber Security Techno-Regulatory Gap. An Application to Crypto-Asset Regulation in the EU. *Conference Paper, June 2022.*

[33] Amos, Z., (2023) The Cybersecurity Risks of Cryptocurrency; Available on *: https://cybersecurity-magazine.com/the-cybersecurity-risks-of-cryptocurrency/*

[34] Yassine M, Alazab M,Shojafar M, Romdhani I (2020). Blockchain for Cybersecurity and Privacy: Architectures, Challenges, and Applications. *CRC Press, 2020.*

[35] Roohparvar (2022). The Cybersecurity Risks of Cryptocurrency; Available on:*https://www.infoguardsecurity.com/the-cybersecurity-risks-of-cryptocurrency/.*

[36] Samreen, N. F., and Alalfi, M. H. (2020). "Reentrancy vulnerability identification in Ethereum smart contracts," *The institute of electrical and electronics engineers, Inc.(IEEE) conference proceedings, London, ON, February 18, 2020 (IEEE), 22–29.*

[37] Hughes, A., Park, A, Kietzmann, J, Archer-Brown, C. (2019). Beyond Bitcoin: What blockchain and distributed ledger technologies mean for firms. *Bus. Horiz. 2019, 62, 273–281*

[38] Madnick, S. (2020). Blockchain isn't as unbreakable as you think. MIT Sloan Manag. Rev. 61 (2), 65–70. *doi:10.2139/ssrn.3542542*

[39] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, 21260.

[40] Wendl, M., Doan, M. H., & Sassen, R. (2023). The environmental impact of cryptocurrencies using proof of work and proof of stake consensus algorithms: A systematic review. *Journal of Environmental Management*, *326*, 116530.

[41] Arslanian, H. (2022). Ethereum. In *The Book of Crypto: The Complete Guide to Understanding*

Bitcoin, Cryptocurrencies and Digital Assets (pp. 91-98). Cham: Springer International Publishing.

[42] Bhudia, A., Cartwright, A., Cartwright, E., Hernandez-Castro, J., & Hurley-Smith, D. (2022, August). Extortion of a Staking Pool in a Proof-of-Stake Consensus Mechanism. In *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)* (pp. 1-6). IEEE.

[43] Gundaboina, L., Badotra, S., & Tanwar, S. (2022, March). Reducing resource and energy consumption in cryptocurrency mining by using both proof-of-stake algorithm and renewable energy. In *2022 International Mobile and Embedded Technology Conference (MECON)* (pp. 605-610). IEEE.

[44] Scharfman, J., & Scharfman, J. (2022). Additional topics in blockchain and distributed ledger technology. *Cryptocurrency Compliance and Operations: Digital Assets, Blockchain and DeFi*, 137-153.

[45] Arslan, C., Sipahioğlu, S., Şafak, E., Gözütok, M., & Köprülü, T. (2021). Comparative analysis and modern applications of PoW, PoS, PPoS blockchain consensus mechanisms and new distributed ledger technologies. *Advances in Science, Technology and Engineering Systems Journal*, *6*(5), 279-290.

[46] El-Hajj, M., Fadlallah, A., Chamoun, M., & Serhrouchni, A. (2019). A survey of internet of things (IoT) authentication schemes. *Sensors*, *19*(5), 1141.

[47] Parham, A., & Breitinger, C. (2022). Non-fungible Tokens: Promise or Peril?. *arXiv preprint arXiv:2202.06354*.

[48] Hevner, Alan R.; March, Salvatore T.; Park, Jinsoo; and Ram, Sudha. (2004). *Design Science in Information Systems Research*, *MIS Quarterly, (28: 1)*.

[49] Hu, Q., Yan, B., Han, Y., & Yu, J. (2021). An improved delegated proof of stake consensus algorithm. *Procedia Computer Science*, *187*, 341-346.

Biographies and Photographs

Professor Gabriel Kabanda

Professor Gabriel Kabanda the Most Notable and Top Distinguished Full Professor of Computer Science, Information Systems, Cybersecurity, Machine Learning and Big Data Analytics. He is an Adjunct Professor of Machine Learning and Visiting Professor of MIS at Woxsen University, Hyderabad, India; Adjunct Professor of Cybersecurity at California State University, Chico (USA); Professor of Applied Business Informatics at the University of Zimbabwe Business School since January 2000, and Professor of Big Data Science in the Department of Computer Science at the National University of Science and Technology (NUST). Gabriel is a Fellow of the African Scientific Institute (USA); Fellow and Vice President of Zimbabwe Academy of Sciences; Secretary General of the Africa-Asia Dialogue Network; and an Editor of 10 international refereed journals. Professor Kabanda has published 126 research papers in international refereed journals. Gabriel holds a Post-Doctorate of Science, Doctor of Science (D.Sc.) in Computer Science from Atlantic International University (USA), a Ph.D. degree in Computer Science (California, PWU), Master of Science in Computer Science (Swansea University, United Kingdom), B.Sc. in Mathematics and Physics (University of Zimbabwe), and four diplomas and certificates.

Dr. Colletor Tendeukai Chipfumbu

Lecturer Information Systems, Cybersecurity, Machine Learning and Artificial Intelligence. She is a lecturer for Information systems specializing in teaching Information security, Cyber security, Artificial intelligence and Machine learning at the Midlands State University in Zimbabwe since 2008. She is an Editor of 1 international refereed journal and has published refereed journals articles in different journals. Colletor holds a Ph.D. degree in Information Technology (Nelson Mandela Metropolitan University), Master of Science Information Systems Management (Midlands State University, Zimbabwe), B.Sc. in Information Systems (Midlands State University, Zimbabwe), and a Post Graduate Diploma in Education.

Mr.Tinashe Chingoriwo

Mr. Tinashe Chingoriwo is a DPhil Candidate in Information Technology with the Zimbabwe Open University. He holds a Master of Business Administration degree (Zimbabwe Open University), Master of Science in Information Systems Management (Midlands State University, Zimbabwe), B.Sc. Honors degree in Computer Science (University of Zimbabwe), Certificate in Project Management (University of Zimbabwe) and several ICT certifications in Project Management (PMP), Business Analysis (PMI-PBA), Agile(PMI-ACP),Risk Management (PMI-RMP) from the Project Management Institute (USA).He is also an IBM Certified Mobile Application Developer for Worklight (IBM, USA) and an Oracle Certified Java SE6 Programmer (Oracle University, USA).Tinashe’s research interests include Cybersecurity, Machine Learning , Business Process Re-engineering and Business Process Management.