future internet

MDPI

Review

Integrating a Blockchain-Based Governance Framework for Responsible AI

Rameez Asif *, Syed Raheel Hassan and Gerard Parr

School of Computing Sciences, University of East Anglia, Norwich Research Park, Norwich NR4 7TJ, UK
* Correspondence: rameez.asif@uea.ac.uk

Abstract: This research paper reviews the potential of smart contracts for responsible AI with a focus on frameworks, hardware, energy efficiency, and cyberattacks. Smart contracts are digital agreements that are executed by a blockchain, and they have the potential to revolutionize the way we conduct business by increasing transparency and trust. When it comes to responsible AI systems, smart contracts can play a crucial role in ensuring that the terms and conditions of the contract are fair and transparent as well as that any automated decision-making is explainable and auditable. Furthermore, the energy consumption of blockchain networks has been a matter of concern; this article explores the energy efficiency element of smart contracts. Energy efficiency in smart contracts may be enhanced by the use of techniques such as off-chain processing and sharding. The study emphasises the need for careful auditing and testing of smart contract code in order to protect against cyberattacks along with the use of secure libraries and frameworks to lessen the likelihood of smart contract vulnerabilities.

Keywords: blockchain; smart contracts; security; privacy; artificial intelligence; cybersecurity; algorithms

check for updates

Citation: Asif, R.; Hassan, S.R.; Parr, G. Integrating a Blockchain-Based Governance Framework for Responsible AI. *Future Internet* **2023**, *15*, 97. https://doi.org/10.3390/fi15030097

Academic Editor: Eirini Eleni Tsiropoulou

Received: 23 January 2023
Revised: 19 February 2023
Accepted: 23 February 2023
Published: 28 February 2023

1. Introduction

Blockchain [1–3] and artificial intelligence (AI) [4–6] are two of the most significant and rapidly evolving technologies today. Blockchain is adaptable and scalable, whereas AI has overcome all of its limits in terms of testing performance in real-world applications. While both are rooted in contemporary technical breakthroughs, their origins are distinct [7]. Blockchain is thought to be a shared and permanent ledger that can be utilised for data encryption in the future. AI engines, on the other hand, allow a person to assess and make judgments based on acquired data [8–10]. It is worth noting that while each technology has several facets, the combination of artificial intelligence and blockchains can provide numerous benefits [11,12], as laid out in Figure 1.

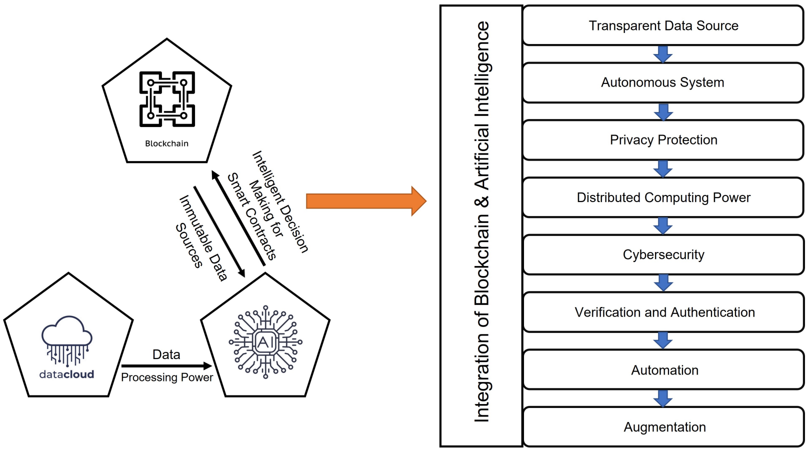
Blockchain
Intelligent Decision Making for Smart Contracts
Immutable Data Sources
datacloud
Data
Processing Power
AI
Integration of Blockchain & Artificial Intelligence
Transparent Data Source
Autonomous System
Privacy Protection
Distributed Computing Power
Cybersecurity
Verification and Authentication
Automation
Augmentation

Figure 1. The interaction of emerging technologies including AI, blockchain, and cloud.

1.1. Blockchain and Distributed Ledger Technology (DLT)

Blockchain is a decentralized digital ledger that records transactions on multiple computing nodes [13]. These transactions are grouped together in blocks, which are then chained together using cryptography [14]. This creates a permanent and unchangeable record of all transactions on the blockchain, as shown in Figure 2. The architecture of a blockchain network typically consists of nodes that participate in the network and a consensus mechanism that ensures the integrity of the transactions [15–18]. In a public blockchain network such as Bitcoin [19], anyone can participate as a node and the consensus mechanism is achieved through a process called mining [20]. In a private blockchain, the nodes are typically restricted and the consensus mechanism can be achieved through other means, such as voting [21]. Blockchain technology has the potential to revolutionize a wide range of industries by providing a secure and transparent way to record and track transactions [22]. Potential use cases include supply chain management, digital identity, and voting systems. Additionally, blockchain can be used to create decentralized applications (dApps) which are build on top of a blockchain network [23–25].

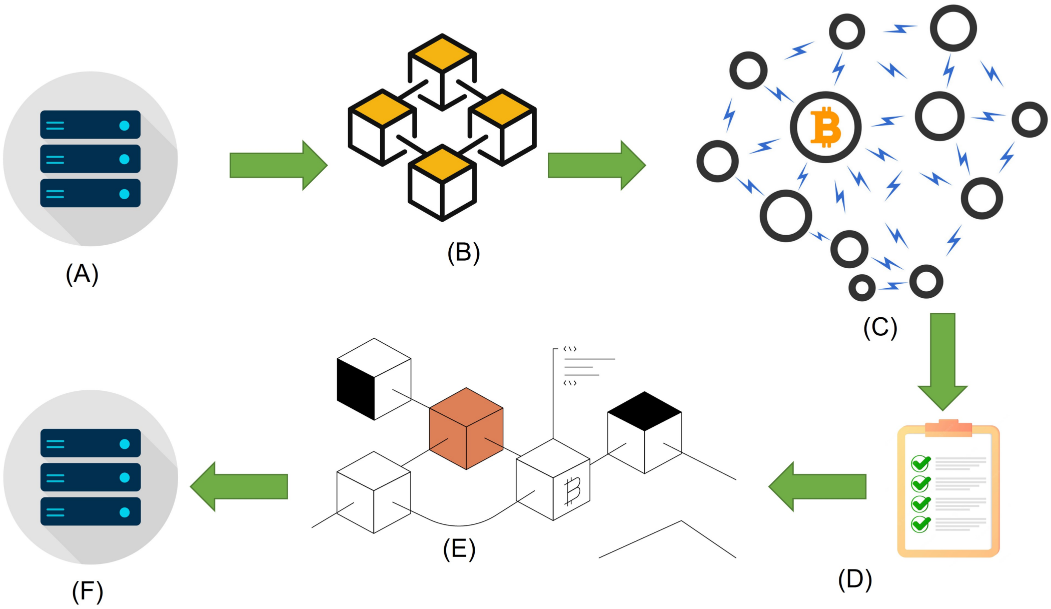

Figure 2. Flowchart of a blockchain network for transaction processing: (**A**) the transaction is initiated by the consumer, (**B**) the network represents the transaction as a block, (**C**) the verification of the block is broadcast to every node in the network, (**D**) validation of transactions is conducted by nodes, (**E**) the resulting block may then be added to the chain of blocks that serve as a transparent transaction ledger, (**F**) the intended consumer receives the transaction.

The complexity of blockchain technology depends on the specific implementation and use case [26]. For example, the complexity of running a full node on the Bitcoin network, which requires validating and relaying transactions, is relatively high and requires a significant amount of computational power [27]. In contrast, using a blockchain for a simple application such as tracking supply chain information would likely be less complex.

In terms of energy usage, the process of creating new blocks on a blockchain, known as mining, can be quite energy-intensive [28]. The most well-known example of this is the Bitcoin (BTC) network, which currently consumes a significant amount of electricity. This is because the process of mining new blocks involves solving complex mathematical problems, which requires a significant amount of computational power. However, newer blockchain protocols such as Ethereum are working on improving energy efficiency by using new consensus mechanism such as proof-of-stake, which is less energy intensive than proof-of-work (POW) [29,30]. It is important to note that energy usage is an issue that involves both the blockchain, the consensus mechanism, and the implementation of the blockchain protocol [31].

Blockchain technology is often considered to be highly secure and private due to its decentralized nature and the use of cryptography to secure transactions. One of the main security features of blockchain is that it is a distributed ledger, meaning that the information is stored on multiple computers in the network rather than in a central location [32]. This makes it much more difficult for hackers to tamper with or corrupt the data, as they would need to gain control of a majority of the nodes on the network. Additionally, after a block has been added to the blockchain it is very difficult to change or remove the information it contains [33].

When a smart contract is created [22,34], the terms of the agreement between the buyer and the seller are written into a computer program. This information is stored and replicated throughout a blockchain network [35,36]. Smart contracts enable the automatic management of digital assets, which can pave the way for the creation of decentralised apps that can be operated entirely by the network itself. Moreover, smart contracts, which are fundamental to blockchain technology, may be vulnerable to flaws, inefficiencies, or malicious malware [37]. It is important to consider the blockchain's underlying infrastructure security as well, such as the cloud where the nodes are hosted [38].

1.2. Responsible AI

Developing, deploying, and using AI systems in a manner that is ethical and useful to society is what is known as "responsible AI", and is a central idea in the study of artificial intelligence (AI) [39,40]. To make sure AI is utilised for the sake of humanity, we must carefully assess its ethical, legal, and societal consequences. Transparency is a fundamental concept of ethical artificial intelligence. Decision-making processes in AI systems need to be transparent in order to ensure that users and stakeholders can comprehend them [41]. This is especially crucial for decision-making systems in highly confidential fields such as medicine, banking, and law enforcement.

Fairness is another fundamental element of ethical AI. In order to achieve equality, it is essential that AI systems be built without bias [42,43]. This is especially crucial for systems used in fields such as employment, finance, and law enforcement [44]. Artificial intelligence systems in these domains should be developed to avoid contributing to the spread of prejudice and bigotry. Data preparation, data augmentation, and model validation are all strategies that may be used to make AI systems more equitable [45]. By employing these methods, it is possible to check whether the data used to train an AI system are indeed representative of the target population and that the model is not being over-fitted to the training data. Fairness in AI systems can be ensured by employing methods such as explainable AI (XAI) [46]. XAI is a collection of methods for increasing the transparency of AI systems, allowing users and other stakeholders to better grasp the reasoning behind the system's actions. This is especially crucial for decision-making systems in highly confidential fields such as medicine, banking, and law enforcement [47,48].

As AI systems become more integrated into society, it is crucial that they operate in an ethical and transparent manner. Smart contracts, which are self-executing agreements recorded on a blockchain, have the potential to play a significant role in promoting responsible AI. A taxonomy of the confluence of blockchain for responsible AI is provided in Figure 3. This paper reviews the ways in which smart contracts can establish a framework of rules and standards that ensure ethical and transparent use of AI systems. This includes defining clear limits on how AI can be used as well as creating a system of checks and balances that monitor key performance indicators such as accuracy, bias, and fairness. Smart contracts can provide an audit trail of AI transactions, which helps to ensure accountability and transparency throughout the life-cycle of AI systems. By promoting responsible AI, smart contracts can help to build trust in AI systems and mitigate potential risks associated with their use, ultimately paving the way for more widespread adoption of AI technology.

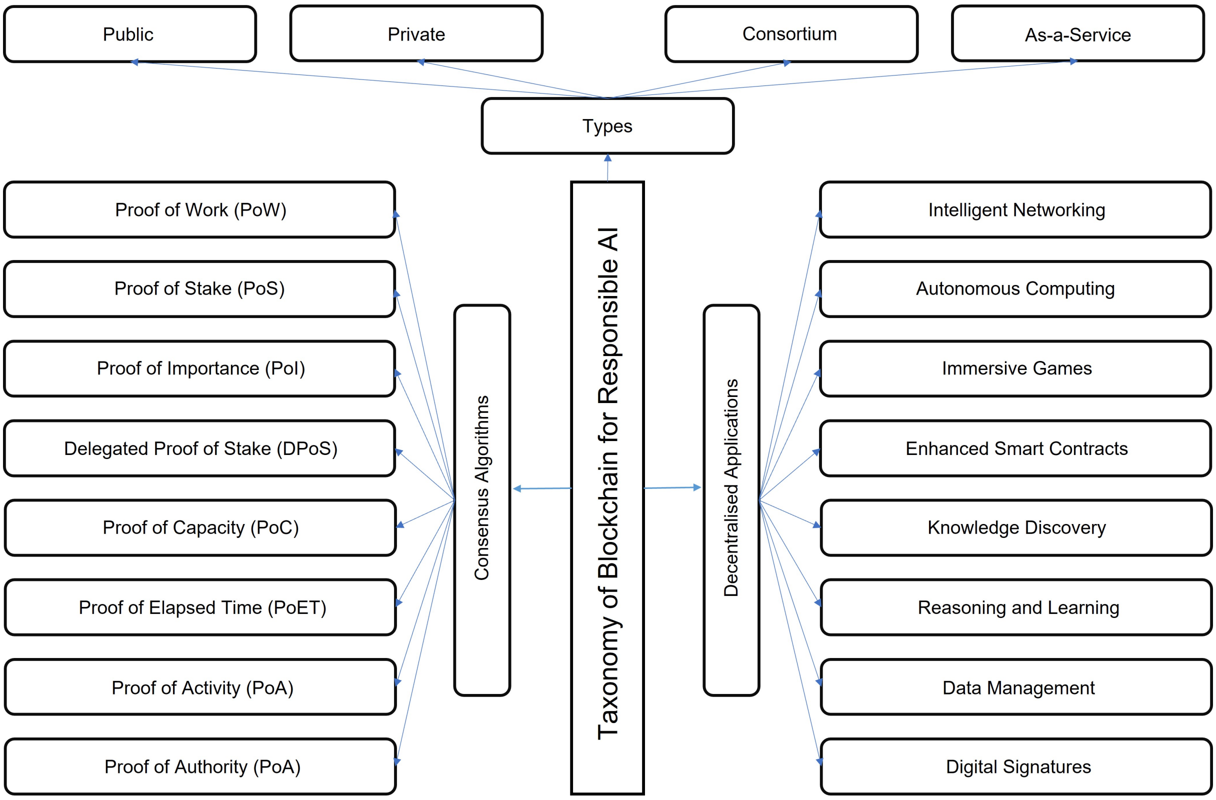

Figure 3. Taxonomy of blockchain for responsible AI.

The rest of this article is organized as follows. Section 2 covers algorithms, frameworks, and implementation tools for deploying smart contracts with AI. In this section, we analyse network system complexity and energy efficiency. Section 3 examines the verification and authentication systems used to deal with the overall convergence of smart contracts and AI. This section discusses the best practises in the smart contract review process that verify whether the contract performs as intended and does not include any bugs or vulnerabilities. Towards the mitigation of external cyber-threats, Section 4 discusses security and privacy issues, safe data management, and the implementation of explainability processes into the design of AI systems. The research in this article is conducted through a combination of a literature review and selected case studies.

2. Programming Smart Contracts for Responsible AI

Smart contracts can be used to enforce specific ethical or compliance-related norms around the development and deployment of AI systems in the context of responsible AI [49]. A smart contract, for example, might be used to verify that an AI system is only utilised for its intended purpose and that the system's decision-making processes are visible and auditable [50]. The following are some of the uses of adopting smart contracts for responsible AI:

- **Algorithmic Transparency:** Smart contracts can be used to ensure that the decision-making processes of an AI system are transparent and auditable [51]. For example, a smart contract can be used to automatically log all of the inputs and outputs of an AI system, as well as the specific decision-making rules that it uses [52].
- **Bias Mitigation:** Smart contracts can be used to ensure that an AI system is not biased against certain groups of people. For example, a smart contract could be used to automatically check that an AI system's training data are diverse and representative of different groups of people, and that the system's decision-making processes do not disproportionately harm certain groups [53].

- **Explainability:** Smart contracts can be used to ensure that the reasoning behind an AI system's decisions are explainable to human stakeholders [54]. For example, smart contracts can be used to automatically generate a human-readable explanation of the decision-making process used by an AI system [55,56].
- **Data Privacy:** Smart contracts can be used to ensure that an AI system's access to and use of sensitive data is controlled and compliant with regulations such as the GDPR [57]. For example, a smart contract could be used to automatically encrypt and decrypt data and to ensure that data are only used for specific pre-approved purposes [58].

It is worth noting that this is an area that is currently under active research and development [59]. There are many challenges to be addressed, such as the complexity of AI systems and the lack of clear regulations and standards for responsible AI [60]. Thus, although smart contracts can be a useful tool for enforcing certain aspects of responsible AI, it is important to keep in mind that they are not a panacea and there is a great deal of work to be done in order to ensure that AI systems are truly responsible and ethical.

2.1. Implementation Tools

Currently, there are a limited number of tools available for implementing smart contracts for responsible AI [61]. The availability of such tools depends on the specific use case and framework that is being implemented. However, there are a number of general-purpose smart contract platforms, such as Ethereum, EOS, and Hyperledger that can be used to build smart contracts for a variety of different use cases, including responsible AI [62].

Limited number of tools and platforms are available that have been developed specifically for implementing responsible AI using smart contracts [63]. For example, Ocean Protocol is a decentralized data exchange protocol that uses smart contracts to ensure that data are shared in a responsible and transparent way [64]. Algorand is another decentralized platform that provides smart contract capabilities and is designed to support scalable and secure execution of smart contracts [65]. Other generic off-the-shelf solutions include:

- **OpenLaw (https://www.openlaw.io/ (accessed on 22 January 2023)):** An open-source platform that allows users to create, manage, and execute smart legal agreements using natural language, it can be used to encode ethical guidelines for AI decision-making into smart contracts.
- **ChainGuard (https://www.chainguard.dev/ (accessed on 22 January 2023)):** A tool that uses formal verification to automatically check whether smart contracts meet specified safety and security properties. It can be used to ensure that AI decision-making smart contracts comply with ethical guidelines.
- **AI-Guard (https://www.revelis.eu/en/products/ai-guard-software-solution/ (accessed on 22 January 2023)):** A tool that uses formal methods to check whether AI models satisfy certain properties, such as fairness and robustness. It can be integrated with smart contract platforms to ensure that deployed AI models comply with ethical guidelines.
- **DeepTrust (https://www.deeptrustalliance.org/ (accessed on 22 January 2023)):** A blockchain-based platform that enables explainable AI and trustworthy machine learning models, DeepTrust allows for automated compliance checking and dispute resolution through smart contracts.
- **AI Ethics Lab (https://www.aiethicslab.com/ (accessed on 22 January 2023)):** A platform that provides a framework for creating, testing and deploying AI systems that align with ethical principles. It provides a library of pre-defined ethical guidelines and can be used to automatically generate smart contracts that enforce these guidelines.

These are only a few examples of the tools that can be used to automatically program smart contracts for responsible AI. It is important to note that these tools may not be suitable for every use case, and that the appropriateness of a tool depends on the specific requirements of the AI system in question. While there are currently no specific standards and regulations specifically for smart contracts in responsible AI, there are general guidelines and best practices that organizations can follow when developing and deploying AI systems. One framework that organizations can follow is the IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems [66], which provides a set of ethical principles for the development and use of AI systems. Another is the Asilomar AI Principles [67], which provide a set of guidelines for the responsible development and use of AI.

In terms of regulations, the General Data Protection Regulation (GDPR) in the European Union includes provisions related to AI and automated decision-making, such as the right to explanation and the right to human intervention [68]. Many other jurisdictions are currently considering or developing regulations specifically for AI. In addition, certain organizations and industries have their own guidelines and regulations for AI; for example, the Federal Reserve, the National Institute of Standards and Technology (NIST), and the Center for Internet Security (CIS) have all issued guidelines for responsible AI use in the financial services sector. It is important to note that regulations and guidelines are evolving, and organizations should keep themselves updated with the latest developments in the field.

2.2. Algorithms and Frameworks

There are several algorithms [69] that can be used to prevent bias and ensure accountability via smart contracts, including:

1. **Fairness Algorithms:** These algorithms can be used to detect and correct bias in AI models by identifying and adjusting for factors that may be causing discrimination. For example, algorithms such as the Zafar–Valera–Gummadi (ZVG) algorithm [70] or the Prejudice Remover algorithm can be used to identify and remove bias in training data [71].
2. **Explainable AI (XAI) Algorithms:** These algorithms can be used to provide transparency and accountability for AI decision-making by generating human-understandable explanations for the decisions made by AI models [72]. For example, the Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) algorithms can be used to generate explanations for individual predictions made by a model [73].
3. **Auditing Algorithms:** These algorithms can be used to monitor and evaluate the performance of AI models over time, including detecting and reporting any issues such as bias or drift. For example, the Fairness and Accountability Audit (FAA) algorithm [74] can be used to evaluate the performance of an AI model with respect to a set of fairness metrics.
4. **Decentralized AI Algorithms:** These algorithms can be used to enable decentralized decision-making for AI systems and ensure the distribution of power between stakeholders; for example, the Federated Learning and Multi-Party Computation (MPC) algorithms can be used for decentralized training of AI models, and blockchain-based consensus algorithms can be used for decentralized decision-making [75,76].

2.3. System Integration and Complexity

The aforementioned algorithms can be integrated into smart contracts and automatically enforced to prevent bias and ensure accountability in AI decision-making. However, it is important to note that these algorithms are not silver bullets and may not be suitable for every use case. The mostappropriate algorithm depends on the specific requirements of the AI system in question.

The real-world integration of these algorithms depends on their flexibility, scalability, and seamless integration with the existing hardware [77]. The hardware complexity is dependent on the specific algorithm and the size of the dataset being processed. Fairness algorithms can be computationally intensive, particularly when working with large datasets. For example, the ZVG algorithm [70] and the Prejudice Remover algorithm [71] both require multiple passes over the dataset, which can be time-consuming. However, these algorithms can be implemented on standard hardware and can be optimized for distributed computing environments. XAI algorithms can be computationally intensive, particularly when working with large and complex models. For example, although the LIME and SHAP algorithms can be computationally expensive [73], they can be optimized for distributed computing environments.

Auditing algorithms can be computationally less intensive than LIME and SHAP, as they require monitoring and evaluating the performance of an AI model over time. The FAA algorithm can be run periodically, and does not need to be run on real-time data. Decentralized AI algorithms can be computationally intensive, particularly when working with large datasets and complex models. While the Federated Learning and MPC algorithms can require significant computational resources, they can be optimized for distributed computing environments. In general, the hardware complexity of these algorithms can be managed by using distributed computing (sharding) and cloud-based resources (off-chain processing) [78]. Additionally, hardware acceleration technologies such as GPUs can be used to speed up the processing time of certain algorithms.

One of the main benefits of using a distributed network for smart contract execution is increased security. By distributing the contract across multiple nodes, it becomes much more difficult for a malicious actor to tamper with or disrupt the execution of the contract. Additionally, distributed networks are often more resistant to outages and other forms of downtime, ensuring that the contract continues to execute as expected [79]. Another benefit of distributed computing is transparency. Smart contracts are stored on a public blockchain, which allows for anyone to view and verify the terms and execution of the contract. This transparency can help to build trust and confidence in the contract, as all parties can see that it is being executed as intended [80].

Another benefit of using cloud-based resources is scalability. Cloud providers offer a wide range of computing and storage resources that can be easily accessed and scaled, allowing for the efficient execution of smart contracts and the ability to handle large amounts of data [81]. This is particularly useful for decentralized applications that are expected to handle a large number of transactions or data. Finally, cloud-based resources can provide cost savings for smart contract deployment and execution. Instead of having to invest in and maintain their own infrastructure, organizations can use the resources provided by a cloud provider on a pay-as-you-go basis, which can be much more cost-effective [82–84].

In summary, distributed computing and cloud-based resources can benefit smart contracts by providing increased security, transparency, scalability, and cost savings for their deployment and execution.

2.4. Network-Driven Energy Efficiency

Energy efficiency is dependent on the frameworks and the hardware on which it is integrated [85]. In general, the energy efficiency of an algorithm is related to the amount of computational power required to run it [86]. Distributed computing and cloud-based resources can be used to manage the hardware complexity of these algorithms in several ways, such as the following:

- Scalability: Distributed computing allows for the distribution of computations across multiple machines, which can significantly increase the overall computational power available. This can be especially useful when working with large data sets or complex models. Cloud-based resources provide access to a large pool of computational resources that can be easily scaled up or down as needed.
- High Availability: Distributed computing and cloud-based resources can provide high availability of computational resources, ensuring that algorithms can be run continuously without interruption. This can be especially important for real-time or time-critical applications.
- Resource Optimization: Distributed computing and cloud-based resources can be used to optimize the use of resources. For example, it is possible to run algorithms

on specialized hardware such as GPUs or TPUs, which can significantly speed up processing times.

- Cost-effectiveness: Using cloud-based resources can be cost-effective, as it eliminates the need to invest in and maintain expensive hardware. Additionally, cloud providers typically offer pay-as-you-go pricing models, which can help organizations to manage costs by only paying for the resources they use.
- Easy Access: Cloud-based resources can be easily accessed from anywhere with an internet connection, making it easy for organizations to collaborate and share resources. Additionally, many cloud providers offer pre-configured machine images with popular deep learning libraries pre-installed, making it easy to set up and run algorithms.

It is important to note that while distributed computing and cloud-based resources can greatly increase the overall computational power available, they may not be suitable for every use case, and their appropriateness depends on the specific requirements of the AI system in question [87]. Additionally, the security of data and the cost of using these resources have to be considered.

2.5. Technological Issues of Smart Contracts

Smart contract implementation entails a variety of technological issues that are currently being solved and improved. Scalability is the most significant bottleneck. The network can become crowded and sluggish as more transactions are added to a blockchain, resulting in high fees and long processing times. Smart contracts can be incorporated throughout chains to mitigate this issue (multiple platforms). There are now several blockchain systems, each with its own set of unique capabilities and restrictions [22,88]. As a result, interoperability solutions that enable smart contracts to communicate and function across multiple blockchains are required [89].

Another significant integration difficulty is that smart contracts are self-executing and irreversible, which means that any faults or vulnerabilities in the code might have catastrophic effects [90]. As a result, comprehensive security testing and audits are required to verify that smart contracts are secure and free of vulnerabilities. Furthermore, while smart contracts have several advantages, they might be difficult to use and comprehend for non-technical people. As a result, improved user interfaces and tools that make it easier for individuals to build, implement, and interact with smart contracts are required [91]. Recently, there has been a lot of discussion about decentralisation and privacy [92]. Smart contracts are often maintained on a public blockchain, which means that the contract's terms are available to everyone on the network. There is a demand for privacy solutions that enable smart contracts to be executed more privately and securely.

In addition to the major technical challenges mentioned earlier, there are a number of minor challenges that can arise when implementing smart contracts. Code complexity is one of these. Smart contracts can be complex, with many lines of code and complex logic. This can make it difficult to debug and test the code, and can lead to errors that are difficult to spot. Integrating smart contracts with existing systems can be another challenge, particularly if those systems are based on legacy technologies [93]. This can require additional development work and may result in interoperability issues. Moreover, smart contracts may need to comply with legal and regulatory requirements, such as data protection laws or financial regulations. Ensuring compliance can be a challenge, particularly as the regulatory landscape is constantly evolving [94]. After a smart contract has been deployed on a blockchain, it can be difficult to make changes to the code. Upgrading a smart contract can require coordination with other parties, and can result in downtime or other disruptions.

While these challenges may be considered minor compared to the major technical challenges, they can have a significant impact on the implementation of smart contracts. Addressing these challenges requires careful planning, collaboration, and ongoing development efforts.

2.6. Use Cases of Smart Contracts for Responsible AI

Possible use cases may vary depending on the specific algorithms in question; however, they can generally be used in a variety of industries and applications to promote responsible AI. Fairness algorithms can be used to detect and correct bias in AI models, and can be applied in a variety of contexts such as healthcare, finance, and criminal justice [95]. For example, they can be used to ensure that medical diagnosis AI systems are not biased against certain groups of patients, or that a credit scoring AI system is not biased against certain groups of people [96]. There are several potential real-world use cases for smart contracts in the field of responsible AI. We discuss few of the examples below:

- Decentralized Machine Learning: Smart contracts can be used to create decentralized machine learning marketplaces, where participants can buy and sell access to machine learning models. The smart contracts can be used to govern access to the models, as well as the payment and settlement of transactions.
- Data Sharing and Ownership: Smart contracts can be used to govern the sharing and ownership of data in AI systems. For example, a smart contract could be used to determine who owns the data used to train a machine learning model and how the data can be used and shared.
- Autonomous Organizations: Smart contracts can be used to create autonomous organizations that are governed by code rather than humans. These organizations could use AI to make decisions and take actions, with the smart contracts governing the rules and processes by which decisions are made.
- Intellectual Property Management: Smart contracts can be used to manage the intellectual property associated with AI systems, such as patents and copyrights. Smart contracts can be used to govern the licensing and use of the IP as well as the distribution of any revenues generated from it.
- Supply Chain Management: Smart contracts can be used to manage the supply chain in industries that rely on AI, such as autonomous vehicles or drones. Smart contracts can be used to govern the movement of goods and services, as well as the payment and settlement of transactions.

As the field of AI continues to evolve, we are likely to see more real-world applications of smart contracts in this area. XAI algorithms can be used to provide transparency and accountability for AI decision-making, and can be applied in a variety of contexts such as finance, healthcare, and autonomous systems [97]. For example, they can be used to explain the decisions made by a fraud detection AI system in a bank, or the decisions made by a medical diagnosis AI system in a hospital. Auditing algorithms can be used to monitor and evaluate the performance of AI models over time and can be applied in a variety of contexts, such as finance, healthcare, and autonomous systems [98]. For example, they can be used to detect and report any issues such as bias or drift in a self-driving car AI system. Decentralized AI algorithms can be used to enable decentralized decision-making for AI systems, and can be applied in a variety of contexts such as finance, healthcare, and autonomous systems. For example, they can be used to enable the collaborative training of AI models in a decentralized network of hospitals, or the decentralized decision-making of an autonomous vehicle fleet.

Below, Listing 1, is an example of a simple smart contract for responsible AI using Solidity (Solidity is a high-level, object-oriented language for implementing smart contracts. Solidity smart contracts are programmes that regulate the behaviour of Ethereum accounts. (https://docs.soliditylang.org/en/v0.8.17/ (accessed on 22 January 2023), Programming Language and the Ethereum Blockchain).

Listing 1. Use-Case: Smart Contract for Responsible AI.

```
pragma solidity ^0.8.0;

contract ResponsibleAI {

    // Event to log when the AI system makes a decision
    event DecisionMade(string decision);

    // Function to make a decision
    function makeDecision(string input) public {
        // Code to run the AI system's decision-making process
        // ...
        // Emit event to log the decision
        emit DecisionMade("The AI system has decided to: " + decision);
    }

    // Function to view the AI system's decision-making process
    function viewDecisionProcess() public view returns (string memory) {
        // Code to return the AI system's decision-making process
        // ...
    }
}
```

This smart contract example defines a contract called “ResponsibleAI” that has two functions: “makeDecision” and “viewDecisionProcess”. The “makeDecision” function is used to trigger the AI system’s decision-making process, and the “viewDecisionProcess” function can be used to view the AI system’s decision-making process. Additionally, the contract has an event called “DecisionMade” which is emitted every time the AI system makes a decision.

Let us say we have an AI system that is used to make loan decisions for a financial institution. The AI system takes in various inputs, such as the borrower’s credit score, income, and employment history, then makes a decision about whether to approve or deny the loan. The financial institution wants to ensure that the AI system is making fair and unbiased decisions, and that its decision-making process is transparent and auditable. To do this, the institution decides to use the “ResponsibleAI” smart contract to enforce certain responsible AI practices.

1. The institution deploys the “ResponsibleAI” smart contract on the Ethereum blockchain.
2. The institution connects the AI system to the smart contract so that it can interact with the contract’s functions.
3. Every time the AI system makes a loan decision, it calls the “makeDecision” function of the smart contract, passing on the relevant inputs and the decision that the system has made.
4. The “makeDecision” function of the smart contract automatically emits the “DecisionMade” event, which logs the decision and the relevant inputs in the blockchain’s public ledger.
5. The financial institution can use the “viewDecisionProcess” function of the smart contract to view the AI system’s decision-making process and ensure that it is transparent and auditable.
6. The financial institution can use the data logged in the blockchain to conduct bias analysis on the AI system’s decision-making process to ensure that it is fair and unbiased.

3. System Architecture and Information Flow

The use case in the previous section illustrates how smart contracts can be used to enforce certain responsible AI practices and make AI systems’ decision-making processes more transparent and auditable, as depicted in Figure 4. It should be noted that this is a simplified example; a real-life use case would be more complex, and would require additional measures such as access control, encryption, and compliance with regulations such as the GDPR.

The conventional method consists of aggregating data from several source systems, cleaning, and reconciling errors, and then loading the data into a single business data warehouse [99]. The scope of an institution's central data repository may vary. Individual apps that own the business process continue to be the holy grail of authenticity. Many organisations have several data warehouses (e.g., for marketing and compliance), whereas others develop specialised data marts from a core warehouse [100]. Consequently, tracing a data discrepancy back to the system that owns that data is, to say the least, a complicated endeavour.

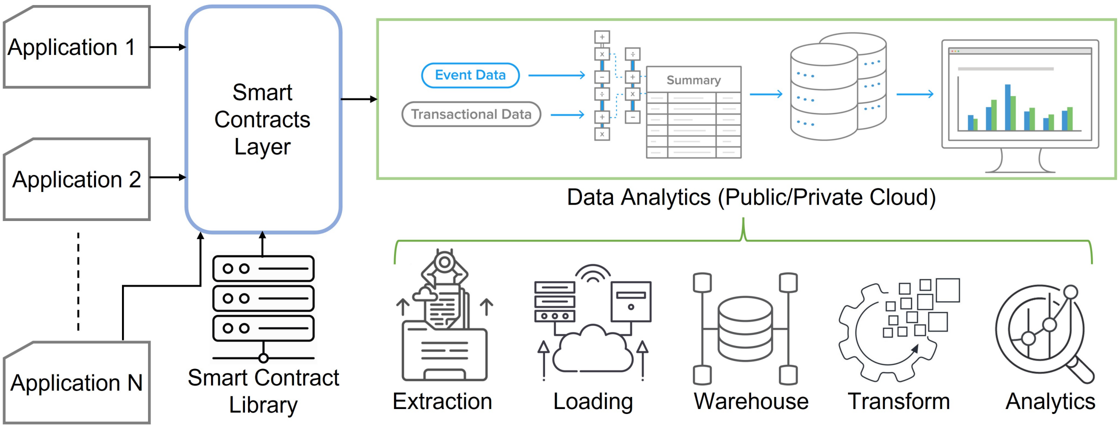

Figure 4. Flowchart of a smart contract-based Responsible AI setup with ETL for business intelligence and analytics (ETL: Extract, Transform, and Load).

Integrating smart contracts with the information flow, as depicted in Figure 4, brings two fundamental changes:

1. First, all business actions result in predictable changes to the underlying data. This means that we may now see important entities as "digital assets" on which actions are performed. A credit card provided to a consumer, for instance, is a digital asset on which a fraud flag is raised, a payment is made, a campaign is conducted, etc.
2. Second, instead of being concealed in various flowcharts and paperwork, a company's whole business process is now embodied in tangible form. Any corporate IT professional is aware that documentation becomes obsolete as soon as it is created. With smart contracts, your business process is simply codified into functional software, which is then maintained and managed in accordance with the organisational hierarchy.

The smart contracts store additionally provides a single source of truth for the applications landscape built on top of smart contracts. These data are produced in real time by operational procedures, meaning that there is no need to perform any post-execution reconciliation or aggregation. It is inevitable that considerable effort is invested on data staging, reconciliation, and aggregation, i.e., ELT (Extract, Load, Transform). The larger the organisation, the greater the number of source systems and application silos, and the more complex the total data management architecture. The complexity of integrating new services presents obstacles for corporate innovation.

Data warehousing can be integrated with smart contracts via either the ETL (Extract, Transform, Load) or ELT (Extract, Load, Transform) approach [101]. In traditional data warehousing, ETL is used to process and store data from various sources into a centralized location for reporting and analysis. In this process, data are extracted from various sources, transformed to fit the scheme of the data warehouse, and then loaded into the warehouse for storage [102]. In contrast, ELT is a variation of ETL in which data are first loaded into the data warehouse in raw form and then transformed. This allows for the data warehouse to be more flexible, as the data can be transformed on-the-fly as needed, rather than requiring preprocessing prior to loading. This comparison between ETL and ELT is summarised in Table 1.

ELT can be beneficial in the smart contract context, as the data stored in a smart contract are immutable and cannot be altered. This means that data must be transformed before being stored in the smart contract; using ELT allows for data to be transformed on-the-fly as they are loaded into the smart contract. This can lead to more efficient and cost-effective data processing [103], in turn resulting in lower energy consumption and cost, as less computation is required to transform the data before storing in the smart contract. Additionally, ELT can reduce the complexity of the data pipeline and the need to maintain a separate data warehousing infrastructure, which can lead to lower energy consumption and cost.

Table 1. Summary of the characteristics of ETL and ELT for integration with smart contracts.

Characteristics	ETL	ELT
Process Execution	Data is collected, moved and transformed at the staging layer. It is then transferred to the target server	Data is collected and moved to the target server where the transformation will take place
Maintenance	Expensive maintenance and technical knowledge required	Virtually cost-effective as we move raw data
Processing Time	Significantly high because of the steps involved	Efficient as it is least dependent on the transformation
Infrastructure	Hybrid or on-premise environment that is difficult to scale as per the data size	Cloud based infrastructure as SaaS and PaaS having dynamic scalability
Costs	High initial and running cost	Low start-up cost and can be adjusted as per the volume of data

Verification and Authentication of Smart Contracts

The verification process for smart contracts for responsible AI involves a combination of technical and non-technical checks to ensure that the contract functions as intended and aligns with ethical and responsible AI principles [104]. One of the key technical checks is code review, where the contract's code is examined line by line to ensure that it functions as intended and does not contain any errors or vulnerabilities that could be exploited [105]. This can be done manually by human experts or through the use of automated tools.

Another important check is testing, where the contract is run through a series of tests to ensure that it behaves as expected in different scenarios. This can include testing the contract's performance, security, and compliance with relevant regulations [106]. In addition to technical checks, there are non-technical checks that must be performed to ensure that the contract is aligned with ethical and responsible AI principles. This may include a review of the contract's impact on privacy, fairness, and accountability [107]. Additionally, it could include the contract's alignment with the organization's values and policies.

After all the checks have been completed and any necessary changes have been made, the contract can be deployed and executed on a blockchain network. It is worth noting that the smart contract verification process is an active area of research and development, and new methods and tools are continually being developed to improve the process.

There are several best practices that can be followed during code review for smart contracts to ensure that the contract functions as intended and does not contain any errors or vulnerabilities.

- Understand the Business Requirements: Before beginning the code review, it is important to have a clear understanding of the business requirements and the intended use case of the contract. This can help to ensure that the review is focused on the relevant issues.
- Use Automated Tools: Automated tools can be used to help identify potential errors and vulnerabilities in the code. This can include static analysis tools, which analyze the code without executing it, and dynamic analysis tools, which execute the code and monitor its behavior.

- Review the Code Manually: While automated tools can be useful, they cannot replace manual code review. It is important to have human experts review the code to ensure that it is clear and readable and that it meets the business requirements.
- Pay Attention to Security: Smart contracts are particularly sensitive to security issues, and it is important to pay special attention to potential vulnerabilities such as re-entrancy, integer overflow, and underflow, and denial of service.
- Check for Compliance: Make sure that the smart contract is compliant with any relevant regulations, industry standards, and best practices.
- Test the Code: After the code has been reviewed and any necessary changes have been made, it is important to test the contract to ensure that it behaves as expected in different scenarios.
- Keep Track of the Changes: It is important to keep track of all the changes made during the code review process to ensure that any issues can be traced back to their source.
- Continuously Monitor: Smart contracts are deployed on the blockchain and are meant to run for a long time, so it is important to continuously monitor the contract for any issues, vulnerabilities, or bugs.

Following these best practices during code review can help to ensure that the smart contract functions as intended, is secure and compliant with relevant regulations, and is aligned with ethical and responsible AI principles. Examples of the industries that are actively working to bring about smart contracts for responsible AI are listed in Table 2.

Table 2. Industrial solutions using smart contracts that could potentially be applied to responsible AI and dApp development.

Smart Contract Industry	Explanation	Functions	References
Ocean Protocol	Decentralized data exchange protocol that uses smart contracts to ensure that data is shared in a responsible and transparent way.	Data Sharing & Cloud	[64]
Algorand	Decentralized platform that provides smart contract capabilities and is designed to support scalable and secure execution of smart contracts.	Security & Scalability	[108,109]
SingularityNET (https://singularitynet.io/ accessed on 22 January 2023)	Decentralized marketplace for AI services that uses smart contracts to ensure that AI services are secure, transparent and compliant with regulations.	Transparency & Compliance	[110]
ChainGuard	Uses smart contracts to ensure that AI algorithms are transparent and auditable.	Audit & Supply Chain	[111,112]

4. Security and Privacy Issues

Security and privacy concerns [113], as shown in Figure 5, must be taken into account across the whole AI life-cycle (from concept to development to deployment to maintenance) if responsible AI is to be created and used [114]. All data must be collected, used, and shared in accordance with applicable laws and regulations, and appropriate technological and organisational measures must be in place to prevent unauthorised access, use, and disclosure [115]. It is crucial to ensure that people have agency with respect to their personal data and that there is openness around the data and algorithms utilised in AI systems.

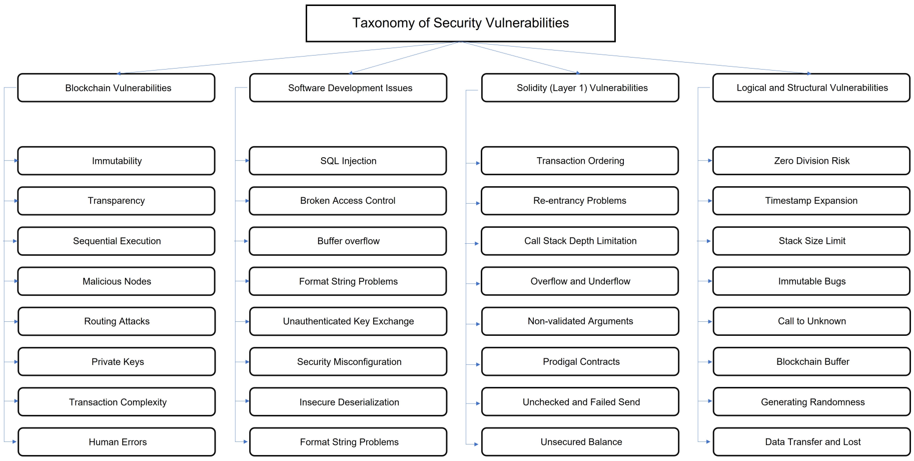

Figure 5. Taxonomy of smart contract security vulnerabilities.

4.1. Design

During the design phase, security and privacy considerations should be integrated into the overall design of the AI system. This includes determining which data are needed to train and operate the system as well as identifying any potential security or privacy risks associated with the data. Additionally, it is important to consider how the system will be used, who will have access to it, and what types of data it needs to handle in order to develop appropriate security and privacy controls.

4.2. Development

During the development phase, security and privacy controls should be implemented to protect the AI system and the data it uses. This includes implementing technical measures such as encryption, access controls, and intrusion detection to protect against unauthorized access and cyberattacks. Additionally, it is important to ensure that the AI system is developed in compliance with relevant laws and regulations, such as data protection regulations, that may impact security and privacy.

4.3. Deployment

During deployment, it is important to test the AI system to ensure that security and privacy controls are working as intended. This includes conducting vulnerability assessments, penetration testing, and other types of security testing to identify and address any potential vulnerabilities. Additionally, it is important to ensure that the AI system is configured and deployed in a way that meets the organization's security and privacy requirements.

4.4. Maintenance

After deployment, the AI system must be continuously maintained to ensure that security and privacy controls remain effective. This includes monitoring the system for any signs of unauthorized access or cyberattacks, as well as updating security controls and patches to address any new vulnerabilities that may be discovered. Additionally, it is important to regularly review and assess the system's compliance with relevant laws and regulations to ensure that it continues to meet the organization's security and privacy requirements.

Overall, security and privacy considerations should be integrated throughout the entire AI development process, from design to maintenance [116]. This requires collaboration between different teams, such as data scientists, engineers, security experts, and legal professionals, to ensure that the AI system is developed and deployed in a responsible way [117]. Despite such collaborative efforts, there is always the possibility of a cyberattack, as smart contracts on the blockchain are stored on a decentralized network and are visible to all users. While this transparency is one of the key benefits of blockchain technology, it makes smart contracts vulnerable to attacks [118]. We have summarised the prominent cyberattacks targeting smart contract applications in Table 3.

Table 3. Summary of different cyberattacks targeting smart contract applications.

Type of Attack	Brief Description	Reference
Re-entrancy Attack	A malicious contract calls another contract multiple times before the first contract's state is updated, allowing the attacker to drain the contract's funds multiple times before the attack is detected.	[119,120]
Front-Running Attack	An attacker can see and act on information in a smart contract before it is visible to other users, allowing the attacker to profit at the expense of other traders.	[121,122]
DoS (Denial of Service) Attack	An attacker overloads a smart contract with a large number of requests, causing it to malfunction or crash, preventing legitimate users from accessing the contract and leading to financial losses.	[123,124]
Oracle Attack	An attacker manipulates external data (oracles) to execute malicious actions within the smart contract.	[125,126]

To counter these attacks [127], it is important for developers to take the following steps:

- Thoroughly test and audit smart contracts before deployment.
- Test for reentrancy and front-running vulnerabilities.
- Implement measures to protect against DoS attacks.
- Use secure oracle providers.
- Constantly monitor the contract for any suspicious activity.

Other cyberattacks thatcan affect the efficiency of smart contracts [128] where responsible AI is the main application include:

1. Adversarial attacks on machine learning models, where an attacker deliberately manipulates input data in order to cause the model to make incorrect decisions.
2. Data poisoning attacks, where an attacker alters the training data used to train a machine learning model, causing it to make incorrect predictions.
3. Explainability attacks, where an attacker manipulates the output of an AI system in a way that makes it difficult or impossible to understand how a decision was made.

Secure data management, routine testing, monitoring of AI systems, and the incorporation of explainability procedures into the architecture of AI systems are all significant ways to diminish these vulnerabilities. To guarantee that an AI is acting in an ethical and responsible manner, it is imperative to have clear governance and oversight mechanisms in place, such as frequent audits and reviews [129].

5. Conclusions

In this review, we have explored the potential of smart contracts for responsible AI. Smart contracts have been proven to be an effective tool for safeguarding the responsible and ethical growth, implementation, and use of AI systems. Smart contracts can improve confidence in AI systems and reduce the likelihood of prejudice, discrimination, and other unexpected effects by automating compliance with ethical and legal standards, thereby providing greater transparency and accountability. While smart contracts may seem like a holy grail, it is crucial to remember that they are not without their challenges, despite the fact that their potential benefits are significant.

Future studies should look into implementing sophisticated cryptographic techniques that can secure data stored on smart contracts without increasing costs. Because using a blockchain to store data comes at a very high cost, hybrid solutions must be developed in order to make use of the efficient and private access and storage afforded by external data repositories as well as the traceability of data transactions provided by blockchain networks. Meanwhile, the likelihood of smart contract vulnerabilities may be reduced via comprehensive auditing and testing of the smart contract's code and by making use of secure libraries and frameworks. In summary, smart contract applications in the context of responsible AI shows great potential and is a promising topic for research. Smart contracts have the ability to ensure that AI systems are designed and implemented in a way that is fair, transparent, and responsible to all stakeholders, which makes them a key area of research as the field of AI continues to proliferate.

Author Contributions: Conceptualization, R.A., S.R.H. and G.P.; resources, R.A. and S.R.H.; data curation, R.A.; writing original draft preparation, R.A.; review and editing, R.A., S.R.H. and G.P.; supervision, G.P.; project administration, R.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to acknowledge the contributions of Satoshi Nakamoto and Vitalik Buterin for developing the blockchain concept.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
BTC	Bitcoin
CIS	Center for Internet Security
CS	Cybersecurity
DApps	Decentralized Applications
DDOS	Distributed Denial-of-Service
DLT	Distributed Ledger Technology
ETH	Ethereum
ELT	Extract, Load, Transform
ETL	Extract, Transform, Load
FAA	Fairness and Accountability Audit
GDPR	General Data Protection Regulation
IoT	Internet-of-Things
IIoT	Industrial Internet-of-Things
LIME	Local Interpretable Model-agnostic Explanations Algorithms
MPC	Multi-Party Computation
NIST	National Institute of Standards and Technology
POW	Proof-of-Work
POS	Proof-of-Stake
SHAP	SHapley Additive Explanations Algorithms
TPS	Transactions Per Second
SegWit	Segregated Witness
XAI	Explainable AI
ZVG	Zafar–Valera–Gummadi Algorithm

References

1. Ismail, L.; Materwala, H. A Review of Blockchain Architecture and Consensus Protocols: Use Cases, Challenges, and Solutions. *Symmetry* **2019**, *11*, 1198. [CrossRef]
2. Ali Syed, T.; Alzahrani, A.; Jan, S.; Siddiqui, M.S.; Nadeem, A.; Alghamdi, T. A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recommendations. *IEEE Access* **2019**, *7*, 176838–176869. [CrossRef]
3. Lao, L.; Li, Z.; Hou, S.; Xiao, B.; Guo, S.; Yang, Y. A Survey of IoT Applications in Blockchain Systems: Architecture, Consensus, and Traffic Modeling. *ACM Comput. Surv.* **2020**, *53*, 18. [CrossRef]
4. Haefner, N.; Wincent, J.; Parida, V.; Gassmann, O. Artificial intelligence and innovation management: A review, framework, and research agenda. *Technol. Forecast. Soc. Chang.* **2021**, *162*, 120392. [CrossRef]
5. Ahmad, T.; Zhang, D.; Huang, C.; Zhang, H.; Dai, N.; Song, Y.; Chen, H. Artificial intelligence in sustainable energy industry: Status Quo, challenges and opportunities. *J. Clean. Prod.* **2021**, *289*, 125834. [CrossRef]
6. Omitaomu, O.A.; Niu, H. Artificial Intelligence Techniques in Smart Grid: A Survey. *Smart Cities* **2021**, *4*, 548–568. [CrossRef]

7. Wang, K.; Dong, J.; Wang, Y.; Yin, H. Securing Data with Blockchain and AI. *IEEE Access* **2019**, *7*, 77981–77989. [CrossRef]
8. Chen, L.; Chen, P.; Lin, Z. Artificial Intelligence in Education: A Review. *IEEE Access* **2020**, *8*, 75264–75278. [CrossRef]
9. Haenlein, M.; Kaplan, A. A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence. *Calif. Manag. Rev.* **2019**, *61*, 5–14. [CrossRef]
10. Nishant, R.; Kennedy, M.; Corbett, J. Artificial intelligence for sustainability: Challenges, opportunities, and a research agenda. *Int. J. Inf. Manag.* **2020**, *53*, 102104. [CrossRef]
11. Salah, K.; Rehman, M.H.U.; Nizamuddin, N.; Al-Fuqaha, A. Blockchain for AI: Review and Open Research Challenges. *IEEE Access* **2019**, *7*, 10127–10149. [CrossRef]
12. Imran, M.; Yao, B.; Ali, W.; Akhunzada, A.; Azhar, M.K.; Junaid, M.; Iqbal, U. Research Perspectives and Challenges of Blockchain for Data-Intensive and Resource-Constrained Devices. *IEEE Access* **2022**, *10*, 38104–38122. [CrossRef]
13. Yang, W.; Aghasian, E.; Garg, S.; Herbert, D.; Disiuta, L.; Kang, B. A Survey on Blockchain-Based Internet Service Architecture: Requirements, Challenges, Trends, and Future. *IEEE Access* **2019**, *7*, 75845–75872. [CrossRef]
14. Seok, B.; Park, J.; Park, J.H. A Lightweight Hash-Based Blockchain Architecture for Industrial IoT. *Appl. Sci.* **2019**, *9*, 3740. [CrossRef]
15. Saghiri, A.M. Blockchain Architecture. In *Advanced Applications of Blockchain Technology*; Kim, S., Deka, G.C., Eds.; Springer: Singapore, 2020; pp. 161–176. [CrossRef]
16. Hakak, S.; Khan, W.Z.; Gilkar, G.A.; Imran, M.; Guizani, N. Securing Smart Cities through Blockchain Technology: Architecture, Requirements, and Challenges. *IEEE Netw.* **2020**, *34*, 8–14. [CrossRef]
17. Latif, S.; Idrees, Z.; Ahmad, J.; Zheng, L.; Zou, Z. A blockchain-based architecture for secure and trustworthy operations in the industrial Internet of Things. *J. Ind. Inf. Integr.* **2021**, *21*, 100190. [CrossRef]
18. Lee, J.; Azamfar, M.; Singh, J. A blockchain enabled Cyber-Physical System architecture for Industry 4.0 manufacturing systems. *Manuf. Lett.* **2019**, *20*, 34–39. [CrossRef]
19. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 15 January 2023).
20. Buterin, V. The Blockchain: A Comparison of Platforms and Their Uses. *Crypto Brief.* **2014.** Available online: https://ethereum.org/en/foundation/ (accessed on 15 January 2023).
21. Febrero, P.; Pereira, J. Cryptocurrency Constellations Across the Three-Dimensional Space: Governance Decentralization, Security, and Scalability. *IEEE Trans. Eng. Manag.* **2022**, *69*, 3127–3138. [CrossRef]
22. Wang, S.; Ouyang, L.; Yuan, Y.; Ni, X.; Han, X.; Wang, F.Y. Blockchain-Enabled Smart Contracts: Architecture, Applications, and Future Trends. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2266–2277. [CrossRef]
23. Xu, J.; Wang, S.; Zhou, A.; Yang, F. Edgence: A blockchain-enabled edge-computing platform for intelligent IoT-based dApps. *China Commun.* **2020**, *17*, 78–87. [CrossRef]
24. Wu, K.; Ma, Y.; Huang, G.; Liu, X. A first look at blockchain-based decentralized applications. *Softw. Pract. Exp.* **2021**, *51*, 2033–2050. [CrossRef]
25. Besançon, L.; Da Silva, C.F.; Ghodous, P.; Gelas, J.P. A Blockchain Ontology for DApps Development. *IEEE Access* **2022**, *10*, 49905–49933. [CrossRef]
26. Bamakan, S.M.H.; Motavali, A.; Babaei Bondarti, A. A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Syst. Appl.* **2020**, *154*, 113385. [CrossRef]
27. Xiong, H.; Chen, M.; Wu, C.; Zhao, Y.; Yi, W. Research on Progress of Blockchain Consensus Algorithm: A Review on Recent Progress of Blockchain Consensus Algorithms. *Future Internet* **2022**, *14*, 47. [CrossRef]
28. Khatoon, A.; Verma, P.; Southernwood, J.; Massey, B.; Corcoran, P. Blockchain in Energy Efficiency: Potential Applications and Benefits. *Energies* **2019**, *12*, 3317. [CrossRef]
29. Xu, X.; Zhao, H.; Yao, H.; Wang, S. A Blockchain-Enabled Energy-Efficient Data Collection System for UAV-Assisted IoT. *IEEE Internet Things J.* **2021**, *8*, 2431–2443. [CrossRef]
30. Lasla, N.; Al-Sahan, L.; Abdallah, M.; Younis, M. Green-PoW: An energy-efficient blockchain Proof-of-Work consensus algorithm. *Comput. Netw.* **2022**, *214*, 109118. [CrossRef]
31. Oyinloye, D.P.; Teh, J.S.; Jamil, N.; Alawida, M. Blockchain Consensus: An Overview of Alternative Protocols. *Symmetry* **2021**, *13*, 1363. [CrossRef]
32. Sim, S.H.; Jeong, Y.S. Multi-Blockchain-Based IoT Data Processing Techniques to Ensure the Integrity of IoT Data in AIoT Edge Computing Environments. *Sensors* **2021**, *21*, 3515. [CrossRef]
33. Pourmajidi, W.; Zhang, L.; Steinbacher, J.; Erwin, T.; Miranskyy, A. Immutable Log Storage as a Service on Private and Public Blockchains. *IEEE Trans. Serv. Comput.* **2021**, *16*, 356–369. [CrossRef]
34. Dustdar, S.; Fernández, P.; García, J.M.; Ruiz-Cortés, A. Elastic Smart Contracts in Blockchains. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1901–1912. [CrossRef]
35. Muneeb, M.; Raza, Z.; Haq, I.U.; Shafiq, O. SmartCon: A Blockchain-Based Framework for Smart Contracts and Transaction Management. *IEEE Access* **2022**, *10*, 23687–23699. [CrossRef]
36. Kushwaha, S.S.; Joshi, S.; Singh, D.; Kaur, M.; Lee, H.N. Ethereum Smart Contract Analysis Tools: A Systematic Review. *IEEE Access* **2022**, *10*, 57037–57062. [CrossRef]

37. Madhwal, Y.; Borbon-Galvez, Y.; Etemadi, N.; Yanovich, Y.; Creazza, A. Proof of Delivery Smart Contract for Performance Measurements. *IEEE Access* **2022**, *10*, 69147–69159. [CrossRef]
38. Liu, G.; Wu, J.; Wang, T. Blockchain-enabled fog resource access and granting. *Intell. Converg. Netw.* **2021**, *2*, 108–114. [CrossRef]
39. Basilan, M.L.J.C.; Padilla, M.A. Assessment of teaching English Language Skills: Input to Digitized Activities for campus journalism advisers. *Int. Multidiscip. Res. J.* **2023**, *4*. [CrossRef]
40. Jobin, A.; Ienca, M.; Vayena, E. The global landscape of AI ethics guidelines. *Nat. Mach. Intell.* **2019**, *1*, 389–399. [CrossRef]
41. Wearn, O.R.; Freeman, R.; Jacoby, D.M.P. Responsible AI for conservation. *Nat. Mach. Intell.* **2019**, *1*, 72–73. [CrossRef]
42. Hagendorff, T. The Ethics of AI Ethics: An Evaluation of Guidelines. *Minds Mach.* **2020**, *30*, 99–120. [CrossRef]
43. Saleiro, P.; Rodolfa, K.T.; Ghani, R. Dealing with Bias and Fairness in Data Science Systems: A Practical Hands-on Tutorial. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20, Virtual Event, 6–10 July 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 3513–3514. [CrossRef]
44. Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; Galstyan, A. A Survey on Bias and Fairness in Machine Learning. *ACM Comput. Surv.* **2021**, *54*, 115. [CrossRef]
45. Winfield, A. Ethical standards in robotics and AI. *Nat. Electron.* **2019**, *2*, 46–48. [CrossRef]
46. Barredo Arrieta, A.; Díaz-Rodríguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, R.; et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **2020**, *58*, 82–115. [CrossRef]
47. Feuerriegel, S.; Dolata, M.; Schwabe, G. Fair AI. *Bus. Inf. Syst. Eng.* **2020**, *62*, 379–384. [CrossRef]
48. Bellamy, R.K.E.; Dey, K.; Hind, M.; Hoffman, S.C.; Houde, S.; Kannan, K.; Lohia, P.; Martino, J.; Mehta, S.; Mojsilović, A.; et al. AI Fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM J. Res. Dev.* **2019**, *63*, 4:1–4:15. [CrossRef]
49. Girasa, R. Legal Issues of Digital Technology. In *Regulation of Cryptocurrencies and Blockchain Technologies: National and International Perspectives*; Springer International Publishing: Cham, Switzerland, 2018; pp. 57–69. [CrossRef]
50. Gupta, R.; Tanwar, S.; Al-Turjman, F.; Italiya, P.; Nauman, A.; Kim, S.W. Smart Contract Privacy Protection Using AI in Cyber-Physical Systems: Tools, Techniques and Challenges. *IEEE Access* **2020**, *8*, 24746–24772. [CrossRef]
51. Larsson, S.; Heintz, F. Transparency in artificial intelligence. *Internet Policy Rev.* **2020**, *9*, 1–16. [CrossRef]
52. Andrada, G.; Clowes, R.W.; Smart, P.R. Varieties of transparency: Exploring agency within AI systems. *AI Soc.* **2022**. [CrossRef]
53. Srinivasan, R.; Chander, A. Biases in AI Systems. *Commun. ACM* **2021**, *64*, 44–49. [CrossRef]
54. Holzinger, A.; Langs, G.; Denk, H.; Zatloukal, K.; Müller, H. Causability and explainability of artificial intelligence in medicine. *WIREs Data Min. Knowl. Discov.* **2019**, *9*, e1312.
55. Ding, W.; Abdel-Basset, M.; Hawash, H.; Ali, A.M. Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey. *Inf. Sci.* **2022**, *615*, 238–292. [CrossRef]
56. Shin, D. The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable AI. *Int. J. Hum.-Comput. Stud.* **2021**, *146*, 102551. [CrossRef]
57. Meszaros, J.; Ho, C.-H. AI research and data protection: Can the same rules apply for commercial and academic research under the GDPR? *Comput. Law Secur. Rev.* **2021**, *41*, 105532. [CrossRef]
58. Hamon, R.; Junklewitz, H.; Sanchez, I.; Malgieri, G.; De Hert, P. Bridging the Gap Between AI and Explainability in the GDPR: Towards Trustworthiness-by-Design in Automated Decision-Making. *IEEE Comput. Intell. Mag.* **2022**, *17*, 72–85. [CrossRef]
59. Gade, K.; Geyik, S.C.; Kenthapadi, K.; Mithal, V.; Taly, A. Explainable AI in Industry. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, Anchorage, AK, USA, 4–8 August 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 3203–3204. [CrossRef]
60. Clarke, R. Principles and business processes for responsible AI. *Comput. Law Secur. Rev.* **2019**, *35*, 410–422. [CrossRef]
61. Peters, D.; Vold, K.; Robinson, D.; Calvo, R.A. Responsible AI—Two Frameworks for Ethical Design Practice. *IEEE Trans. Technol. Soc.* **2020**, *1*, 34–47. [CrossRef]
62. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, W.; Chen, X.; Weng, J.; Imran, M. An overview on smart contracts: Challenges, advances and platforms. *Future Gener. Comput. Syst.* **2020**, *105*, 475–491. [CrossRef]
63. Thomas, L.; Zhou, Y.; Long, C.; Wu, J.; Jenkins, N. A general form of smart contract for decentralized energy systems management. *Nat. Energy* **2019**, *4*, 140–149. [CrossRef]
64. McConaghy, T. Ocean Protocol: Tools for the Web3 Data Economy. In *Handbook on Blockchain*; Tran, D.A., Thai, M.T., Krishnamachari, B., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 505–539. [CrossRef]
65. Bartoletti, M.; Bracciali, A.; Lepore, C.; Scalas, A.; Zunino, R. A Formal Model of Algorand Smart Contracts. In Proceedings of the Financial Cryptography and Data Security, Virtual Event, 1–5 March 2021; Borisov, N., Diaz, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; pp. 93–114.
66. Chatila, R.; Havens, J.C. The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. In *Robotics and Well-Being*; Aldinhas Ferreira, M.I., Silva Sequeira, J., Singh Virk, G., Tokhi, M.O., E. Kadar, E., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 11–16. [CrossRef]
67. Floridi, L.; Cowls, J. A Unified Framework of Five Principles for AI in Society. In *Machine Learning and the City*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2022; Chapter 22, pp. 535–545.
68. Butterworth, M. The ICO and artificial intelligence: The role of fairness in the GDPR framework. *Comput. Law Secur. Rev.* **2018**, *34*, 257–268. [CrossRef]

69. Hang, L.; Ullah, I.; Yang, J.; Chen, C. An improved Kalman filter using ANN-based learning module to predict transaction throughput of blockchain network in clinical trials. *Peer-to-Peer Netw. Appl.* **2022**. [CrossRef]
70. Zafar, M.; Valera, I.; Gomez Rodriguez, M.; Gummadi, K.P. Fairness Constraints: Mechanisms for Fair Classification. In Proceedings of the 21th ACM Conference on Computer and Communications Security, ACM, Scottsdale, AZ, USA, 3–7 November 2014; pp. 987–998.
71. Madhavan, R.; Wadhwa, M. Fairness-Aware Learning with Prejudice Free Representations. In Proceedings of the CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, 19–23 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 2137–2140. [CrossRef]
72. Adadi, A.; Berrada, M. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* **2018**, *6*, 52138–52160. [CrossRef]
73. Hailemariam, Y.; Yazdinejad, A.; Parizi, R.M.; Srivastava, G.; Dehghantanha, A. An Empirical Evaluation of AI Deep Explainable Tools. In Proceedings of the 2020 IEEE Globecom Workshops (GC Wkshps), Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
74. Shneiderman, B. Bridging the Gap between Ethics and Practice: Guidelines for Reliable, Safe, and Trustworthy Human-Centered AI Systems. *ACM Trans. Interact. Intell. Syst.* **2020**, *10*, 26. [CrossRef]
75. Akbari-Nodehi, H.; Maddah-Ali, M.A. Secure Coded Multi-Party Computation for Massive Matrix Operations. *IEEE Trans. Inf. Theory* **2021**, *67*, 2379–2398. [CrossRef]
76. Zhang, C.; Ekanut, S.; Zhen, L.; Li, Z. Augmented Multi-Party Computation Against Gradient Leakage in Federated Learning. *IEEE Trans. Big Data* **2022.** Available online: https://ieeexplore.ieee.org/abstract/document/9900067 (accessed on 15 January 2023). [CrossRef]
77. Wang, S.; Yuan, Y.; Wang, X.; Li, J.; Qin, R.; Wang, F.Y. An Overview of Smart Contract: Architecture, Applications, and Future Trends. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 108–113. [CrossRef]
78. Kumari, A.; Gupta, R.; Tanwar, S.; Kumar, N. Blockchain and AI amalgamation for energy cloud management: Challenges, solutions, and future directions. *J. Parallel Distrib. Comput.* **2020**, *143*, 148–166. [CrossRef]
79. Hamilton, M. Blockchain distributed ledger technology: An introduction and focus on smart contracts. *J. Corp. Account. Financ.* **2020**, *31*, 7–12.
80. Christidis, K.; Devetsikiotis, M. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* **2016**, *4*, 2292–2303. [CrossRef]
81. Aloqaily, M.; Boukerche, A.; Bouachir, O.; Khalid, F.; Jangsher, S. An Energy Trade Framework Using Smart Contracts: Overview and Challenges. *IEEE Netw.* **2020**, *34*, 119–125. [CrossRef]
82. Gai, K.; Guo, J.; Zhu, L.; Yu, S. Blockchain Meets Cloud Computing: A Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2009–2030. [CrossRef]
83. Sharma, P.; Jindal, R.; Borah, M.D. Blockchain Technology for Cloud Storage: A Systematic Literature Review. *ACM Comput. Surv.* **2020**, *53*, 89. [CrossRef]
84. Tosh, D.; Shetty, S.; Liang, X.; Kamhoua, C.; Njilla, L.L. Data Provenance in the Cloud: A Blockchain-Based Approach. *IEEE Consum. Electron. Mag.* **2019**, *8*, 38–44. [CrossRef]
85. Sharma, V. An Energy-Efficient Transaction Model for the Blockchain-Enabled Internet of Vehicles (IoV). *IEEE Commun. Lett.* **2019**, *23*, 246–249. [CrossRef]
86. Liu, Y.; Su, Z.; Wang, Y. Energy-Efficient and Physical Layer Secure Computation Offloading in Blockchain-Empowered Internet of Things. *IEEE Internet Things J.* **2022.** Available online: https://ieeexplore.ieee.org/abstract/document/9733890 (accessed on 15 January 2023). [CrossRef]
87. Wong, S.; Yeung, J.K.W.; Lau, Y.Y.; So, J. Technical Sustainability of Cloud-Based Blockchain Integrated with Machine Learning for Supply Chain Management. *Sustainability* **2021**, *13*, 8270. [CrossRef]
88. Khan, S.N.; Loukil, F.; Ghedira-Guegan, C.; Benkhelifa, E.; Bani-Hani, A. Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-Peer Netw. Appl.* **2021**, *14*, 2901–2925. [CrossRef]
89. Negara, E.S.; Hidayanto, A.N.; Andryani, R.; Syaputra, R. Survey of Smart Contract Framework and Its Application. *Information* **2021**, *12*, 257. [CrossRef]
90. Harris, C.G. The Risks and Challenges of Implementing Ethereum Smart Contracts. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Republic of Korea, 14–17 May 2019; pp. 104–107. [CrossRef]
91. Hanada, Y.; Hsiao, L.; Levis, P. Smart Contracts for Machine-to-Machine Communication: Possibilities and Limitations. In Proceedings of the 2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS), Bali, Indonesia, 1–3 November 2018; pp. 130–136. [CrossRef]
92. Cano-Benito, J.; Cimmino, A.; García-Castro, R. Toward the Ontological Modeling of Smart Contracts: A Solidity Use Case. *IEEE Access* **2021**, *9*, 140156–140172. [CrossRef]
93. Ali, J.; Syed, T.A.; Musa, S.; Zahrani, A. Towards Secure IoT Communication with Smart Contracts in a Blockchain Infrastructure. *arXiv* **2020**, arXiv:2001.01837.

94. Corrales, M.; Jurčys, P.; Kousiouris, G. Smart Contracts and Smart Disclosure: Coding a GDPR Compliance Framework. In *Legal Tech, Smart Contracts and Blockchain*; Corrales, M., Fenwick, M., Haapio, H., Eds.; Springer: Singapore, 2019; pp. 189–220. [CrossRef]
95. Sgantzos, K.; Grigg, I. Artificial Intelligence Implementations on the Blockchain. Use Cases and Future Applications. *Future Internet* **2019**, *11*, 170. [CrossRef]
96. Akter, S.; Michael, K.; Uddin, M.R.; McCarthy, G.; Rahman, M. Transforming business using digital innovations: The application of AI, blockchain, cloud and data analytics. *Ann. Oper. Res.* **2020**, *308*, 7–39. [CrossRef]
97. Sandner, P.; Gross, J.; Richter, R. Convergence of Blockchain, IoT, and AI. *Front. Blockchain* **2020**, *3*, 522600. [CrossRef]
98. Chamola, V.; Hassija, V.; Gupta, V.; Guizani, M. A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain, and 5G in Managing its Impact. *IEEE Access* **2020**, *8*, 90225–90265. [CrossRef]
99. O'Leary, D.E. Embedding AI and Crowdsourcing in the Big Data Lake. *IEEE Intell. Syst.* **2014**, *29*, 70–73. [CrossRef]
100. O'Leary, D.E. Artificial Intelligence and Big Data. *IEEE Intell. Syst.* **2013**, *28*, 96–99. [CrossRef]
101. Moreno, C.; Carrasco, R.A.; Herrera-Viedma, E. Data and Artificial Intelligence Strategy: A Conceptual Enterprise Big Data Cloud Architecture to Enable Market-Oriented Organisations. *Int. J. Interact. Multimed. Artif. Intell.* **2019**, *5*, 7. [CrossRef]
102. Baumer, B.S. A Grammar for Reproducible and Painless Extract-Transform-Load Operations on Medium Data. *J. Comput. Graph. Stat.* **2019**, *28*, 256–264. [CrossRef]
103. Raza, M.Q.; Khosravi, A. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renew. Sustain. Energy Rev.* **2015**, *50*, 1352–1372. [CrossRef]
104. Patil, A.S.; Hamza, R.; Hassan, A.; Jiang, N.; Yan, H.; Li, J. Efficient privacy-preserving authentication protocol using PUFs with blockchain smart contracts. *Comput. Secur.* **2020**, *97*, 101958. [CrossRef]
105. Rouhani, S.; Deters, R. Security, Performance, and Applications of Smart Contracts: A Systematic Survey. *IEEE Access* **2019**, *7*, 50759–50779. [CrossRef]
106. Sun, T.; Yu, W. A Formal Verification Framework for Security Issues of Blockchain Smart Contracts. *Electronics* **2020**, *9*, 255. [CrossRef]
107. Hang, L.; Kim, D.H. Reliable Task Management Based on a Smart Contract for Runtime Verification of Sensing and Actuating Tasks in IoT Environments. *Sensors* **2020**, *20*, 1207. [CrossRef] [PubMed]
108. Leung, D.; Suhl, A.; Gilad, Y.; Zeldovich, N. Vault: Fast Bootstrapping for the Algorand Cryptocurrency. In Proceedings of the 2019 Network and Distributed System Security Symposium, Internet Society, San Diego, CA, USA, 24–27 February 2019. [CrossRef]
109. Gilad, Y.; Hemo, R.; Micali, S.; Vlachos, G.; Zeldovich, N. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In Proceedings of the 26th Symposium on Operating Systems Principles, SOSP '17, Shanghai, China, 28 October 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 51–68. [CrossRef]
110. Montes, G.A.; Goertzel, B. Distributed, decentralized, and democratized artificial intelligence. *Technol. Forecast. Soc. Chang.* **2019**, *141*, 354–358. [CrossRef]
111. Steichen, M.; Hommes, S.; State, R. ChainGuard—A firewall for blockchain applications using SDN with OpenFlow. In Proceedings of the 2017 Principles, Systems and Applications of IP Telecommunications (IPTComm), Chicago, IL, USA, 25–28 September 2017; pp. 1–8. [CrossRef]
112. Flittner, M.; Scheuermann, J.M.; Bauer, R. ChainGuard: Controller-independent verification of service function chaining in cloud computing. In Proceedings of the 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Berlin, Germany, 6–8 November 2017; pp. 1–7. [CrossRef]
113. Sayeed, S.; Marco-Gisbert, H.; Caira, T. Smart Contract: Attacks and Protections. *IEEE Access* **2020**, *8*, 24416–24427. [CrossRef]
114. Stahl, B.C.; Wright, D. Ethics and Privacy in AI and Big Data: Implementing Responsible Research and Innovation. *IEEE Secur. Priv.* **2018**, *16*, 26–33. [CrossRef]
115. Mothukuri, V.; Parizi, R.M.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; Srivastava, G. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.* **2021**, *115*, 619–640. [CrossRef]
116. Zhu, T.; Ye, D.; Wang, W.; Zhou, W.; Yu, P.S. More Than Privacy: Applying Differential Privacy in Key Areas of Artificial Intelligence. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 2824–2843. [CrossRef]
117. Lee, J.H.; Kim, H. Security and Privacy Challenges in the Internet of Things [Security and Privacy Matters]. *IEEE Consum. Electron. Mag.* **2017**, *6*, 134–136. [CrossRef]
118. Dilmaghani, S.; Brust, M.R.; Danoy, G.; Cassagnes, N.; Pecero, J.; Bouvry, P. Privacy and Security of Big Data in AI Systems: A Research and Standards Perspective. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 5737–5743. [CrossRef]
119. Alkhalifah, A.; Ng, A.; Watters, P.A.; Kayes, A.S.M. A Mechanism to Detect and Prevent Ethereum Blockchain Smart Contract Reentrancy Attacks. *Front. Comput. Sci.* **2021**, *3*, 598780. [CrossRef]
120. Ferreira Torres, C.; Iannillo, A.K.; Gervais, A.; State, R. The Eye of Horus: Spotting and Analyzing Attacks on Ethereum Smart Contracts. In Proceedings of the Financial Cryptography and Data Security, Virtual Event, 1–5 March 2021; Borisov, N., Diaz, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; pp. 33–52.

121. Xue, Y.; Fu, J.; Su, S.; Bhuiyan, Z.A.; Qiu, J.; Lu, H.; Hu, N.; Tian, Z. Preventing Price Manipulation Attack by Front-Running. In Proceedings of the Advances in Artificial Intelligence and Security, Qinghai, China, 15–20 July 2022; Sun, X., Zhang, X., Xia, Z., Bertino, E., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 309–322.
122. Stathakopoulou, C.; Rüsch, S.; Brandenburger, M.; Vukolić, M. Adding Fairness to Order: Preventing Front-Running Attacks in BFT Protocols using TEEs. In Proceedings of the 2021 40th International Symposium on Reliable Distributed Systems (SRDS), Chicago, IL, USA, 20–23 September 2021; pp. 34–45. [CrossRef]
123. Carl, G.; Kesidis, G.; Brooks, R.; Rai, S. Denial-of-service attack-detection techniques. *IEEE Internet Comput.* **2006**, *10*, 82–89. [CrossRef]
124. Moore, D.; Shannon, C.; Brown, D.J.; Voelker, G.M.; Savage, S. Inferring Internet Denial-of-Service Activity. *ACM Trans. Comput. Syst.* **2006**, *24*, 115–139. [CrossRef]
125. Bardou, R.; Focardi, R.; Kawamoto, Y.; Simionato, L.; Steel, G.; Tsay, J.K. Efficient Padding Oracle Attacks on Cryptographic Hardware. In Proceedings of the Advances in Cryptology—CRYPTO 2012, Santa Barbara, CA, USA, 19–23 August 2012; Safavi-Naini, R., Canetti, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 608–625.
126. Venturini, I. Counteracting Oracle Attacks. In Proceedings of the 2004 Workshop on Multimedia and Security, MM&Sec '04, Magdeburg, Germany, 20–21 September 2004; Association for Computing Machinery: New York, NY, USA, 2004; pp. 187–192. [CrossRef]
127. Fang, L.; Zhao, B.; Li, Y.; Liu, Z.; Ge, C.; Meng, W. Countermeasure Based on Smart Contracts and AI against DoS/DDoS Attack in 5G Circumstances. *IEEE Netw.* **2020**, *34*, 54–61. [CrossRef]
128. Aggarwal, S.; Kumar, N. Chapter Twenty—Attacks on blockchain. In *The Blockchain Technology for Secure and Smart Applications across Industry Verticals*; Advances in Computers; Aggarwal, S., Kumar, N., Raj, P., Eds.; Elsevier: Amsterdam, The Netherlands, 2021; Volume 121, pp. 399–410. [CrossRef]
129. Balcerzak, A.P.; Nica, E.; Rogalska, E.; Poliak, M.; Kliestik, T.; Sabie, O.M. Blockchain Technology and Smart Contracts in Decentralized Governance Systems. *Adm. Sci.* **2022**, *12*, 96. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.