IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON CLOUD - FOG - EDGE COMPUTING IN CYBER-PHYSICAL-SOCIAL SYSTEMS (CPSS)

Received January 17, 2020, accepted January 27, 2020, date of publication January 30, 2020, date of current version February 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2970576

Smart Contract Privacy Protection Using AI in Cyber-Physical Systems: Tools, Techniques and Challenges

RAJESH GUPTA[1], SUDEEP TANWAR[1], FADI AL-TURJMAN[2,3], PRIT ITALIYA[1], ALI NAUMAN[4], AND SUNG WON KIM[4]

[1]Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad 382481, India
[2]Artificial Intelligence Department, Near East University, 99138 Mersin, Turkey
[3]Research Center for AI and IoT, Near East University, 99138 Mersin, Turkey
[4]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, South Korea

Corresponding author: Sung Won Kim (swon@yu.ac.kr)

This work was supported in part by the Brain Korea 21 Plus Program funded by the National Research Foundation of Korea (NRF), under Grant 22A20130012814, in part by the MSIT (Ministry of Science and ICT), South Korea, through the Information Technology Research Center (ITRC) Support Program supervised by the Institute for Information and communications Technology Planning and Evaluation (IITP) under Grant IITP-2019-2016-0-00313, and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2018R1D1A1A09082266.

ABSTRACT Applications of Blockchain (BC) technology and Cyber-Physical Systems (CPS) are increasing exponentially. However, framing resilient and correct smart contracts (SCs) for these smart application is a quite challenging task because of the complexity associated with them. SC is modernizing the traditional industrial, technical, and business processes. It is self-executable, self-verifiable, and embedded into the BC that eliminates the need for trusted third-party systems, which ultimately saves administration as well as service costs. It also improves system efficiency and reduces the associated security risks. However, SCs are well encouraging the new technological reforms in Industry 4.0, but still, various security and privacy challenges need to be addressed. In this paper, a survey on SC security vulnerabilities in the software code that can be easily hacked by a malicious user or may compromise the entire BC network is presented. As per the literature, the challenges related to SC security and privacy are not explored much by the authors around the world. From the existing proposals, it has been observed that designing a complex SCs cannot mitigate its privacy and security issues. So, this paper investigates various Artificial Intelligence (AI) techniques and tools for SC privacy protection. Then, open issues and challenges for AI-based SC are analyzed. Finally, a case study of retail marketing is presented, which uses AI and SC to preserve its security and privacy.

INDEX TERMS Cyber-physical system, blockchain, smart contract, artificial intelligence, security, privacy.

I. INTRODUCTION

Cyber-Physical Systems (CPS) are integrating the networking, computing, and physical processes, where modern sensors handle its major components efficiently, such as a cyber system and a physical process [1], [2]. However, as the interaction between these two components increases, then the physical systems is more prone to the security vulnerabilities. To handle these vulnerabilities, Blockchain (BC) is a breakthrough technology, which secure and execute transactions in an open network environment without the involvement of any centralized third-party system (TTS) [3], [4]. It is a distributed ledger, which records all transactions into the chain of blocks. A transaction is being verified by all the participating members of a BC before storing it into the chain of blocks [5]. Decentralized consensus algorithms such as proof-of-work (PoW), proof-of-stake (PoS), and proof-of-identity (PoI) can be used for data synchronization in peer-to-peer (P2P) BC network [6]. The peers' nodes need a digital agreement (known as SC) to complete a transaction that can be self-executable, self-verifiable, and self-enforced.

From the past few years, BC technology is growing at an exponential rate globally. As per the report *Fortune Business Insights* [7], the estimated BC market capital will increase up to $21070.2 Million (approx. $20 Billion) by 2025 as shown in FIGURE 1. Many renowned financial institutions and

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Arafatur Rahman.

TABLE 1. Details on blockchain acceptability reports.

Organization	Year	Country	Report Title	Cryptocurrency status
Australian Parliament's Senate Economic References Committee	2015	Australia	Digital Currency - Game Changer or Bit Player	Legal
Central Bank of Kenya	2015	Kenya	NA	Not Legal
Mark Walport	2016	U.K.	Distributed ledger technology: beyond BC	Legal
Pinna et al.	2016	Europe	Distributed ledger technologies in securities post-trading	Legal
China BC Technology and Industry Development Forum	2016	China	China BC Technology and Application Development White Paper	Legal
Rebecca Campbell	2017	USA	Delaware Passes Groundbreaking BC Regulation Bill	Legal
Nepal Rastra Bank	2017	Nepal	NA	Not Legal
Austrian Ministry of Finance	2017	Austria	NA	Partially Legal
Reserve Bank of Zimbabwe	2017	Zimbabwe	NA	Not Legal
Saudi Arabian Monetary Agency	2017	Saudi Arabia	Pilot project to issue a local digital currency (Riyal)	Legal
Government of India	2018	India	NA	Not Legal
Zambia Securities and Exchange Commission	2018	Zambia	NA	Not Legal

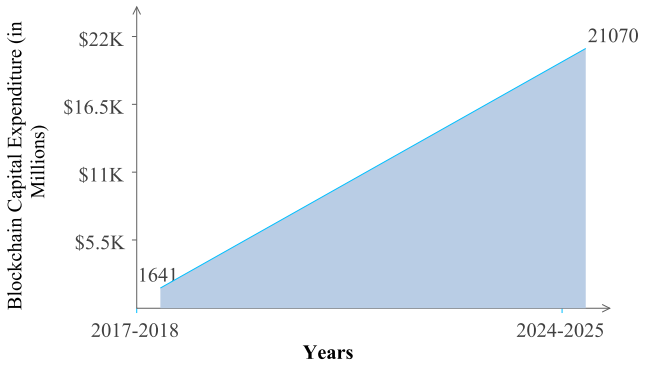

FIGURE 1. Blockchain capital market size [7].

multi-national IT sectors around the world are doing exhaustive research on the acceptance of BC technology [8], [9]. India has also started exploring BC for various applications like a land registry, digital certifications, insurance, agriculture, banking [10], and e-governance to prevent it from forgery [11].

Governments across the globe have put their efforts and released the technical reports on BC. It shows the development and adoption of BC technology, as shown in Table 1. This survey is merely focusing on the security and privacy issues in SCs and how AI solves those.

However, SC is a piece of software code written in a specific programming language such as solidity, Go, Kotlin, or Python. It is an enforceable and immutable program code [12], [13]. It is responsible for implementing, compiling, and deploying logics of digital assets for automated execution [14]. SCs are hard to modify once written. The trust, credibility, and the complexity of transactions in a decentralized system can be well managed and maintained with the involvement of SC [15]. It also results in ease of designing real-world decentralized problems (such as financial systems) with low cost and improved accuracy. One of the issues with SC is that it is impossible to fix the discovered bugs after its deployment [16].

Currently, the progress and development of SCs are still is infancy. Soon, most of the organizations will acquire BC technology and are governed by SCs [17]. This increased adoption of SCs needs efficient and robust security procedures [18]. It is quite challenging for developers to create a secure and bug-free SCs. The possible security vulnerabilities in SCs are re-entrancy, transaction-ordering dependence, forcibly sending ether to a contract, DoS with revert, and integer overflow & underflow. SC developers must be aware of such vulnerabilities while designing SCs [19]. These vulnerabilities can potentially lead to monetary losses in a million USD in the past few years. In 2016, 150 million were stolen from the DAO contract, then in 2017, 30 million were stolen from the Parity multi-signature wallet [20].

Further, Kiffer *et al.* [21] analyzes the diversity in designing SCs are quite low, i.e., most of them are almost the reflection of each other. Due to their similar nature, the forging in SC becomes easy for malicious users to disrupt or modify the BC transaction procedures [22]. Due to this, the unexpected behavior of a malicious user is hard to capture. For example, an automatic compensation will be given to the air ticket holders whose flights are delayed by a certain time bound [23]. Due to privacy breach (or modification attack) [24] on SC by a malicious user, this analysis may go wrong, i.e., passengers may not be benefited in case of flight delay [25].

For such cases, AI techniques can be used to analyze the SC and the behavior of the participating members who have given their consent for the execution of a transaction. It would help in analyzing the SC transaction execution patterns and identify malicious patterns. Moreover, AI with natural language processing (NLP) can also be used for an in-depth analysis of SC patterns. The power of NLP can be used for semantic parsing and entity recognition. Both AI and NLP techniques empower the analysis and help to create highly complex and compelling SC [26]. The more the data, the more effective the analysis will be. So, the idea of integrating AI with the BC and SC offers a powerful solution in maintaining the privacy and security of SCs and BC transactions [27].

So, the integration of BC with AI, cryptography algorithms, digital signatures, and big data are the next-generation computing technologies for Industry 4.0 [28], [29] in context to security, privacy, and analytics.

A. SCOPE OF THE SURVEY

So far, several surveys and methodologies conducted by different authors across the globe, which considered

TABLE 2. Comparison of the existing surveys with the proposed survey.

Author	Year	Description	Methods Used	1	2	3	4	5	6	7	8	Merits	Demerits
Geng *et al.* [34]	2017	Proposed an incentive-based model for BC SCs to achieve privacy in location based services	Group key encryption, blind collective signature	✓	✓	X	X	X	X	✓	X	User incentive mechanism, SC accessible to K-anonymity group	Results of privacy and efficiency are not discussed
Tasatanattakool *et al.* [30]	2018	Presented a survey on BC applications and challenges faced	-	X	X	X	✓	X	X	X	✓	Financial and non-financial applications aspects	Only healthcare related issues discussed
Wang *et al.* [31]	2018	Given a survey on SC and its applications along with recent and future trends.	-	✓	✓	✓	✓	X	X	X	✓	Challenges in SC	Potential attacks on SC are not discussed
Kurtulmus *et al.* [32]	2018	Proposed a contract to give rewards to machine learning models	Machine learning models	X	✓	✓	X	X	X	✓	X	A market for effective machine learning trained models	GPU training cost is high.
Marwala *et al.* [36]	2018	Discussed the issues and benefits of integration of AI and BC	-	X	✓	✓	X	X	X	X	X	Better understanding of BC and AI integration	Requires more explanation on countermeasures
Dinh *et al.* [35]	2018	Discussion over integration of AI with BC and different aspects of security and accuracy	AI	X	✓	✓	X	X	X	X	X	Explanation of AI for BC for AI with all related terms	No statistical data or defence mechanism described
Mamoshina *et al.* [33]	2018	BC and AI to improve biomedical research	Incentive mechanism and AI	✓	✓	✓	X	✓	X	✓	✓	User controlled health monitoring	-
Dika *et al.* [38]	2018	Analyzed the security vulnerabilities of SC and their countermeasures	Code analysis tools such as solidity and EVM	X	✓	X	X	X	X	✓	X	Analysis of SC vulnerabilities (DAO and Re-entrancy hacks)	-
Salah *et al.* [37]	2019	Discussion over BC applications for AI	-	✓	✓	✓	✓	✓	X	X	✓	In-depth discussion on BC, SC and consensus mechanisms	No case study mentioned by considering any scenario
Cheng *et al.* [39]	2019	Presented an Ekiden system secure SCs	Trusted Execution Environment	✓	X	✓	X	X	X	✓	✓	confidentiality and privacy	only mathematical model, no implementation
Li *et al.* [40]	2019	Discusses the scalability and privacy issues of SCs	off-chain contract and an on-chain contract	X	X	X	X	X	X	✓	✓	Scalability and privacy achieved	No security achieved
Proposed Survey	2020	Proposed a Survey of AI techniques for SC privacy protection	-	✓	✓	✓	✓	✓	✓	✓	✓	Discussed privacy protection techniques with their current challenges	-

Parameters: 1. Architecture Suggested, 2. Security Analysis, 3. AI Techniques, 4. Open Issues and Research Challenges, 5. Taxonomy, 6. Case Study, 7. SC Implementation Scenario, 8. Applications Discussed
Considered: ✓, Not considered: *X*

different features of SC with AI [30]–[38]. By considering all the literature surveys, most of them are using BC to ensure security and privacy, whereas others are using AI techniques for the same. As per our knowledge, only a few surveys are there, which considered AI to use with BC to ensure privacy. But there is no proper explanation of mechanisms through which privacy can be achieved. The proposed survey spans mostly over the last four years (2015 to date). Tasatanattakool and Techapanupreeda [30] described BC applications and its challenges. Wang *et al.* [31] described SC introduction and its applications. Kurtulmus and Daniel [32] proposed an incentive mechanism for each successful machine learning trained model. Mamoshina *et al.* [33] focused on using AI and BC for bio-medical research. Geng *et al.* [34] proposed a secure authentication model for BC with k-anonymity. Dinh and Thai [35] focused on integration of AI and BC. Marwala and Xing [36] discussed issues and benefits of integration of BC and AI. Salah *et al.* [37] described various BC applications with AI. Table 2 shows the comparison of existing surveys published in the domain of SC and its privacy.

B. CONTRIBUTIONS

In this paper, we present a review on the integration of AI techniques in SCs for privacy preservation. We explored various existing solutions to secure SCs in an open public environment. Our primary focus is to analyze the role of AI in SC applications and how they can address security, privacy, and computation issues. Based on the above discussion, the following are the significant contributions of this paper.

- We present a systematic and comprehensive review of SC vulnerabilities and privacy-preserving using AI-techniques.
- Review of the recently proposed decentralized AI platforms for SCs.
- A discussion on SC working principles using a case study.
- Important open issues and challenges in the integration of AI in SCs.

C. ORGANIZATION AND READING MAP

The structure of the paper is as shown in FIGURE 2. Table 3 lists all the acronyms used in the paper. The organization of

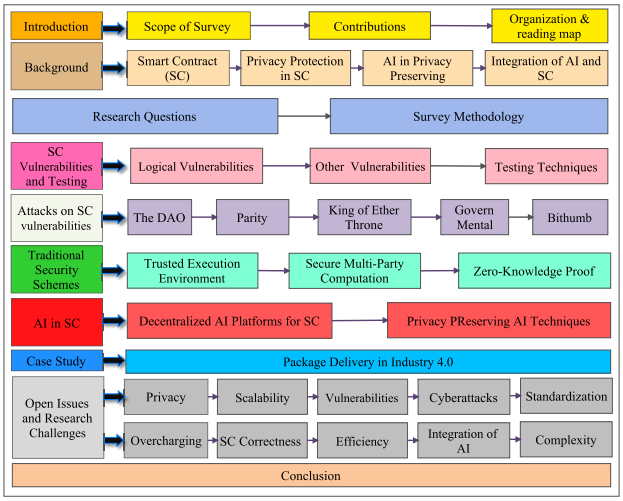

FIGURE 2. Organization of the survey.

TABLE 3. Abbreviations table.

Acronym	Description
BC	Blockchain
SC	Smart Contract
IoT	Internet of Things
AI	Artificial Intelligence
PoW	Proof-of-Work
P2P	Peer-to-peer
NLP	Natural Language Processing
TC	Turing Complete
TIC	Turing Incomplete
RBAC-SC	Role-based Access Control using Smart Contract
D2D	Device-to-Device
LV	Logical Vulnerabilities
OV	Other Vulnerabilities
DAO	Decentralized Autonomous Organization
TEE	Trusted Execution Environment
SMPC	Secure Multi-Party Computation
ZKP	Zero-Knowledge Proof
SGX	Intel Software Guard Extension
CWE	Current Work Environment
SWE	Secure Work Environment
REE	Rich Execution Environment
NEW	Normal Execution World
SEW	Secure Execution World
TZ	Trust Zone
MPC	Multi-Party Computation
PoI	Proof of Identity
DAI	Decentralized AI
MAN	Matrix AI Network
DBC	Deep Brain Chain
ANN	Artificial Neural Network
IDS	Intrusion Detection System
IPS	Intrusion Protection System
SI	Swarm Intelligence
CI	Computational Intelligence
AP	Access Point
ML	Machine Learning
ICRS	Icecream Retailing Store

the rest of the paper is as follows. In Section II, the basic overview of SC and its integration with AI for privacy protection. In Section IV, we discuss various SC vulnerabilities and their testing procedures. Section V discusses about the real-life security attacks on decentralized organizations. Further, Section VI highlights the traditional security schemes to secure SC security and privacy. Then, Section VII discusses various AI-based tools and techniques to secure SCs.

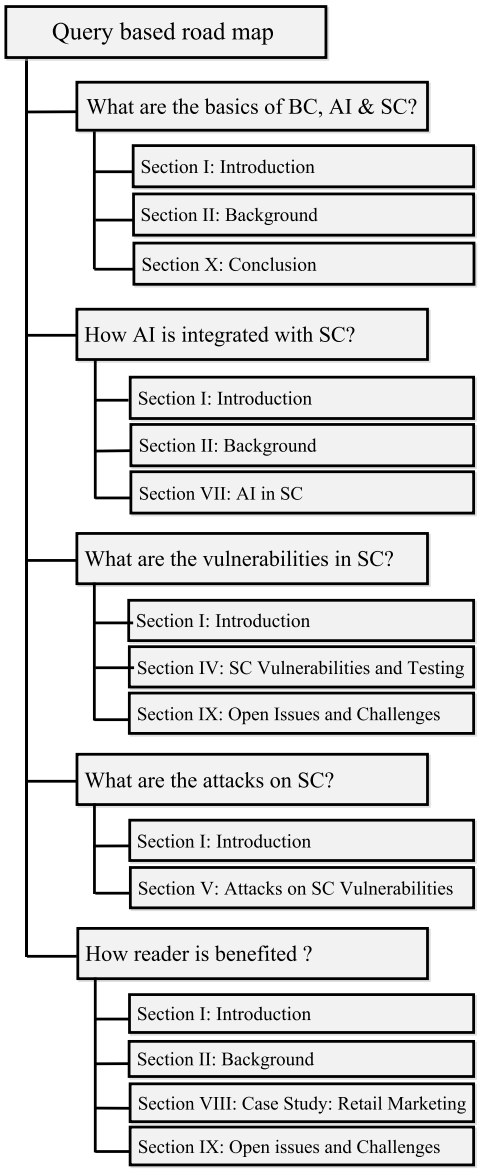

FIGURE 3. A reading map.

Section VIII presents the case study of AI-based SC in retail marketing. Further, Section IX discussed the open issues and challenges in implementing SCs using AI techniques. Then the paper is concluded in Section X.

FIGURE 3 provides a reading map. Readers with interests in the basics of BC, AI, and SC can focus their reading on Sections I, II, and X. The applicability of AI in BC network and SCs for security and efficiency are given in Sections I, II, and VII. Readers with interest in knowing the vulnerabilities of SC such as logical as well as other vulnerabilities can focus their reading on Sections I, IV, and IX. Various real-life scenario attacks are given in Sections I and V. Finally, we recommend Sections I, II, VIII, and IX to the readers interested to gain a high-level overview of AI-enabled SCs including open issues and research challenges that need to be addressed to enhance the security of SCs.

II. BACKGROUND

This section describes the background, history, and significance of SC along with its privacy protection using AI techniques. This section is divided into four subsections. Firstly, we discuss the structure and working of SCs and languages preferred for writing it. Then, we discuss the privacy protection schemes for SC and future trends. Then, we explained the basics of AI and its application areas. At last, we highlight the benefits of the integration of AI and SC, which forms a base for the proposed survey.

A. SMART CONTRACT

A contract is an agreement signed between the parties to do certain things. Once the contract signed, it cannot be void. Traditionally, paper-based contracts were managed under the supervision of TTS (for validation check) [41]. The issues with the traditional contracts are as follows:

- They are passive in nature (manual enforcement of contract).
- Asymmetric information (both parties may not have same data).
- They are inefficient (manual surveillance).
- They are costly (involvement of TTS).

Then, the concept of SC came into the picture around the early 1990s was given by Nick Szabo (a cryptographer). It is a transaction protocol that executes automatically and performs various operations like funds transfer, calculations, and information storage [42]. It is applicable in multiple fields such as smart homes, asset management, and e-commerce [43]. It helps BC applications run efficiently and solve real-world problems with minimum time and cost. But, still, the world is using paper-based contracts that involves the trusted TTS, which can lead to security issues and high transaction cost [44]. BC technology solved such issues by eliminating the need for trusted TTS [45]. SC is one of the successful applications of the BC technology that reduces the transaction cost, execution time, and increases the security [46].

As mentioned above, SC is a programming code with a defined set of rules and allows decentralized automation. Upon execution, SC executed itself and enforced the agreement conditions. Various important characteristics of SCs are shown in FIGURE 4. An SC includes value, address, functions, and state of the transactions executed between the participating nodes. It accepts transactions as an input, applies a function to execute the code, and provides the output [42].

The working procedure of SC creation between two parties is stated in FIGURE 5. It works on conditional principles (*if-then*) that settle the transactions only if agreed on the stated conditions. A SC received the data from the parties, initiates the transaction, executes it, and transfers the digital assets to the said party if satisfied with the specified set of rules and conditions. Finally, the transaction is settled between the parties.

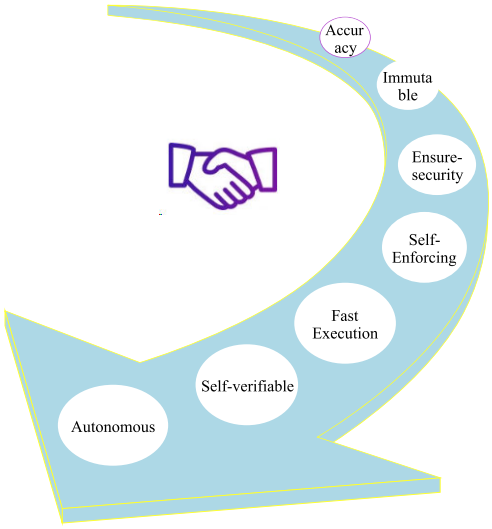

FIGURE 4. Smart contract characteristics.

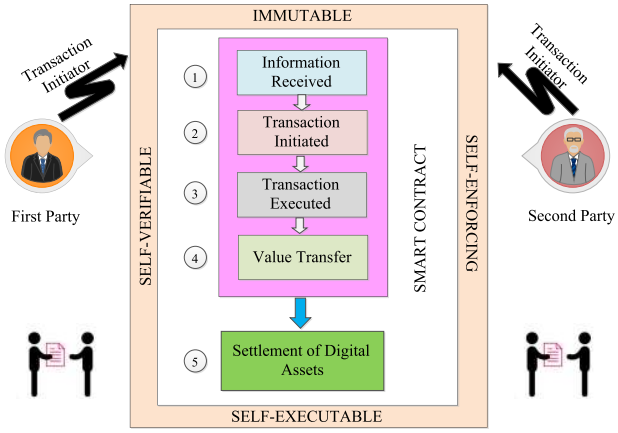

FIGURE 5. Smart contract creation between two parties [55].

SCs are generally developed in a programming language known as *Solidity, Go, Kotlin/Java, or C++*' over various BC platforms like Ethereum, TRON, Stellar, and Hyperledger. Table 4 shows the comparative analysis of SC platforms over the parameters like execution environment, language, Turing completeness, consensus mechanism, cryptocurrency, and applications [47]. Such platforms are Ethereum, Rootstock (RSK), EOS, The Real-time Operating system Nucleus (TRON), Steller, Hyperledger Fabric, cardano, and corda. These platforms ensures the data to be protected from Cybersecurity attacks (such as modification, spoofing, and fabrication attacks) and real-time settlement of digital payment. However, SC can be categorized into Turing Complete (TC) and Turing Incomplete (TIC). TC has considerably adaptable in various BC platforms (Ethereum, Hyperledger Fabric, R3 Corda, and Hyperledger Sawtooth) and can be used to develop complex business rules (may go into an infinite loop) [14]. However, TIC is simple to implement, efficient, and hard to make complex rules.

TABLE 4. Comparison of smart contract platforms [47].

	Execution Environment	Language	Turing Completeness	Consensus Mechanism	Permission Type	Cryptocurrency	Cryptocurrency market value	Applications
Etherium	EVM	Solidity	Turing complete	Proof-Of-Work	Public	Ether	US$141.71	Decentralized exchanges, gambling
RSK	RVM (Rootstock Virtual Machine)	Solidity	Turing complete	merge-mining/ federated	Public	SBTC	US$5862.15	Charity, Insurance
EOS	Web Assembly	C++	Turing Complete	BFT-DPOS	Public	EOS Token	USUS$2.56	Profit sharing, copyright security
TRON	Tron Virtual Machine	Solidity	Turing Complete	TRON (DPOS)	Public	Tronix	US$0.013939	Gaming application, currency
Steller	Docker	Net, Scala, C++, GO	Turing Incomplete	Steller Consensus Protocol	Consortium	Lumen	US$0.051458	Universal payment solution, oil trade
Hyperledger Fabric	Docker	Javascript, GO, java	Turing Complete	Custom protocols	Private	None	-	Smart Energy management, supply chain
Nem	JVM	NEM, Java	Turing Incomplete	Proof-Of-Importance (POI)	private/ public	XEM	US$0.035290	Cryptocurrency, liquid assets
Cardano	K-EVM and IELEVM	Plutus, Rust, C, Javascript	Turing-complete	Ouroboros (POS)	Public	Cardano - SL	US$0.036161	Decentralized, security, gambling
Corda	JVM	Java, Kotlin	Turing Incomplete	Raft	Private	Real-World currencies	-	Construction, Healthcare

Sato and Himura [48] proposed a permissioned BC system with SC to handle cross-organizational operations in a unified and synchronized way. Watanabe *et al.* [49] proposed a secure mechanism to prevent attackers from manipulating the organizational resources with digital rights specified in SC. Ellul and Pace [50] described a virtual machine for BC, which allows IoT devices [51], [52] to maintain communication with BC network using SC. Dickerson *et al.* [53] stated the execution process of SC and Cruz *et al.* [54] introduced a secure access control model RBAC-SC (Role-based access control using SC). They have used SC to fulfill the security requirements of the BC system.

B. PRIVACY PROTECTION IN SMART CONTRACT

In conventional BC systems, the privacy of data can be affected due to its distributed nature. Involvement SC in BC can resolve this issue. It exists across the decentralized BC network and auto-executes when the parties are interested. Currently, Ethereum SCs (Public BC) manages an enormous number of assets; its execution correctness is essential and prevents against security attacks. Authors in [18] proposed a *"Securify"* system, which analyzes the security of Ethereum SC and displays the result as safe/unsafe. This system has tested over 18K SCs designed by users and successfully classified them into true errors and actual errors. Then, a security assurance method was proposed to analyze, identify, and report the security risks of SC [56]. They experimented over 2952 SC instances and found the efficiency and usability of their approach is quite better compared to other discussed approaches.

Wright and Serguieva [57] proposed an auto-management method with hierarchical structures using cryptographic key-pairs to improve the BC services, mainly SC. In addition to this, they also introduced a method for effective and secure transfer of SC entities among other SCs. Ramachandran and Kantarcioglu [58] suggested a novel idea of using BC as a service to make data provenance easier. Here, SCs are used to utilize the developed model and work securely and effectively. No third person who is not a part of BC can access the distributed ledger, which ensures the privacy of data.

Another privacy-preserving application of SC over the Ethereum BC network is discussed in [59]. They applied SC mechanism in trading systems to keep seller and buyer data safe and immutable. To build trust among depositors, a device-to-device (D2D), secure communication is used to collect the deposits. This system ensures high performance, low communication cost, and privacy. Since the release of the Ethereum network, SC is used to manage the ethers. But, managing such a vast amount of currency led to conflicts in SCs. The traditional software engineering has no standardized framework to resolve the issue, as mentioned above. Destefanis *et al.* [60] discussed the need for BC software engineering to address the issue related to SC-based applications. A case study of SC library is analyzed where an attack on *Parity* [61] took place due to a software bug or unsafe programming.

SC systems allow cryptocurrency transactions to be placed securely without any trusted third parties. Kosba *et al.* [62] proposed HAWK system, which does not store any financial transaction details on BC. Using HAWK, a programmer can write the private SC without using any cryptographic schemes that automatically generate an effective cryptographic model. It was the first formal BC cryptography model.

C. ARTIFICIAL INTELLIGENCE IN PRIVACY PRESERVING

AI is the science and engineering of developing intelligent machines that can think as humans [63]. This concept was first introduced by John McCarthy (also known as the father of AI). The goal of AI is to develop expert systems that can do intelligent tasks, learn new things, understand the situation, behave, and advice like humans can do [64], [65]. The very first simple application of AI was developed as "*Weather Forecasting*". Nowadays, almost all applications (such as movie recommendations, early disease diagnosis,

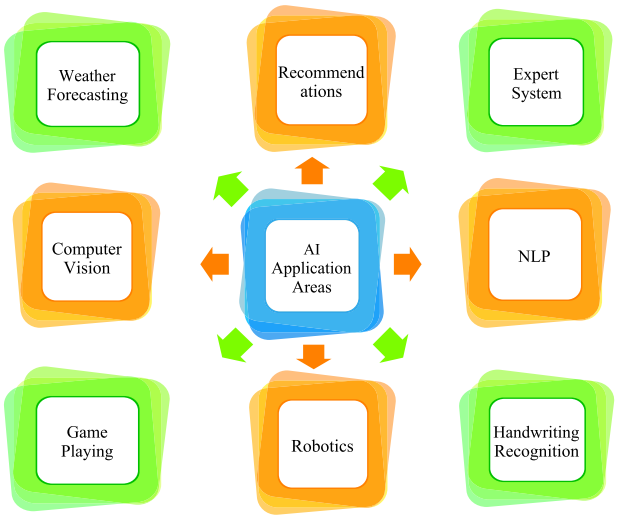

FIGURE 6. Artificial intelligence applications.

Google navigators, and others are shown in FIGURE 6) are AI-based, which require a massive amount of data for accurate decisions and recommendations. This can lead to data security and privacy loss.

One of the procedures for preserving data privacy is *data-anonymization*. It removes personal information from the training data sets and is not a guaranteed procedure. Various privacy-preserving AI techniques are described as follows [66]:

- *Federated Learning*: It is an open-source framework to train AI models on data that we do not have access to it [67].
- *Differential Privacy*: It is a system that shares dataset information publicly without disclosing the individuals personal information [68].
- *Secure Enclaves*: It is a method that provides an isolated execution environment to ensure security guarantees [69]. It assures confidentiality, integrity, and attestation of data.

D. INTEGRATION OF SMART CONTRACT AND AI IN PRIVACY PRESERVING

BC and AI are the giant hot technologies being referred nowadays. AI does the prediction and processing of large datasets, whereas BC provides immutability, security, and decentralized data access [36]. Allowing SC to incorporate AI techniques, makes the SC truly intelligent and will be potentially used in diverse, decentralized applications such as Autonomous cars, Recommendation systems, and AI-based models competitions [70]. AI makes SC self-learning, which adapts itself as per the environmental changes. AI can automate the BC parameters for improved performance [35]. All data (personal and transaction data) over the BC is publicly available; hence AI plays a crucial role in maintaining the privacy and confidentiality of the data. 'Hacking' a BC network is not easy, as an adversary requires higher mining power. AI is useful in developing a highly secure BC application that identifies the attack (by analyzing attack patterns) and necessary action can be taken immediately. In an extreme case, if the attack is unavoidable, then AI can create an isolated environment (secure enclaves) to achieve considerable security. A machine learning algorithm (differential privacy) runs over users devices that do not disclose sensitive data to anyone.

AI can be used to utilize the performance of SCs, that makes the outcome of the analysis is highly trusted and unarguable. Salah *et al.* [37] presented a survey on AI and BC technology to understand how a BC can be useful in AI-based applications and vice-versa. They have discussed current AI problems and how the decentralized scheme can resolve those issues. They also discussed some future research areas and challenges related to BC and AI-based systems. FIGURE 7 states that AI can work effectively with presence of BC platforms. Later, [71] proposed a system called Cortex, which is based on BC technology. They have used AI algorithms to utilize the performance of the BC system. They also introduced an incentive mechanism that can inspire others to submit their optimized scheme for SC and can take away rewards. The goal of the system is to improve the Artificial General Intelligence within itself.

III. RESEARCH QUESTIONS AND SURVEY METHODOLOGY

This section highlights the intended research questions to be identified from the existing literature in the same area and the survey methodology followed fo achieve the same.

A. RESEARCH QUESTIONS

The research questions addressed by this study are:

RQ1. What is smart contract?
RQ2. What are the different platforms for the smart contract?
RQ3. What are the different security vulnerabilities in the smart contract?
RQ4. What are the different decentralized Artificial Intelligence platforms?
RQ5. How smart contract can be framed for a smart application?
RQ6. what is the complexity of programming language used for developing smart contracts?
RQ7. How to standardize the smart contracts to ensure their ease of use?

With respect to RQ1 and RQ2, we may be a concern with the detail description of SCs (from creation to deployment) and the comparison of various platforms like Ethereum, hyper ledger, Corda, etc. RQ3 is concerned with the security risks and vulnerabilities associated with the SC programming code. Then, RQ4 focuses on the decentralized AI platforms for SCs designed by different organizations based on specific applications. Finally, RQ5 is concern with the implementation of SC in certain business applications in the form of

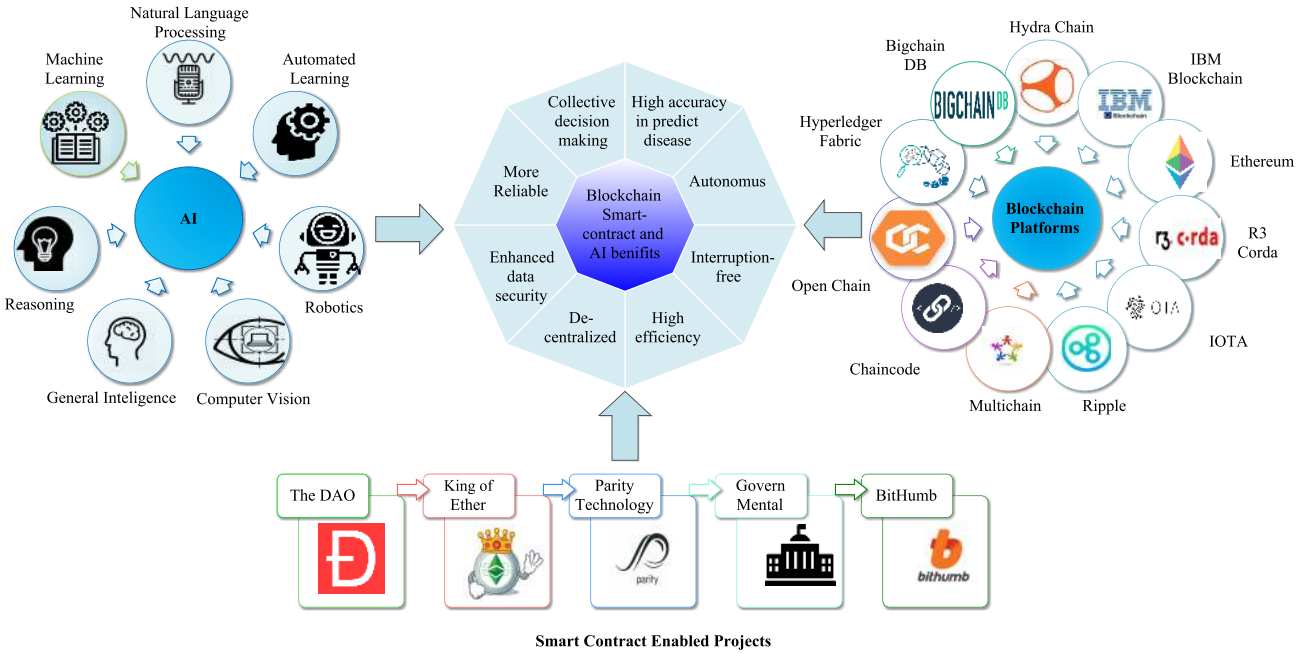

FIGURE 7. Integration of blockchain and AI [37].

TABLE 5. Research questions coverage.

Research Questions	Sections
RQ1	I, II
RQ2	II
RQ3	II, IV, V, IX
RQ4	VII
RQ5	II, VIII
RQ6	VIII, IX
RQ7	IX

a case study. These research questions are covered in the sections mentioned in Table 5.

B. SURVEY METHODOLOGY

In this subsection, we presented the methodology used to conduct this study, such as, search approach and criteria for the final set of selected papers.

We used standard peer-reviewed journal databases (such as IEEEXplore, Science Direct, ACM Digital Library, and Springer link) for existing literature search, using keywords such as "Artificial Intelligence in Smart Contract," "Artificial Intelligence in Blockchain" AND "Artificial Intelligence-enabled Smart Contracts." In the initial phase, publications emphasis mainly on vulnerabilities and security issues in SC (e.g., re-entrancy and ether lost). Then, in the next phase of the search, we concentrated on an Artificial Intelligence based solution for designing SCs.

Further, we conducted the search procedure with various journals, conferences, and magazines dedicated to the parent field, BC SC, security and privacy, and AI. Based on that, we found 238 publications. Then, we reviewed the different sections of the articles like title, abstract, conclusion, and introduction. Then we categorized these publications as "relevant" or "non-relevant" to SC Privacy Protection Using AI Techniques: Tools, Techniques, and Challenges. The comparison of recent trends and research evolutions over the years is shown in Table 2.

IV. SMART CONTRACT VULNERABILITIES AND TESTING

SC is a software code that has the possibility of bugs or security vulnerabilities in it. It can be easily traced and patched by the team of developers. But, in the case of SCs, it is not feasible due to its immutable characteristic. Once it is deployed in the BC, it cannot be altered or modified. The Vulnerabilities in SC can lead to monetary and privacy loss. So, SC developers need to be careful in the initial as well as all design phases of the SC development [72]. This section is bifurcated into (i) logical vulnerabilities (LV), (ii) other vulnerabilities (OV), and (iii) testing vulnerabilities [73]. In (i) and (ii), we explain some of the possible security vulnerabilities of SC and (iii) highlights the various testing procedures, which detect these vulnerabilities. The explanation of these types is described in the subsequent Subsections IV-A, IV-B, and IV-C. The taxonomy of SC security vulnerabilities and their testing procedures is shown in FIGURE 8.

A. LOGICAL VULNERABILITIES

LVs are the semantic errors (logical errors) in the programming code. It is hard to detect, as software code does not generate any kind of error. But, the result produced by the code is not appropriate. It is further sub-divided into various sub-taxonomies and the details about them is discussed in the following subsections.

1) TRANSACTION ORDERING

It occurs when multiple dependent transactions of the same block execute the same SC. This causes transaction

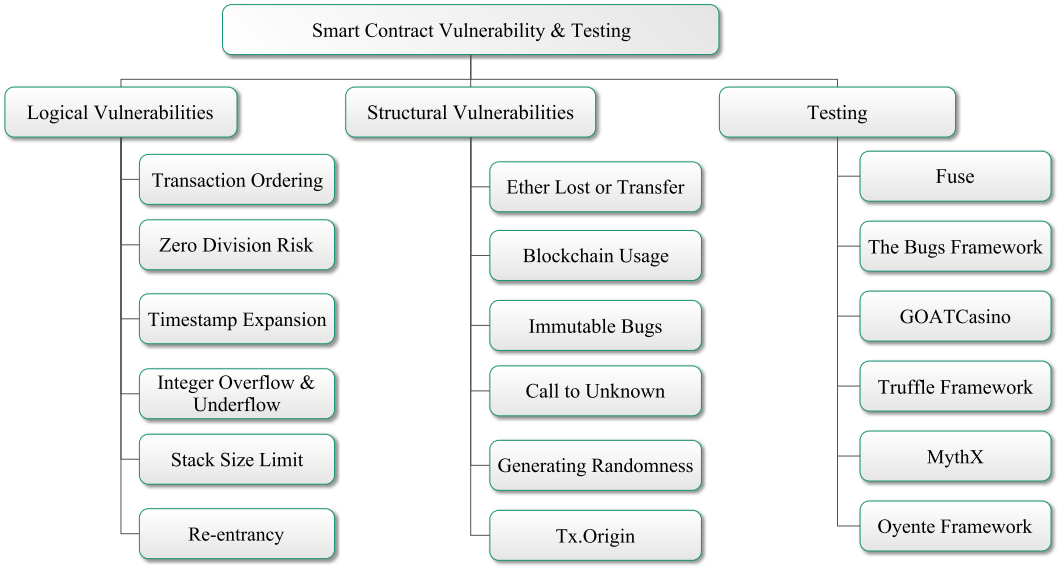

FIGURE 8. Taxonomy of smart contract security vulnerabilities and testing procedures.

concurrency issues, as the next state in the chain of block depends upon the transaction order sequence. In this case, malicious nodes can launch an attack if the transaction ordering is not correct or pending [74].

2) ZERO DIVISION RISK

It is a logical risk where a program must throw an exception if a number is divided by zero. But, in Solidity 0.4 version, divide by zero is not an exception, instead generate the output as zero. This is a development risk and needs to be very careful in writing SCs [74]. Later, this problem was resolved in the Solidity 0.4.15 version.

3) TIMESTAMP EXPANSION

In Ethereum SC, a miner can choose arbitrary timestamp with certain tolerance value while creating a new block. This random timestamp can expose SC to attack. An attacker can modify the timestamp within the stipulated tolerance value to affect the output of the system.

4) INTEGER OVERFLOW AND UNDERFLOW

Overflow and underflow defined as a number reach to maximum and minimum values, respectively. Solidity language can handle the maximum size of the integer number is 256-bits. If a number exceeds 256-bits, then overflow exception occurs. OpenZeppelin's SafeMath library can be used to mitigate such attacks.

5) STACK SIZE LIMIT

In the BC process, each SC can invoke either itself or other SCs using a function. A call stack is associated with the function for the purpose of backward tracing of calling sequence and is capable to keep 1024 at maximum. A BC is growing with the rate of miners capability in solving a reverse hash problem. This increases the SC function calls and may exceed the bounded limit, i.e., 1024. If limit exceeds, then exceptions will be generated. This needs to be taken care of at the developer end and manage the call stack accordingly.

6) RE-ENTRANCY

In this vulnerability, a malicious party calls the vulnerable function of the SC again and again before the previous call completed. This can lead to unexpected behavior of the program. This vulnerability is highly threatening in financial exchanges.

B. STRUCTURAL VULNERABILITIES

These are the SC security vulnerabilities that are other than the logical errors. The detailed description of such vulnerabilities is discussed in the following subsections.

1) ETHER LOST OR TRANSFER

In Ethereum BC, if one party wants to send ether to other parties, then it has to specify the recipient address. SC developers must ensure the recipient address is correct and associated with the SC [74]. If the address is not associated with any SC, then it is known as an orphan address. It party will send ether to the orphan address, then there is no way to recover or trace it, which causes ether loss.

2) BLOCKHASH USAGE

It has a similar issue as with block timestamps. The hash of a block is stored into the current as well as in the next block of the BC. Nowadays, high computing systems are available by which miners can solve the PoW problem (computing reverse hash) quickly and efficiently and accurately [74]. So, block hash can easily be retracted and susceptible to manipulation attacks.

3) IMMUTABLE BUGS

SCs are published in the BC; it possesses the same characteristics as BC, such as immutability. The data or code written into the SC cannot be either altered or modified. If any bug or vulnerability is identified, then there is no way to fix it. So, programmers need to be very careful while designing the SC codes [75].

4) ALL DATA IS PUBLIC

In public BC, the complete data is available to all participating members. Anyone who will solve the PoW can join the BC and get access to the ledger. This is one of the causes of privacy breaches.

5) GENERATING RANDOMNESS

This type of vulnerability is generally found in games or gambling, where the winners are selected randomly. A random number is generated from the blocks private information such as block number, block hash, and timestamp. In this, any malicious miner can modify these block variables and make itself winner [76].

6) TX.ORIGIN

This type of vulnerability occurs when SC uses tx.origin for user authorization. It is a global variable, which initiates the transaction and confirms the owner of the SC. This can be possibly compromised by the phishing attack.

C. TESTING TECHNIQUES

Various authors and organizations across the globe have given testing platforms for SCs.

1) FUSE FRAMEWORK

It is a system to test the SC vulnerabilities using a fuzz test generation module. It works over the SC preparation phase and generates different test cases. It is capable of detecting gasless send, exception disorder, re-entrancy, timestamp dependency, block hash dependency, freezing ether, and dangerous delegate call vulnerabilities with true positive rates between 96-100%. To detect the vulnerability, an execution scenario must pass the test scenario encoding module, then further pass to the test report generation module. Then, an encrypted test report will be generated by the system and send it back to the developer for fixing up the vulnerabilities [77].

2) THE BUGS FRAMEWORK

This framework classifies the SC security vulnerabilities precisely and accurately using the National Institute of Standards and Technologies Bugs (NIST'S) Framework [78]. It has a big repository of attributes used for bug classification and their related properties. It analyzes and standardized the way of categorizing BC and SC system vulnerabilities.

3) GOATCasino

GOATCasino aims to exploit and deliver a variety of security vulnerabilities to the SC developers so that they can avoid such loopholes for other SCs designing. It is a Truffle project that deploys vulnerable SCs to the testing network. It fully exploits the vulnerabilities with proof of concept (how and why vulnerabilities exist) [79].

4) TRUFFLE FRAMEWORK

It is the most acceptable framework for the Ethereum platform based on *Node.js* framework. It is the modular framework, which allows developer to choose only the required functionality. It has many other features like library linking, binary management, scriptable framework, and everything to prevent failures in SC [80]. It is widely adopted IDE in the Ethereum community. It has many features mentioned below, which allows developers to use this particular framework.

- Support both web as well as console apps.
- Create and test new SCs.
- Support various programming and web platforms like JavaScript, CoffeeScript, SASS, ES6, and JSX built-in.
- Self-verification and testing of SCs with Mocha and Chai

5) MythX

It is a security analysis tool for Ethereum SCs used in Truffle, Embark, and Remix platforms. This project was started in 2018 with fully funded by ConsenSys. The step-by-step working of MythX tool is as follows:

- Clients need to submit bytecode and source code to the MythX service analyzer.
- Analyzer then forwards the clients input the microservices.
- Results will be evaluated and then prepare a response sheet for detected security vulnerabilities along with their line numbers.

6) OYENTE FRAMEWORK

It is a SC security analysis tool designed by the scientists from the National University of Singapore, which detects explicitly four types of logical vulnerabilities such as call stack risk, re-entrancy risk, transaction order risk, and timestamp risk [81]. It is a symbolic tool that directly works with EVM bytecode. This framework is unable to locate risks, which are part of a specific function [82].

V. ATTACKS ON SMART CONTRACT VULNERABILITIES

In this section, we discuss few of the real-life cases when SC vulnerabilities were exploited. FIGURE 9 shows few of the attacks.

A. THE DAO ATTACK

The term DAO refers to "Decentralized Autonomous Organization", the first decentralized BC technology-based

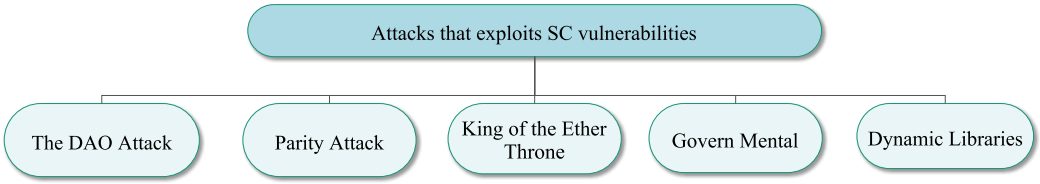

FIGURE 9. Real-life attacks that exploited the smart contract vulnerabilities.

organization (SC). It aims to automate the policies and rules of the organization, which eliminates the need for exhaustive paperwork and trusted TTS. It is not owned by any single person, rather it is software code running on Ethereum BC network. How this DAO works? The steps are as follows: (i) A group of people design the SCs, that run the organization, (ii) Crowdsale: people started purchasing tokens from DAO (represents ownership) by adding funds to it. It is also known as the initial coin offering phase, (iii) DAO starts to operate after receiving sufficient funds, and (iv) People can send proposals to the DAO, which represents how to spend funds, and the members can vote to approve it [83].

The DAO was launched in early 2016 and managed to raise $150 million after the funding period with >11000 members. In mid-June 2016, attackers targeted the vulnerabilities of DAO system and created their "*child DAO (C-DAO)*" system. Then, attackers managed to transfer 3.6 million ($70 million) ether into C-DAO. This incident brings the ether price down to under $13 (originally was $20). The DAO hack was not due to the problem in Ethereum BC; it was due to loopholes in the coding [84].

B. PARITY ATTACK

The Parity Technologies is an organization that develops platforms and applications for the Ethereum network. It allows users to store and manage the BC in they own systems. In mid-2017, an attacker froze multiple accounts, which are being managed by the parity technologies (standard and non-multi-signature application wallets). An intelligent attacker exploited the wallet vulnerability and made himself or herself as the owner of SC library. This attacker has selflessly destroyed the SC and froze multiple wallet accounts. The estimated amount of frozen ether coins was 513,774.16 ether ($162 million) [20]. It was witnessed as the second most largest hack in the history of Ethereum, in terms of ETH stolen. This attack was known as *parity wallet* attack. To date, the parity hack compromised approx. 154K ethers, which costs around $155 million (i.e., 1% of the total Ethereum value) [85]. In this, an attacker executed two transactions- (i) to get the complete ownership of multi-signature and (ii) to locomote all its funds.

In response to the attack, Parity Technologies has refined their development process as follows:

- Live alteration to the contract code.
- Reanalyze the parity wallet at high level.
- Increase the number of ownerships.

C. KING OF THE ETHER THRONE

The King of the Ether Throne SC was deployed in 2016, i.e., a multi-player SC game where players are competing among themselves to earn the title of "king of the Ether". If anyone wants to become a king, then he/she must pay some ether to the existing king (increased cost) along with a small fee to the SC. Upon becoming the king, a player will be immortal. If no player purchases a throne from the current king to be crowned as king within 14 days, then the game will be started all over again. If the successor is not having sufficient amount to become a king, then the payment to the dethroned king was sent to the contract-based wallet. In this case, a new king can never be crowned and SC becomes stuck. This will come under the category of a denial-of-service attack.

The catastrophic flaw of King of the Ether Throne SC was the use of *address.send()* and fail to check exception upon unsuccessful call. This can expose SC to the re-entrancy attack and can be protected by involving *contract-based wallet*, which requires more gas than contract account.

D. GOVERN MENTAL

It is a flaw in ponzi game in which players can join by paying a specific amount of ether to the SC. If no one joins the game for the next 12 hours, then the last participant can claim to get all the ether in the contract by paying a small amount of fee to the SC. It records the participants and funds details in two different arrays. These arrays will be reseted at the and of games. It exploits the security vulnerabilities like exception disorder and stack size limit of the SC and become the owner. Its goal is to keep all ether with himself (not pay to winner) and use it for later transactions. In this particular attack, an attacker can also change the timestamp of the last joined block to keep the amount with itself [75].

E. BITHUMB

It is a cryptocurrency exchange system, and in the year 2017, they identified a data breach. Attackers managed to stole users personal data and money. Attackers were succeeded to grab the personal information of around 32K users including name, contact number, and email id. This attack was not occurred directly on the Bithumb exchange network, instead of on employees personal computers. An attacker claimed as a Bithumb executive and to the victims for their identification number in the form of OTP. These credentials can be used for voice phishing on behalf of Bithumb representative [86]. After the attack, the Bithumb exchange announced

the compensation to the victims against their personal information stolen.

VI. TRADITIONAL SECURITY SCHEMES FOR SMART CONTRACT

This section discusses the traditional security schemes for SC, such as Trusted Execution Environment (TEE), Secure Multi-Party Computation (SMPC), and Zero-Knowledge Proof (ZKP). The detailed explanation of these schemes is given in the following subsections.

A. TRUSTED EXECUTION ENVIRONMENT

It is an independent, isolated process, and temper proof execution environment, where the applications can be executed securely without affecting the system performance. Being in an infancy stage, it lacks some design features but still fulfilling the requirement of security for any system. Confidentiality of design code, run-time state data, the authenticity of executed code, and the integrity of hardware states (for example registers, stack, and memory) are also maintained. It opposes all kinds of software attacks and physical main memory attacks. A third party must attest its trustworthiness in order to use the TEE. Sabt *et al.* [87] gave a definition of TEE, which stated that any untrusted code could not execute any action within it. It is developed over Intel Software Guard Extensions (SGX), which does not always provide system availability. In academics, the ample number of TEE prototypes exist, such as Genode TEE (Genode Labs), Open TEE(Intel Collaborative Research Institute for Secure Computing), Andix OS (TU Graz University of Technology), SafeG (Nagoya University), and TLR (Microsoft Research).

TEE is aimed to ensure the confidentiality of the data, but cannot prevent data loss caused by side-channel attacks. This causes the delay in communication between enclaves (Intel SGX based CPU) and CPU components. Enclaves are specifically used for general-purpose computations. Both BC and TEE can be used to overcome the issues of delay and side-channel attack. Cheng *et al.* [88] proposed a BC-based TEE system called *Ekiden*. This system separates the consensus mechanism from the execution environment, which enables efficient TEE-based secure SCs with high scalability. This system was able to achieve 600 times more throughput and 400 times low latency than the Ethereum system. This system classifies the data based on their spatial and temporal properties. However, they have provided a system that overcomes challenges in BC SC.

The development of trusted applications for ordinary developers is quite challenging for developers. To get access of TEE hardware device for anyone is difficult or impossible without the vendor support. Authors in [89] proposed a software-based TEE platform to debug and develop trusted applications without any hardware support. This system, called *Open-TEE*, is an open-source system, i.e., available for everyone. When a developer completes their debugging process over Open-TEE, they can use the actual hardware interface for implementation. It aims to make the system

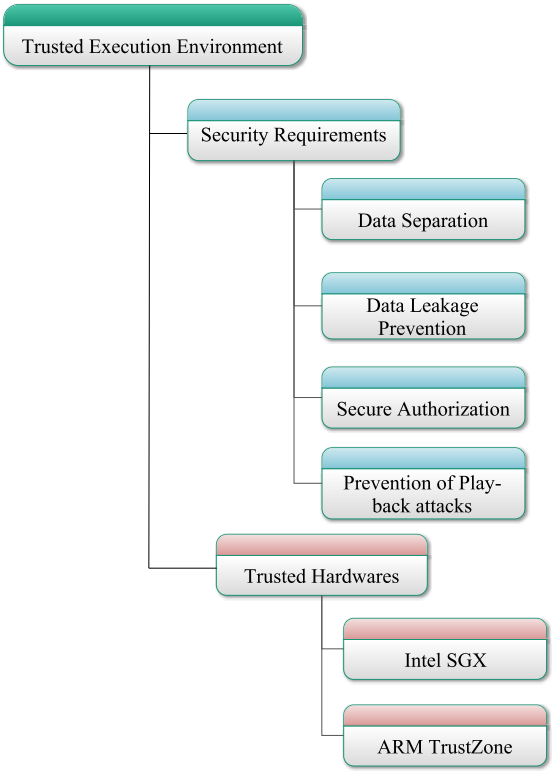

FIGURE 10. TEE security requirements for SC.

efficient and consume less memory. Also, it can be easily deployed and configured without the need for any extra package on the target system, and also independent of the TEE hardware environment.

There exists various security requirements for SC-based systems, which are shown in FIGURE 10 and described as follows:

1) *Data Separation:* It means that one partition of memory has no information about the data stored in another partition. The domains of the system environment are used to protect data files from malicious users. Such domains would be *current work environment* (CWE) used to store encrypted partitioned file parts and *secure work environment* (SWE) used to keep decryption key [90].
2) *Data Leakage Prevention:* Data leakage is the illegal sharing of information with unauthorized TTS. It can be due to accidental breach, intentionally done by an employee, and communication under Sybil attack. This must be prevented; otherwise, it will create many obstacles in organization growth. It can be prevented with low privileges to data access, security policies, and security tools.
3) *Secure Authorization:* For authentication, a unique PIN is required to validate the trusted user. To establish a communication with server, the public key of server is used to encrypt the data, which can only be decrypted with the private key of the server. If an attacker tries

to perform a man-in-the-middle attack, then he will not be able to decrypt the data packets. Since it responds with an authentication packet with a valid signature and secured with a random key so, the adversary cannot generate a response message to communicate with the client [91].

4) *Prevention of Play-back Attacks:* As per the scheme defined by Fan *et al.* [90], a data packet is used to communicate with the server only once per session, after that it is removed. Moreover, a malicious user can only capture one packet with a single communication key. In order to access the encrypted packet, its private key is needed, which is not publicly available (only available with server). This scheme ensures the prevention of Playback or Replay attacks.

Isolation of TEE with Rich Execution Environment (REE) is not enough for the system to be secure, a communication channel between TEE and REE can be also be attacked. However, TEE security solutions are suitable for static components/devices, but not for communication channels. To resolve such issue, Jang *et al.* [93] proposed *SeCReT* scheme, which provides a secure communication channel between REE and TEE. It ensures the processes, which are under unsafe regions can use session keys for encryption. Session keys are only provided to the process when the requesters integrity and authenticity is verified. *SeCReT* system flushes the session key from memory once the processor switched to the kernel mode, and it will prevent an attacker from accessing the session key. They also have done the security analysis and performance evaluation of their proposed system with kernel root-kit and LMBench micro-benchmark, respectively.

The different Trusted Hardware can be used for TEE to ensure the privacy of SCs are discussed as follows:

1) *Intel Software Guard Extension (SGX):* TEE implementation on any system needs hardware support, which is provided by Intel manufactured Software Guard Extension (SGX). SGX is only available to a few people for research purposes. Due to the unavailability of the hardware platform, researchers who do not have the privilege to access the hardware cannot contribute to its development. Intel® SGX (TEE-based) is the perfect solution for a decentralized BC-based cloud ecosystem. It allows the execution of code and data such as SC in its protected area called enclaves. It provides better data protection and data-in-transit.
TEE development is rapidly growing, but the software technologies are lagging, which is due to the unavailability of hardware support. To address these issues, Jain *et al.* [94] introduced OpenSGX, which emulates the Intel SGX hardware components at the instruction level and also includes system software to perform full exploration of TEE. It is publicly available for research purposes and derives proof-of-concept for TEE environment. For an interactive environment, it is necessary to build a secure communication between the users and enclaves. For that, Intel-based hardware are available to use: Intel Protected Audio and Video (Intel PAVP) and Intel Identity Protection Technology (Intel IPT) are used to create a secure encrypted channel for IO communication with enclaves. It creates a hidden communication channel using bot-nets. Although Intel SGX is secure and maintains SC confidentiality, yet malicious intent of vendor or attacker can expose critical information [97].
The iExec is good at building BC-enabled decentralized applications on decentralized nodes with Intel® SGX. It ensures the data protection, end-to-end privacy, and validity of results [98]. Characteristics of the iExec system are: open, results are based on SCONE platform, compatible with BC technology, and 100% compatible with TEE.

2) *ARM Trustzone:* It is another security extension similar to Intel SGX, which provides secure and TEE. It splits the resources into two execution environments: normal execution world (NEW) and secure execution world (SEW). It allows the execution of SC in SEW and ensures its confidentiality in the BC network. Ngabonziza *et al.* [95] discussed the architectures of ARM, which supports TrustZone technology and how the technology can be implemented over the hardware and software. Hua *et al.* [99] discussed the Virtualization of TrustZone technology. TrustZone assures the NEW cannot access the secure part of the memory, while the SEW can access the entire memory [99]. TrustZone is widely used over mobile platforms and microcontroller architectures. TrustZone based applications are discussed as follows:
   - *Storage Protection:* Due to the isolation property of TrustZone, it is an ideal option to store sensitive information such as SC, encryption keys, and passwords. This approach can effectively prevent data exposure attacks like heart-bleed or buffer over-read attack. Rubinov *et al.* [96] proposed an approach that automates the partitioning of critical Android applications into NEW and SEW. The client code will be executed in NEW, while the TEE commands and confidential data will be in SEW. Their proposed approach can be useful for maintaining BC-enabled SC privacy.
   - *Enhanced Rich Execution Environment security:* TEE can also be used to enhance the security of the Rich Execution Environment (REE) using SEW. Azad *et al.* [92] proposed TZ-RKP (TrustZone Real-time Kernel Protection), which provides security to the NEW using SEW. This approach is recently deployed in Samsung Galaxy series smartphones and tablets. Ge *et al.* [100] presented *SPROBES* for introspection of ARM TrustZone hardware. It is a mechanism protected by TZ (TrustZone). Whenever any *SPROBE* is executed, it receives an immutable trap, which can detect

TABLE 6. Trusted execution environment.

Author	Year	Description	Merits	Demerits	1	2	3	4
Azab *et al.* [92]	2014	Proposed TZ-RKP (Real-time kernel Protection), for mobile platforms	It can effectively prevents attacks against kernel binaries	-	Model	X	X	X
Sabt *et al.* [87]	2015	Proposed a precise definition of TEE and analysis of its core properties	A redefined definition of TEE for better understanding of its features	Effects of major attacks on TEE like side channel are not mentioned	Survey	X	✓	✓
McGillion *et al.* [89]	2015	Proposed an open-source TEE software to debug trusted applications	Researchers can use all functionalities of actual hardware TEE	Performance tuning cannot be achieved	Model	X	X	✓
Jang *et al.* [93]	2015	Proposed SeCReT, which builds a secure communication channel between TEE and REE	Protect the session encryption key from being exposed	-	Model	X	X	
Jain *et al.* [94]	2016	Proposed an open source platform for TEE research, OpenSGX.	Provides an open platform to be used as SGX research	There is no research platform to explore all the potential risks of hardware attacks	Model	X	✓	✓
Ngabonziza *et al.* [95]	2016	Discussed different ARM architectures, which supports TrustZone	This technology is used to secure the mobile platforms	-	Survey	X	✓	✓
Rubinov *et al.* [96]	2016	Proposed an automated partitioning method for android applications into normal and secure world	It provides efficient partitioning of the real world applications	It works with code fragments of applications, which may cause in leakage of source code	Model	X	✓	✓
Cheng *et al.* [88]	2018	Proposed Ekiden by combining TEE with BC	It reduces the privacy gaps of previous schemes	Privacy protection for BC platform requires great threat models, which are not discussed	Model	✓	X	✓
Fan *et al.* [90]	2018	Proposed a secure access scheme for mobile devices	Privacy is ensured with file slices and verification of operations	Not efficient to access multiple files at the same time	Model	X	X	X

Parameters- 1: BC, 2: Case Study, 3: Cryptography
Considered: ✓, Not considered: *X*

kernel root-kit attacks. This approach can be used to improve the security of Server operating system and applications.

Table 6 provides a detailed comparison of existing approaches of TEE with reference to parameters such as -BC, case study, cryptography, pros, and cons of the existing approaches.

B. SECURE MULTI-PARTY COMPUTATION

SMPC is a cryptographic scheme, which ensures input privacy and corrects the computations. Given n participants as P_1,P_2,P_3,...,P_n and each of them having private data as d_1,d_2,...,d_n respectively, participants will keep their own inputs as private and compute the public function $F(d_1,d_2,\ldots,d_n)$ using the private data of individuals [101]. It enables multiple parties to jointly compute a function over inputs without revealing their inputs. It is a multiparty computation (MPC). MPC in BC helps to distribute the private keys among servers and nodes for BC operation and gives cryptographic assurance in BC workflow [102].

It allows the participants to perform arbitrary computations along with new inputs, which will be provided to participants. These inputs will remain private and the result of computation is assured to be correct. Maurer [103] presented a simple approach to secure MPC with less complicated security proofs. Due to its simplicity, it can be well-suited for educational purposes. Zhao *et al.* [104] presented a survey on the theoretical and practical aspects of SMPC protocols.

To prove that the protocol is secure, researchers have provided various definitions of security, which ensures the following requirements:

- *Privacy:* No party is allowed to access the information other than its own output.
- *Correctness:* The output obtained by the participant should be mathematically and technically correct.
- *Independence of Input:* Input taken by an honest participant must be unique and independent of the input taken by a corrupted participant.
- *Guarantee of Output:* There should not be any interference by an adversary in obtaining the output of an honest participant.
- *Fairness:* The adversary should receive their own output if and only if the legitimate participant obtained their own output.

MPC has a fundamental issue of security, and it can be resolved by using secret sharing in an efficient way. The secret data packet is partitioned into multiple pieces and distributed among gatherings and further recovered by making some specific arrangements of gatherings. SMPC can be performed in two ways: centralized and decentralized methods. In the centralized method, there is always a risk of single-point failure or central authority misbehaving. To address this issue, Patel [105] proposed a decentralized system to ensure users privacy. It also reduces the communication and computation cost compared to the centralized system. In order to ensure the privacy of aggregated data, Lapets *et al.* [106] suggested the use of MPC protocol over the web environment. Depending upon MPC protocol, aggregate data will be computed, while the confidentiality of the contributor's credentials is protected simultaneously. The SMPC is only utilized if more parties participate and use it. Using SMPC over web service should satisfy some goals like better adoption (comprehensibility), code available open-source (transparency), no special hardware or software requirements (easy-deployment), valid input checking, among others [110]. Some authors have used SMPC to integrate with hybrid networks [111], and others for vulnerability analysis [112].

TABLE 7. Secure multi-party computation.

Author	Year	Description	Merits	Demerits	1	2	3	4	5	6
Patel *et al.* [105]	2016	Proposed a method to execute SMPC in a distributed environment	Low communication and computation cost	No practical explanation for privacy protection	Model	X	X	X	X	X
Lapets *et al.* [106]	2016	Proposed a web-based MPC service to compute the statistics	Effective Performance	Compatibility with platform and human error	Model	X	X	X	X	X
Marwan *et al.* [107]	2016	Proposed a privacy-preserving collaborative environment for healthcare field	Ensured Privacy	Encryption key management	Model	X	✓	✓	✓	✓
Kaur *et al.* [108]	2018	Proposed a model to ensure privacy of recommendation generator	Better accuracy and security	High computation time for offline model generation	Model	X	✓	✓	X	X
Ma *et al.* [109]	2018	Proposed a model for multi-party deep learning for cloud computing	Privacy and accuracy	-	Model	X	✓	X	✓	X
Feng *et al.* [101]	2019	Discussed Privacy issues with BC and existing defense mechanism	Privacy preserving mechanisms	-	Survey	✓	✓	X	X	✓
Zhao *et al.* [104]	2019	Discussion over theoretical and practical aspects along with security requirements	Systematic overview on every aspects of SMPC	No enough schemes for cloud assisted SMPC mentioned	Survey	X	✓	X	✓	X

Parameters- 1: Type of paper, 2: BC concept, 3: Cryptography techniques, 4: Healthcare application, 5: Cloud concept, 6: Research challenges.
Considered: ✓, Not considered: *X*

Table 7 gives the detailed comparison of existing approaches in Secure Multi-Party Computation with reference to parameters such as- BC, cryptography, healthcare, cloud, Research challenges, pros, and cons of the existing approaches.

C. ZERO-KNOWLEDGE PROOF

ZKP is used when one person has to provide evidence of knowing a secret without actually revealing it to another person. In this method, the person who has to prove his/her knowledge is called a *prover* and another person who verifies the knowledge is called a *verifier*. The usual method of proving the secret knowledge is to tell the secret to the *verifier*. However, the privacy of the secret is violated in the usual method. To resolve the issue, ZKP enables the *prover* to convince the *verifier*, that the *prover* has some secret knowledge without revealing its secret. This helps *verifier* to identify that the *prover* is a legit person. *Prover* can answer the questions correctly to *verifier* and if *prover* does not know the secret, he/she may cheat and there is 50% chance to success in cheating the *verifier*. That is why ZKP is an iterative method to verify the identity of *prover*. With repetitive steps for many iterations, the chances of *prover* to cheat successfully can be decreased.

In public BC like Ethereum, the transactions are validated with sender and receiver addresses along with their input and output values. Using Zcash and ZKP, the validation of transactions can be proved even without sharing any critical information such as addresses and I/O values. The Aztec development team ensures the Aztec protocol can perform confidential transactions for Ethereums digital assets using SC with ZKP. It uses homomorphic encryption technique to secure transactions.

ZKPs can be used in Proof of Identity (PoI) to verify the identity of a person [113]. The conventional schemes of identification are private key and password or four-digit pin, which are not secure as they can be easily guessed by the brute-force method. The new methods of identification are bio-metric fingerprints, iris scan, and facial recognition, which include body parts as an identification, but it has its own disadvantages [114]. Digital signatures and public-key cryptography are also other identity verification schemes.

Zero Knowledge Proof has three characteristics as follows [61]:

- *Completeness:* The completeness of ZKP is ensured when the statement is true, and both prover and verifier follow the same protocol, and the verifier is convinced. In other words, if both the parties are honest with each other, then the prover can pass the proof.
- *Soundness:* When the prover does not know the secret, the chances of passing the proof is negligible. It means there is no way of cheating successfully, and the verifier is not convinced.
- *Zero-Knowledge:* If the prover truly knows the secret and he/she can give the answers correctly, It is ensured that the verifier has no knowledge about the secret.

The conventional ZKPs have many iterations, which can affect the computational and time complexity of an application. To address this issue Almuhammadi and Neuman [113] proposed a new approach to reduce the computational and communicational costs by reducing the number of iterations to one. The new approach is called as One-Round Zero-Knowledge Proofs and much improved in terms of execution time, communication cost, bits transfer, and latency compared to other techniques. Both one round and multiple-round ZKPs are powerful and efficient for privacy-preserving. To choose one round ZKP or multiple-round ZKP all depends upon the type of application used. Zaw *et al.* [115] showed the SSL connection can be more secured with ZKPs. By integrating ZKPs with SSL protocol, it will be difficult or impossible for an attacker to forge the certificate of identity. With this integration, no intruder can invade the privacy of the client or server.

ZKP is an effective cryptographic scheme and can be used as an authentication system for many network devices and applications. In [116], [117], authors have discussed what the conventional IoT systems are facing (security and privacy issues) and how to resolve them with the integration of ZKP. The different applications of ZKP include Quantum computations and quantum cryptography is further discussed

in [118] [119] [120]. ZKP has the benefit of not revealing the users personal information. It has the limitation that, for every new property, both prover and verifier go through a costly and time-consuming trust establishment procedure. It can become a barrier to the adoption of ZKPs. This can be efficient with the integration of distributed BC technology. It allows the network to participate in computation without exposing the private data.

An e-commerce application can be developed over the BC network where all data are public, and the transaction details are not protected due to the public disclosure of data. Li *et al.* [121] proposed a method known as RZKPB, which aimed to hide the sensitive transaction details. It does not store any financial transaction details and helps BC to ensure transaction privacy. The authors experimented RZKPB on e-commerce application as an example of sharing economy. The results of their experiments showed that the new method is more efficient than the traditional methods of privacy-preserving BC. Another method for privacy-preserving of permissioned BC for sharing economy is proposed in [121], where authors have proposed a Fast and Privacy-preservingg method based on the Permissioned BC (FPPB) for honest transactions in sharing economy. With the use of cryptographic schemes, such as ZKP and stealth address, FPPB can protect the privacy of financial transactions and can ensure uniform transaction contents for honest trading. Experiments over FPPB showed that compared with conventional transactions without cryptographic schemes, FPPB can slightly slow down the transaction with changes in different parameters.

VII. ARTIFICIAL INTELLIGENCE IN SMART CONTRACT

AI is used to make systems intelligent that can imitate as humans or even better than humans [122]. The amalgamation of AI (offers big data processing [123]) and BC (offers security) technologies is quite powerful and is improving industrial automation and analysis [124], [125]. It can ameliorate everything in almost all types of industries, such as agriculture, healthcare, media, and financial industries [126]. SC is the main driver of the BC, which facilitates the self-execution and self-enforcement of digital contracts. Current SCs have limited computing capability and mismanaged governance. These limitations are imposing a barrier to SC in implementing real-world problems [127].

Integration of AI with SC is done through the policies and rules of the BC. AI understands the complex rules and policies of the BC and analyze how AI can be used to create and execute complex SCs. This process makes the SC effective and self-learnable. It helps to analyze the already existing SCs and identify those factors, which were not incorporated earlier and extend it into future SCs.

A. PRIVACY-PRESERVING ARTIFICIAL INTELLIGENCE TECHNIQUES

The BC is decentralized, so there is no central authority for controlling. The development of BC is done by individual programmers in SC. It is less likely to be secure from bugs and loopholes. Marwala and Xing [36] briefed about how artificial intelligence can be used to develop bug-free SC applications. The goal is to achieve BC 2.0, to highlight that various artificial intelligence techniques can be used to increase the performance of BC. They discussed security and privacy issues that BC faces, which can be resolved by artificial intelligence techniques.

BC technology itself is used for providing security to the data stored within it. Intrusion Detection System (IDS) and Intrusion Protection System (IPS) are critical components, which are used to monitor the threats at the application, layer. Swarm Intelligence (SI) techniques are applied to increase the effectiveness of the IDS. SI is the field of artificial intelligence that is based on the laws of behavior governed by agents such as social insects, fish schools, and flocks of birds [128]. Computational Intelligence (CI) is also a branch of artificial intelligence, which plays an essential role in cryptographic systems. The applications of CI are ranged from cryptography, cryptanalysis, hash function computation. The key benefit of using CI in BC is that it can create robust cipher hashes to store data in BC. It can improve the privacy of the data stored and make the system more defensive to attacks [36]. Since BC consists of a huge amount of personal data, data encryption is a major issue regarding user's data privacy. In the current Bitcoin BC system, the elliptic curve public and private key cryptography are being utilized. But no one has developed the public key algorithm, which is devoid of bugs. AI search techniques are being used to address these problems related to public key algorithms [36].

AI techniques are used to analyze complex data to infer the facts and design the intelligent systems of various disciplines. It offers various benefits, such as high computing power, increased storage capabilities, and improved data collection. It helps to predict the risk factors of security measures (both defensive and offensive) and protect the system form security attacks automatically and efficiently. Advancements in AI techniques help systems to protect against new security threats also. This allows developers to create dynamic and self-configurable SCs.

Cryptography is not sufficient for privacy protection. Users' privacy may be at risk due to other factors as well. For example, smart home devices store a variety of information about the user and their family. Li and Zhang [131] discussed how privacy can be endangered by some technologies. The data is stored within it for many years or decades, which can be used either for good or bad purposes also. There is a possibility where sensitive data might get stolen or sold illegally to some technology company, which uses the data for commercial purpose. Electronic data produced by IoT devices, network devices, and mobile devices (geographical coordinates) contains some private information. Privacy is invaded when such private information is collected by some firm for illegal purposes. The geographical coordinates include the daily routine life of an individual, and it provides private information about them. The goal is to secure the

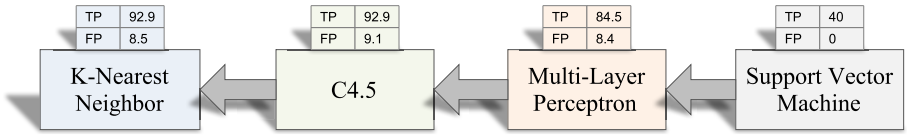

FIGURE 11. Comparison of AI approaches in identification of malicious access points [129].

location-based services for better privacy of location data. Location-based services are protected with many approaches of AI. Following is the model discussed regarding location service privacy:

1) K-ANONYMITY TECHNIQUE

The K-anonymity model is widely used to ensure privacy for location-based data. It provides privacy based on users' preferences with more control [132]. Natesan and Liu [132] proposed a theoretical approach for user-controlled privacy protection using k-anonymity technique. This approach helps users to set their privacy profiles based on the past decisions made by them. This methodology is adaptive to users' obvious changes to their privacy levels. As the model is trained completely, no further user involvement is required to set the privacy profile on per day basis, unless the user wants to change the profile explicitly. K-learning algorithm proposed in the paper [132] for privacy management (user-controlled), which can be used for social-driven location sharing management. Ye *et al.* [133] proposed a location information authentication system based on trusted TTS (''CliqueCloak'' theorem). It prevents the privacy disclose caused by side information. The use of a TTP gives rise to another issue of trusting the third party for sharing sensitive data.

However, the third party can be attacked, and sensitive data can be compromised. To avoid the usage of TTS secure systems, BC SC technology is beneficial. Geng *et al.* [34] introduced an enhanced k-anonymity approach to use for privacy protection with SC. It is a BC-based secure incentive mechanism that motivates users to prefer a k-anonymity algorithm for privacy preservation. The incentive mechanism system must be private and secure enough to store the transaction related details. By participating in a k-anonymous group, they can receive rewards or incentives, which is the key point to motivate users to use this system. The security of the transaction system is maintained by group encryption and blind collective digital signature.

The transaction details stored in BC SCs eliminates the need for third-party dependency for privacy. Data stored in SCs can be visible to anyone as it is in plain text; an attacker can read the contract and track the information of the requester. This issue is addressed by Geng *et al.* [34], they have used public and private contract approach where private contracts can only be accessed by users who are participants of the k-anonymous group. Group encryption is used to keep the public contract secret, and to verify the contract, blind collective DS (digital signature) is used. By introducing this method, it can be possible to secure the location-based data by implementing the k-anonymity algorithm with BC SC for privacy protection. Electronically produced data, including GPS coordinates, are transferred through wireless networks [134], [135], which are vulnerable to access point based and brute force attacks. For using SC, wireless networks should be secure enough to ensure privacy.

The access point (AP) of the network should be protected from data leakage and other attacks to achieve the privacy of the confidential information. The identification of authorized APs and unauthorized APs must be made to protect the data from leakage. Kim *et al.* [129] used machine learning algorithms to differentiate between authorized and unauthorized APs. They concluded that the K-Nearest Neighbor (KNN) gave the highest accuracy in the identification of authorized APs. In [129], authors conducted the comparison of various AI approaches to find out which approach is more effective for finding malicious APs using round-trip time as a parameter. In FIGURE 11, various approaches to AI are compared with their correctness.

The machine learning approaches are discussed as follows:

1) *Support Vector Machine (SVM):* It is a supervised machine learning method. It separates the classes with a hyper-plane, which maximizes the margin between classes. From the given data, the binary linear classification model is developed through which the boundaries of the classification data is derived [129], [136], [137].
2) *C4.5:* It is used to classify the data using a decision tree. C4.5 algorithm is developed based on the ID3 (Iterative Dichotomiser) algorithm. It also makes a decision tree for classifying, but while analyzing continuous data, it does not give assurance of optimal results. C4.5 uses gain ratio for information entropy to get an optimal solution, and it creates a decision criteria to set the boundaries [129]. Boundaries are used to classify the data set more effectively [138].
3) *KNN (k–Nearest Neighbors):* KNN algorithm first takes k nearest training samples as a testing data set, and then it predicts the sample data set with major class among the testing data set [139]. In the selection of k nearest neighbors, KNN needs to compute the distance of all training samples for each test sample. It costs more linear time complexity, which is why it not being used for big data applications. Based on the requirements of researchers, they have designed new methods to improve the efficiency of the conventional KNN algorithm. Deng *et al.* [139] proposed an approach

TABLE 8. AI privacy techniques.

Author	Year	Description	Merits	Demerits	Article Type	Algorithm Used	1	2
Blum *et al.* [129]	2015	Overview of swarm intelligence is discussed.	Optimization of robotics.	Implications over healthcare technology.	Survey	Robotics	*X*	*X*
Geng *et al.* [34]	2017	Proposed a model for K-anonymity and BC.	User incentive mechanism	Results of privacy and efficiency are not discussed.	Model	K-anonymity	✓	✓
Marwala *et al.* [36]	2018	Discussed AI and BC integration issues and benefits	Better understanding of BC and AI integration.	Requires more explanation on defence mechanism	Survey	-	✓	✓
Kim *et al.* [130]	2018	Proposed a method to detect unauthorized access points.	Effective algorithm	Classification parameters not discussed.	Model	KNN	*X*	*X*
Senavirathne *et al.* [131]	2018	Proposed integral privacy based linear regression model.	High accuracy	lags at different parameters configuration.	Model	Linear Regression	*X*	*X*

Parameters: 1-Cryptography Techniques Used, 2-BC
Considered: ✓, Not considered: *X*

for an efficient KNN algorithm. They first performed k-means clustering on the whole data-set to separate the complete data-set in several parts. Then, the nearest sample is used as a training sample, and after that, the classification is done by KNN. The time complexity of the proposed method is linear to the sample data size of medical images and the classification performance is better than the conventional KNN.

Another challenge for the KNN algorithm is that it has the same impact on all features and characteristics during the classification, even if some characteristics are less important. As a result, it may deviate the classification and decrease the efficiency of the algorithm. To address this issue, Kuhkan [140] suggested a new method to use KNN, which was to use a weighted approach for classification. They suggested allocating certain weights to all the characteristics based on their importance. The classification process will take place based on the weights allocated to the characteristics, which will avoid the same impact of all features to the classification and also reduces the deviation of the process. After the comparative study of five classifiers with ten different data-sets, they concluded that there was a considerable amount of improvement in the performance of the KNN algorithm.

4) *MLP (Multilayer Perceptron):* It is a neural network methodology where a hidden layer is added in between the input and output layer. The supervisory learning is performed with backpropagation algorithm, which classifies linearly non-separable data [141] [129].

After performing test cases, Kim *et al.* [129] concluded that KNN has a higher efficiency than the other discussed machine learning methods. Due to the better performance of KNN, it is used in the identification of authorized and unauthorized access points to ensure privacy. However, to ensure the privacy of any system, an important task is to verify whether the security model itself is secure or not. Senavirathne and Torra [130] implemented linear regression approximation with ''integral privacy'' to ensure high robustness and accuracy of ML(Machine Learning) models. In the proposed method, re-sampling-based estimator is used, which constructs the linear regression model. The evaluation of the output models is done by comparison with privacy, accuracy, and robustness. After comparison, the solution based on integral privacy gives a better performance with the given criteria. This method provides better privacy to linear regression models without compromising performance.

2) ARTIFICIAL NEURAL NETWORK

It is a computational model derived from biological neural networks [142]. Its structure and functionality are similar to the nervous system of the human brain. It has the capability to learn any type of data. The information related to public BC transactions is too large and can be efficiently used by ANN [143]. It can do the text categorization and improves the classifier scalability. This empowers the security aspects of SC.

Table 8 provides a detailed comparison of existing approaches in Artificial Intelligence for Privacy with reference to parameters such as -AI Algorithm, cryptography, BC, pros, and cons of the existing approaches.

B. DECENTRALIZED ARTIFICIAL INTELLIGENCE PLATFORMS FOR SMART CONTRACTS

It is an approach to solve complex learning and decision-making problems. It consists of a large number of distributed and automated learning processing nodes. These decentralized AI (DAI) platforms are robust, scalable, and fault tolerant. It does not demand the data must be at single location for processing. Some of the available DAI platforms are as shown in FIGURE 12 and further explained in subsequent subsections.

1) CORTEX

It is the first-ever decentralized AI platform developed by Cortex Labs to support SCs and its execution. AI developers around the world can upload their proposed models to the BC and decentralized application developers can access these models by paying native tokens to the Cortex. It brings AI inference engine directly on BC and inferences the results, which removes the role of the third-party organization such

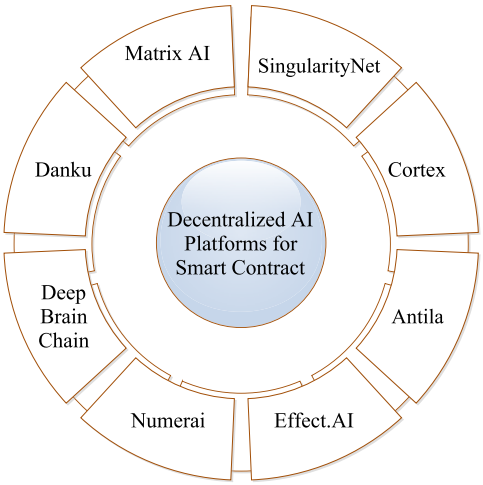

FIGURE 12. DAI platforms for smart contracts.

as Oracle in generating inference results [144]. It can boost AI competitions, AI model developments, and decentralized application development. Ethereum developers are now migrating their SCs to the Cortex BC and add-up some AI functionalities.

2) ANTILA

It is an open-source AI-based intelligent SCs platform (also called Synergetic SCs) programmed by Fetch.AI, UK-based Startup. It facilitates SC with decentralized searching and also helps to optimize their processes and operations. Synergetic SC enables the solution to complex decentralized applications using off-chain computations with N-number of parties [145]. It has various applications such as- taxi network: managing the taxi network in optimizing the journey, shipping industry: matches suppliers with delivery vehicles, and hotel industry: matches visitors with empty rooms.

3) MATRIX AI NETWORK (MAN)

It is a distributed and open-source computing platform with AI algorithms and the BC network. It aims to revamp the efficiency, flexibility, and intelligence of the BC network. It is highly capable to support 50K block transactions per second. To maintain the integrity of the BC network, it uses more than one consensus algorithms with Markov chain [146] and Monte Carlo computations (PoW and proof-of-stake) [147]. Various objectives MAN are-automatic and secured SCs, high-speed transaction execution, and flexible management. The main advantage of MAN is the automatic code generation for SCs, and users are unaware of its code.

4) DEEP BRAIN CHAIN (DBC)

DBC provides the AI platform with BC technology for decentralized application development. It uses the concept of artificial neural network (ANN) in a decentralized fashion over N-number of nodes with the use of BC [148]. It helps AI developers to save up to 70% of computing cost. It separates the SCs, data providers, and data training parties for privacy purposes. It involves the maximum partition of AI developers around the world by making their own currency as a universal currency. The computation cost of DBC is 30% lesser compared to the self-built ANN servers [149].

5) EFFECT.AI

It is an open-source, democratic, and decentralized network for AI. Its main objective is to develop a platform for NEO-BC to simulate AI algorithms and services. It offers the freedom and ease of use of AI services worldwide. At a high level, it includes the components like Effect M-Turk (workforce that develops AI algorithms), Effect Smart Market (buy and sell AI services), and Effect M-Power (distribute computational power) [150].

6) NUMERAI

It provides a BC-empowered AI hedge fund platform. The hedge fund is a collection of capital from various investors with risk management. It allows anonymous data analysts across the globe to submit their predictions based on ML models for managing the hedge fund [151]. Based on the accuracy achieved in the prediction model, analysts will be awarded monetary rewards or incentives. These rewards encourage more data analysts to participate in developing prediction models. Numerai raised almost $7.5 million in 2016 in two funding rounds.

7) SingularityNET

It is an open-source system and collection of SCs for the decentralized AI platform. It reveals the details of AI agents, which help in cryptocurrency exchange. It also permits anyone to write an AI algorithm and allows organizations and developers to buy, sell, and use AI at large scale [148]. This platform ensures the amount of money and incentives earned are correctly transferred to the concerned user. It has features like interoperability, data privacy, modularity, and scalability of AI models.

8) DanKu

Algorithmia has started a new project called DanKu, a public BC protocol used to evaluate and purchase AI models. It fetches the ideas from machine learning (ML) content platforms (like Kaggle) and uses SCs to eliminate trusted third-party. It is completely trustless ML SC, which allows data scientists to publish their data sets along with their evaluation functions and a kind of monetary reward for those who give the best-trained ML model [152].

VIII. CASE STUDY: RETAIL MARKETING

To demonstrate the AI-enabled SC for security as well as creating and understanding complex SCs, we present a case study on retail marketing. In this, we specifically focus on a single application area, i.e., *Icecream Retailing Store (ICRS)*.

An ICRS is a BC-based store with its own rules and agreements for selling out the various Icecream products.

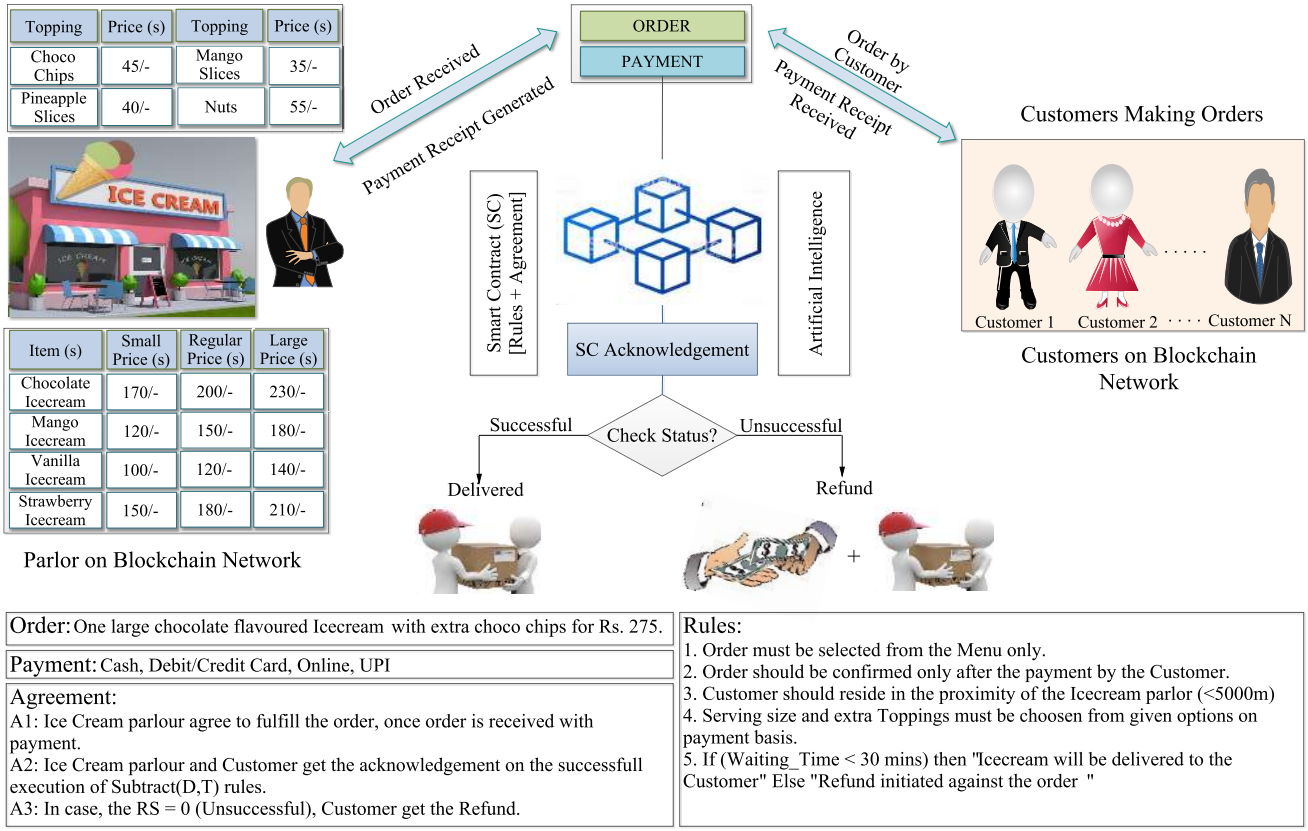

FIGURE 13. AI-enabled smart contract for retail marketing: Use case.

It has a variety of Icecreams in its menu with different serving sizes (*small, regular and large*) and additional toppings (*choco chips, mango slices, pineapple slices, and nuts*). In this scenario, we considered both customers and ICRS are on the BC network. Customers can order Icecream only if he is ordering within the range of *5000m*. Then, he can make the payment using available options (cash, UPI, credit/debit card, and Internet ng) while placing the order. A customer can also add some additional toppings to his order by paying extra for it. An ICRS is known for its service, as it claims < 30 *mins* order delivery, otherwise full amount to be refunded along with the order. The details of the workflow of SC execution between the customer and ICRS is shown in FIGURE 13 as well as in Algorithm 1.

The details of symbols used in Algorithm 1 is given in the Table 9. Customers $(C_1, C_2, C_3 \ldots C_n)$ can order their favorite icecream with different serving sizes and special toppings from distributed locations (D_L) over BC network only if they are within $5000m$ range of the ICRS. This invokes the BC SC to verify its feasibility (each request in BC network is considered as a transaction (T_x)). The order will be confirmed by the ICRS only if customer has sufficient balance, i.e., $C_{balance} > O_{amount}$. Once the order is confirmed, a payment receipt is generated and sent to the customer as an acknowledgment over the distributed network. This transaction increments and decrements the retailer ans customer balance by an order amount respectively, i.e., $(R_{balance} = R_{balance} + O_{amount})$ and $(C_{balance} = C_{balance} - O_{amount})$. Thereafter, ICRS started preparing the customers order ad set the timer as $1800sec$. If the order will not be delivered within the time bound of $1800sec$, then a full refund (O_{amount}) will be initiated by the ICRS along with the dispatched order as a penalty. The fulfillment of aforementioned rules and agreement will be taken care by the SC (self-executable, self-verifiable. and immutable).

For designing and analysis of such complex SCs, we need AI algorithms. The other use of AI techniques is to identify the user preferences (based on their order frequency) to offer them seasonal discounts. This helps to attract customers and increases the sale. The complexity of SCs can be reduced by using logic-based programming language. AI analyses the order and uses dimensionality reduction techniques (principal component analysis) to reduce the complexity and increases the security of the SC. Security in terms of hiding sensitive customer information such as D_L and $C_{balance}$. So, AI-enabled SCs are quite efficient and secure as compared to traditional SCs.

IX. OPEN ISSUES AND CHALLENGES

In this Section, we discuss research challenges and issues for the amalgamation of SC and AI. Few of the issues are as shown in FIGURE 14 and explained as follows:

Algorithm 1 Smart Contract Execution Between ICRS and Customer

Input:
$O_t, O_d, R_{menu}, R_{id}, C_w, C_d, C_{id}, P_m, BC$
$\exists\,(C_{id}, T_x) \in BC$
Output:
$R\_S = 0$ (Order Unsuccessful) or 1 (Order Successfully Delivered)

procedure Blockchain_Transaction_Data(T_x, BC)
  while $(C_{id} \in BC)$ **do**
    if $C_d < 5000\ m$ **then**
      BC permits customer to place an order at time O_t.
      if $((O_{id} \rightarrow item) \in R_{menu})$ **then**
        Customer has selected the correct items enlisted
        in ICRS menu.
      else
        Order will not proceed further for confirmation.
        if $((C_{id} \rightarrow C_{balance}) > O_{amount})$ **then**
          Customer will choose appropriate payment
          mode $(P_c, P_{debit}, P_{online}, P_{upi})$.
          $C_{balance} = C_{balance} - O_{amount}$
          $R_{balance} = R_{balance} + O_{amount}$
          Order is Confirmed by the ICRS.
        else
          Insufficient customer balance (Refil Balance
          before placing the order).
          if $(Subtract(O_t, O_d) < 1800sec)$ **then**
            Order successfully delivered.
          else
            Order successfully delivered and also full
            refund will initiated against the order as a
            penalty.
          end if
        end if
      end if
    else
      Outlet is not in the range of customer ordering location.
    end if
  end while
end procedure

A. PRIVACY

BC is a public, authentic, and secure distributed data processing system, where collected data is publicly available to all the users within the network. This raises certain privacy issues due to the increasing usage of IoT sensor-based devices.

TABLE 9. Case study abbreviations.

Acronym	Description
O_t	Customer Order Placing Time in ICRS
O_d	Order Delivery Time by ICRS
R_{menu}	Icecream Parlor Menu Card
R_{id}	Retailer ID on BC Network
C_{id}	Customer ID on BC Network
C_w	Customer Order Waiting Time
C_d	Customer distance from ICRS
P_m	Payment Mode
P_c	Cash Payment
P_{debit}	Payment through Debit Card
P_{credit}	Payment through Credit Card
P_{online}	Payment through Internet banking
P_{upi}	Payment through UPI
R_s	Return Status
T_x	BC Transaction x Data
$C_{balance}$	Customer Balance Amount
O_{id}	Order Identification Number
O_{amount}	Order Amount
$R_{balance}$	Retailer Balance Amount
D_L	Distributed Locations

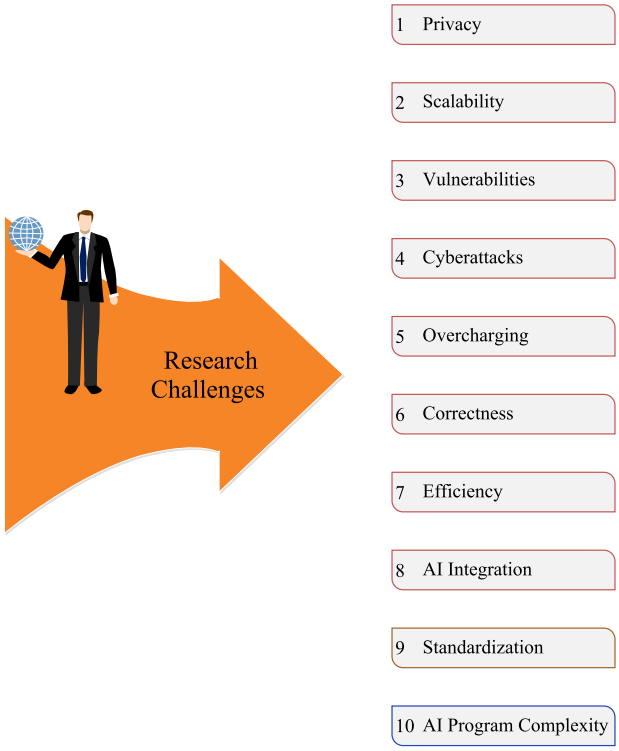

FIGURE 14. Research challenges in AI-enabled smart contract.

To handle the privacy of the user's sensitive and personal data is a challenging task. Moreover, storing the collected data over the public ledger can use the encryption and access control mechanism [37]. This will limit the access and disclosure of massive data and requires AI in performing learning and decision making tasks.

B. SCALABILITY

Presently, BC platforms have significant latencies while dealing with a large number of transactions. For example, the Bitcoin platform can perform four transactions per second, and the Ethereum platform can perform 12 transactions per second. The performance of BC transactions can be faster by

using sidechains, which can sort out the transactions among parties quickly and outside the main chain [153]. Few of the BC platforms improve the consensus algorithm of mining nodes, for instance, Algorand and IoTA, to provide higher performance than Ethereum or Hyperledger [154], [155].

C. VULNERABILITIES

Developing SC-based secure and bug-free applications is a very challenging task. As a security measure, the source code and other data over the network should be safeguarded. The specific vulnerability in SC programming can lead to major data loss. For example, SC-based DAO (Decentralized Autonomous Organization) had a programming vulnerability due to which it was hacked back in 2016 with the loss of approximately 3.6 million ether. To address the above mentioned issues, an important step is to test SCs for any bugs or vulnerabilities before deployment, and some tools are developed to evaluate the SC security states [18] [82] [156]. The execution results of SCs are in the deterministic form and not in a probabilistic form that can be an issue for decentralized AI. In decentralized AI, both machine learning and AI-based decision-making algorithms get executed as SCs through mining nodes. It usually provides non-deterministic outcomes that are random or unpredictable.

D. CYBERATTACKS

BC provided strong and secure schemes for IoT and AI analysis, but they are vulnerable to cyberattacks, such as the 51% attack [157]. The fundamental security mechanism in the BC is a consensus algorithm, and when any miner has more hashing power, he/she can compromise the consensus algorithm. Due to this reason, the decentralized system will be centralized around more farming power, where several miners are farming with higher hash powers. The security problem discussed here is more obvious in public BC, such as Bitcoin and Ethereum. Consensus algorithms are predefined among participating parties, so private BC platforms will suffer less from this problem. The private BC, like Hyperledger, the outcomes of execution can be tempered. The solution for this problem can be achieved by taking hardware support and create a trusted execution environment (TEE), for example, Intel SGX [158], ARM: TrustZone.

E. OVERCHARGING

It is the situation where SC code is not well optimized, such as the presence of dead codes, costly operations in loops, recursion, etc. Developers need to be careful in designing complex SCs by considering all those codes, which makes SC inefficient [47].

F. SMART CONTRACT CORRECTNESS

SCs are stored in the BC network, so they also possess immutability characteristics. Once the SC is designed and deployed in BC, it is impossible to update it. It is important to evaluate the correctness of the SC before deploying it into the BC network. It is challenging for a SC development team to verify the correctness of the quite long and complex SCs.

G. EFFICIENCY

The execution efficiency of a SC is of utmost importance in critical applications such as healthcare and financial systems. A SC can take data (shared data) from other SCs also. Here, the efficiency of SC matters, otherwise deadlock condition can evolve.

H. AI INTEGRATION

Integration of AI and SC is a challenging task, and not much work done on it. AI processes big data that requires powerful computing machines and is time-consuming too. SCs with AI techniques must be balanced in such a way that, the efficiency of the SC execution should not be compromised.

I. SMART CONTRACT STANDARDIZATION

The biggest challenge in creating the SCs are its standardization and social acceptability. Standardization of SCs can increase its acceptability among the BC-based systems around the world [159]. But, standardizing the SCs is quite complex and time consuming process.

J. AI PROGRAM COMPLEXITY

It is also one of the challenging task for the SC developers to develop AI-enbled SCs with low complexity. Higher the complexity of SC can lead to lower energy efficiency, which is not suitable for low powered IoT devices. This can become a barrier in its acceptability.

X. CONCLUSION

This article presents an overview of the state-of-the-art SC security vulnerabilities. In particular, we first provide a detailed review of SC, BC, and AI technologies, along with the different SC platforms. Then, the security vulnerabilities in SC code and possible solutions with traditional security schemes such as TEE, SMPC, and ZKP are explored. These schemes have high communication and computation costs. Then, we describe the integration of AI in SC that can solve the issues mentioned above. The third part of the survey discussed open issues and research challenges, which arise because of SC and AI integration issues. In future, we will implement the SC for various smart applications.

REFERENCES

[1] J. Srinivas, A. K. Das, and N. Kumar, "Government regulations in cyber security: Framework, standards and recommendations," *Future Gener. Comput. Syst.*, vol. 92, pp. 178–188, Mar. 2019.
[2] H. Tao, M. Z. A. Bhuiyan, M. A. Rahman, T. Wang, J. Wu, S. Q. Salih, Y. Li, and T. Hayajneh, "Trust data: Trustworthy and secured data collection for event detection in industrial cyber-physical system," *IEEE Trans. Ind. Informat.*, to be published.
[3] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Comput. Surv.*, vol. 52, pp. 51:1–51:34, Jul. 2019.
[4] P. Bhattacharya, S. Tanwar, U. Bodke, S. Tyagi, and N. Kumar, "BinDaaS: Blockchain-based deep-learning as-a-service in healthcare 4.0 applications," *IEEE Trans. Netw. Sci. Eng.*, to be published.

[5] R. Singh, S. Tanwar, and T. P. Sharma, "Utilization of blockchain for mitigating the distributed denial of service attacks," *Secur. Privacy*, to be published.
[6] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019.
[7] F. B. Insights. *Blockchain Technology Market to Value US$ 21,070.2 MN at CAGR of 38.4% by 2025*. Accessed: 2019. [Online]. Available: https://www.prnewswire.com/news-releases/blockchain-technology-market-to-value-us-21-070-2-mn-at-cagr-of-38-4-by-2025-exclusive-report-by-fortune-business-insights-300855866.html
[8] Z. Zheng, S. Xie, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," in *Proc. IJWGS*, vol. 14, 2018, pp. 352–375.
[9] S. Beyer. *Dealing With Privacy in Smart Contract-Based Insurances*. Accessed: 2018. [Online]. Available: https://medium.com/blackinsurance/dealing-with-privacy-in-smart-contract-based-insurances-4087319f9b48,
[10] A. Srivastava, S. K. Singh, S. Tanwar, and S. Tyagi, "Suitability of big data analytics in Indian banking sector to increase revenue and profitability," in *Proc. 3rd Int. Conf. Adv. Comput., Commun. Autom. (ICACCA)*, Sep. 2017, pp. 1–6.
[11] NASSCOM and A. Comprehensive Analysis. *How Indian States are Driving Public Sector Blockchain Adoption in India*. Accessed: 2019. [Online]. Available: https://avasant.com/report/blockchain-adoption-indian-states/
[12] A. Nejjari. *Blockchain Demystified: Smart Contracts*. Accessed: 2018. [Online]. Available: https://medium.com/the-archer/blockchain-demystified-smart-contracts-c23ab844ef4a
[13] I. Mistry, S. Tanwar, S. Tyagi, and N. Kumar, "Blockchain for 5G-enabled IoT for industrial automation: A systematic review, solutions, and challenges," *Mech. Syst. Signal Process.*, vol. 135, Jan. 2020, Art. no. 106382.
[14] China Academy of Information and Communication Technology. *Blockchain White Paper*. Accessed: 2018. [Online]. Available: http://www.caict.ac.cn/english/yjcg/bps/201901/P020190131402018699770.pdf
[15] S. K. Singh, M. M. Salim, M. Cho, J. Cha, Y. Pan, and J. H. Park, "Smart contract-based pool hopping attack prevention for blockchain networks," *Symmetry*, vol. 11, no. 7, p. 941, Jul. 2019.
[16] D. B. D. and F. Al-Turjman, "A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks," *Ad Hoc Netw.*, vol. 97, Feb. 2020, Art. no. 102022.
[17] N. Kabra, P. Bhattacharya, S. Tanwar, and S. Tyagi, "MudraChain: Blockchain-based framework for automated cheque clearance in financial institutions," *Future Gener. Comput. Syst.*, vol. 102, pp. 574–587, Jan. 2020.
[18] P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Bünzli, and M. Vechev, "Securify: Practical security analysis of smart contracts," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Jan. 2018, pp. 67–82.
[19] D. Perez and B. Livshits, "Smart contract vulnerabilities: Does anyone care?" 2019, *arXiv:1902.06710*. [Online]. Available: https://arxiv.org/abs/1902.06710
[20] Cointelegraph. *Parity Multisig Wallet Hacked, or How Come?* Accessed: 2017. [Online]. Available: https://cointelegraph.com/news/parity-multisig-wallet-hacked-or-how-come
[21] L. Kiffer, D. Levin, and A. Mislove, "Analyzing Ethereum's contract topology," in *Proc. Internet Meas. Conf. (IMC)*, New York, NY, USA, 2018, pp. 494–499.
[22] S. K. Singh, S. Rathore, and J. H. Park, "Blockiotintelligence: A blockchain-enabled intelligent IoT architecture with artificial intelligence," *Future Gener. Comput. Syst.*, to be published.
[23] H. Nguyen. *How ai Will Make Smart Contract Smart*. Accessed: 2018. [Online]. Available: https://www.techuk.org/insights/opinions/item/12960-how-ai-will-make-smart-contract-smart
[24] J. Vora, P. Italiya, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and K.-F. Hsiao, "Ensuring privacy and security in E-health records," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Jul. 2018, pp. 1–5.
[25] L. Guan, F. Hu, F. Al-Turjman, M. B. Khan, and X. Yang, "A non-contact paraparesis detection technique based on 1D-CNN," *IEEE Access*, vol. 7, pp. 182280–182288, 2019.
[26] L. Blockchain. *Combining Blockchain and ai to Make Smart Contracts Smarter*. Accessed: 2018. [Online]. Available: https://medium.com/blockchain-for-law/combining-blockchain-and-ai-to-make-smart-contracts-smarter-88081712b192
[27] W. Oscar. *Ai Smart Contracts-The Past, Present and Future*. Accessed: 2018. [Online]. Available: https://hackernoon.com/ai-smart-contracts-the-past-present-and-future-625d3416807b
[28] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "Tactile Internet and its applications in 5G era: A comprehensive review," *Int. J. Commun. Syst.*, vol. 32, no. 14, p. e3981, Sep. 2019.
[29] I. Budhiraja, S. Tyagi, S. Tanwar, N. Kumar, and J. J. P. C. Rodrigues, "Tactile Internet for smart communities in 5G: An insight for NOMA-based solutions," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 3104–3112, May 2019.
[30] P. Tasatanattakool and C. Techapanupreeda, "Blockchain: Challenges and applications," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 473–475.
[31] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, "An overview of smart contract: Architecture, applications, and future trends," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 108–113.
[32] A. B. Kurtulmus and K. Daniel, "Trustless machine learning contracts; Evaluating and exchanging machine learning models on the ethereum blockchain," *CoRR*, vol. abs/1802.10185, pp. 1–11, Feb. 2018.
[33] P. Mamoshina, L. Ojomoko, Y. Yanovich, A. Ostrovski, A. Botezatu, P. Prikhodko, E. Izumchenko, A. Aliper, K. Romantsov, A. Zhebrak, I. O. Ogu, and A. Zhavoronkov, "Converging blockchain and next-generation artificial intelligence technologies to decentralize and accelerate biomedical research and healthcare," *Oncotarget*, vol. 9, pp. 5665–5690, Jan. 2018.
[34] Z. Geng, Y. He, T. Niu, H. Li, L. Sun, W. Cheng, and X. Li, "Poster: Smart-contract based incentive mechanism for K-anonymity privacy protection in LBSs," in *Proc. IEEE Symp. Privacy-Aware Comput. (PAC)*, Aug. 2017, pp. 200–201.
[35] T. N. Dinh and M. T. Thai, "AI and blockchain: A disruptive integration," *Computer*, vol. 51, no. 9, pp. 48–53, Sep. 2018.
[36] T. Marwala and B. Xing, "Blockchain and artificial intelligence," Feb. 2018, *arXiv:1802.04451*. [Online]. Available: https://arxiv.org/abs/1802.04451
[37] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for AI: Review and open research challenges," *IEEE Access*, vol. 7, pp. 10127–10149, 2019.
[38] A. Mense and M. Flatscher, "Security vulnerabilities in Ethereum smart contracts," in *Proc. 20th Int. Conf. Inf. Integr. Web-Based Appl. Services (iiWAS)*, Jul. 2018, pp. 955–962.
[39] R. Cheng, F. Zhang, J. Kos, W. He, N. Hynes, N. Johnson, A. Juels, A. Miller, and D. Song, "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Jun. 2019, pp. 185–200.
[40] C. Li, B. Palanisamy, and R. Xu, "Scalable and privacy-preserving design of ON/OFF-chain smart contracts," in *Proc. IEEE 35th Int. Conf. Data Eng. Workshops (ICDEW)*, Apr. 2019, pp. 7–12.
[41] S. Tanwar, K. Parekh, and R. Evans, "Blockchain-based electronic healthcare record system for healthcare 4.0 applications," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102407.
[42] B. K. Mohanta, S. S. Panda, and D. Jena, "An overview of smart contract and use cases in blockchain technology," in *Proc. 9th Int. Conf. Comput., Commun. Netw. technol. (ICCCNT)*, Jul. 2018, pp. 1–4.
[43] U. Bodkhe, P. Bhattacharya, S. Tanwar, S. Tyagi, N. Kumar, and M. S. Obaidat, "BloHosT: Blockchain enabled smart tourism and hospitality management," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Aug. 2019, pp. 1–5.
[44] M. Pratap. *Everything you Need to Know About Smart Contracts: A Beginner—Guide*. Accessed:2018. [Online]. Available: https://hackernoon.com/everything-you-needto-know-about-smart-contracts-a-beginners-guide-c13cc138378a
[45] R. Gupta, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and B. Sadoun, "HaBiTs: Blockchain-based telesurgery framework for healthcare 4.0," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Aug. 2019, pp. 1–5.
[46] J. Vora, A. Nayyar, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and J. J. P. C. Rodrigues, "BHEEM: A blockchain-based framework for securing electronic health records," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–6.
[47] Z. Zheng, S. Xie, H.-N. Dai, W. Chen, X. Chen, J. Weng, and M. Imran, "An overview on smart contracts: Challenges, advances and platforms," *Future Gener. Comput. Syst.*, vol. 105, pp. 475–491, Apr. 2020.
[48] T. Sato and Y. Himura, "Smart-contract based system operations for permissioned blockchain," in *Proc. 9th IFIP Int. Conf. New technol., Mobility Secur. (NTMS)*, Feb. 2018, pp. 1–6, Feb. 2018.

[49] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. Kishigami, "Blockchain contract: Securing a blockchain applied to smart contracts," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2016, pp. 467–468.

[50] J. Ellul and G. J. Pace, "AlkylVM: A virtual machine for smart contract blockchain connected Internet of Things," in *Proc. 9th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Feb. 2018, pp. 1–4.

[51] S. Tanwar, P. Patel, K. Patel, S. Tyagi, N. Kumar, and M. S. Obaidat, "An advanced Internet of Thing based Security Alert System for Smart Home," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Jul. 2017, pp. 25–29.

[52] J. Vora, S. Tanwar, S. Tyagi, N. Kumar, and J. J. P. C. Rodrigues, "Home-based exercise system for patients using IoT enabled smart speaker," in *Proc. IEEE 19th Int. Conf. E-Health Netw., Appl. Services*, Oct. 2017, pp. 1–6.

[53] T. Dickerson, P. Gazzillo, M. Herlihy, and E. Koskinen, "How to add concurrency to smart contracts," *Bull. Eur. Assoc. Theor. Comput. Sci.*, no. 124, pp. 22–33, Feb. 2018.

[54] J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC: Role-based access control using smart contract," *IEEE Access*, vol. 6, pp. 12240–12251, 2018.

[55] A. Rosic. *Smart Contracts: The Blockchain Technology That will Replace Lawyers*. Accessed: 2016. [Online]. Available: https://blockgeeks.com/guides/smart-contracts/

[56] E. Zhou, S. Hua, B. Pi, J. Sun, Y. Nomura, K. Yamashita, and H. Kurihara, "Security assurance for smart contract," in *Proc. 9th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Feb. 2018, pp. 1–5.

[57] C. Wright and A. Serguieva, "Sustainable blockchain-enabled services: Smart contracts," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 4255–4264.

[58] A. Ramachandran and M. Kantarcioglu, "Using blockchain and smart contracts for secure data provenance management," *CoRR*, vol. abs/1709.10000, pp. 1–11, Sep. 2017.

[59] S. R. Niya, F. Shupfer, T. Bocek, and B. Stiller, "Setting up flexible and light weight trading with enhanced user privacy using smart contracts," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2018, pp. 1–2.

[60] G. Destefanis, M. Marchesi, M. Ortu, R. Tonelli, A. Bracciali, and R. Hierons, "Smart contracts vulnerabilities: A call for blockchain software engineering?" in *Proc. Int. Workshop Blockchain Oriented Softw. Eng. (IWBOSE)*, Mar. 2018, pp. 19–25.

[61] A. Ashish. *Introduction to Zero Knowledge Proof: The Protocol of Next Generation Blockchain*. Accessed: 2018. [Online]. Available: https://medium.com/@kotsbtechcdac/introduction-to-zero-knowledge-proof-the-protocol-of-next-generation-blockchain-305b2fc7f8e5

[62] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 839–858.

[63] *Artificial Intelligence Overview*. Accessed: Mar. 2018. [Online]. Available: https://www.tutorialspoint.com/index.htm

[64] R. Chu. *What is AI? A Brief Explanation for Layman*. Accessed: 2018. [Online]. Available: https://medium.com/datadriveninvestor/what-is-ai-a-brief-explanation-for-layman-f79f368702ea

[65] D. Alayon. *Understanding Artificial Intelligence*. Accessed: 2018. [Online]. Available: https://medium.com/future-today/understanding-artificial-intelligence-f800b51c767f

[66] B. Garware. *Privacy-Preserving ai (Private ai)—The Rise of Federated Learning*. Accessed: 2019. [Online]. Available: https://www.persistent.com/blogs/privacy-preserving-aiprivate-ai-the-rise-of-federated-learning/

[67] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat.*, Fort Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.

[68] J. Soria-Comas, J. Domingo-Ferrer, D. Sanchez, and D. Megias, "Individual differential privacy: A utility-preserving formulation of differential privacy guarantees," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 6, pp. 1418–1429, Jun. 2017.

[69] T. O. Team. *Towards an Open-Source Secure Enclave*. Accessed: 2018. [Online]. Available: https://medium.com/oasislabs/towards-an-open-source-secure-enclave-659ac27b871a

[70] A. S. Almasoud, M. M. Eljazzar, and F. K. Hussain, "Toward a self-learned smart contracts," *CoRR*, vol. abs/1812.10485, pp. 269–273, Feb. 2018.

[71] Z. Chen, W. Wang, X. Yan, and J. Tian, "Cortex—AI on blockchain the decentralized ai autonomous system," Cortexlabs, Beijing, China.

[72] *Common Smart Contract Vulnerabilities and how to Mitigate Them*. Accessed: 2018. [Online]. Available: https://yos.io/2018/10/20/smart-contract-vulnerabilities-and-how-to-mitigate-them/

[73] S. K. Singh and S. Tanwar, "Analysis of software testing techniques: Theory to practical approach," *Indian J. Sci. Technol.*, vol. 9, no. 32, 2016.

[74] A. Mense and M. Flatscher, "Security vulnerabilities in Ethereum smart contracts," in *Proc. 20th Int. Conf. Inf. Integr. Web Appl. Services*, New York, NY, USA, 2018, pp. 375–380.

[75] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on Ethereum smart contracts (SOK)," in *Principles Security Trust*, M. Maffei and M. Ryan, eds. Berlin, Germany: Springer, 2017, pp. 164–186.

[76] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on Ethereum systems security: Vulnerabilities, attacks and defenses," *CoRR*, vol. abs/1908.04507, pp. 1–29, Aug. 2019.

[77] W. Chan and B. Jiang, "Fuse: An architecture for smart contract fuzz testing service," in *Proc. 25th Asia–Pacific Softw. Eng. Conf. (APSEC)*, Dec. 2018, pp. 707–708.

[78] W. Dingman, A. Cohen, N. Ferrara, A. Lynch, P. Jasinski, P. E. Black, and L. Deng, "Defects and vulnerabilities in smart contracts, a classification using the Nist bugs framework," *Int. J. Netw. Distrib. Comput.*, vol. 7, pp. 121–132, Jun. 2019.

[79] N. Group. *Discovering Smart Contract Vulnerabilities With Goatcasino*. Accessed: 2018. [Online]. Available: https://www.nccgroup.trust/uk/our-research/discovering-smart-contract-vulnerabilities-with-goatcasino//

[80] *How to Test Ethereum Smart Contracts: Audit Best Practices*. Accessed: 2018. [Online]. Available: https://dev.to/smartym/how-to-test-ethereum-smart-contracts-audit-best-practices-53kg

[81] E. Zhou, S. Hua, B. Pi, J. Sun, Y. Nomura, K. Yamashita, and H. Kurihara, "Security assurance for smart contract," in *Proc. 9th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Feb. 2018, pp. 1–5.

[82] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2016, pp. 254–269.

[83] D. Siegel. *Understanding the Dao Attack*. Accessed: 2016. [Online]. Available: https://www.coindesk.com/understanding-dao-hack-journalists

[84] S. Falkon. *The Story of the DAO—Its History and Consequences*. Accessed: 2017. [Online]. Available: https://medium.com/swlh/the-story-of-the-dao-its-history-and-consequences-71e6a8a551ee

[85] D. Hays. *Smart Contracts*. Accessed: 2018. [Online]. Available: https://cryptoresearch.report/crypto-research/smart-contracts/

[86] T. Bahrynovska. *Smart Contract as a Safe and Secure Method of Conducting Transactions*. Accessed: 2017. [Online]. Available: https://applicature.com/blog/blockchain-technology/history-of-ethereum-security-vulnerabilities-hacks-and-their-fixes

[87] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2015, pp. 57–64.

[88] R. Cheng, F. Zhang, J. Kos, W. He, N. Hynes, N. Johnson, A. Juels, A. Miller, and D. Song, "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contract execution," Apr. 2018, *arXiv:1804.05141*. [Online]. Available: https://arxiv.org/abs/1804.05141

[89] B. McGillion, T. Dettenborn, T. Nyman, and N. Asokan, "Open-TEE—An open virtual trusted execution environment," Jun. 2015, *arXiv:1506.07367*. [Online]. Available: https://arxiv.org/abs/1506.07367

[90] Y. Fan, S. Liu, G. Tan, X. Lin, G. Zhao, and J. Bai, "One secure access scheme based on trusted execution environment," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Aug. 2018, pp. 16–21.

[91] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, N. Kumar, Y. Park, and S. Tanwar, "Design of an anonymity-preserving group formation based authentication protocol in global mobility networks," *IEEE Access*, vol. 6, pp. 20673–20693, 2018.

[92] A. M. Azab, P. Ning, J. Shah, Q. Chen, R. Bhutkar, G. Ganesh, J. Ma, and W. Shen, "Hypervision across worlds: Real-time kernel protection from the ARM trustzone secure world," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2014, pp. 90–102.

[93] J. Jang, S. Kong, M. Kim, D. Kim, and B. B. Kang, "SeCReT: Secure channel between rich execution environment and trusted execution environment," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, Feb. 2015, pp. 1–5.

[94] P. Jain, S. Desai, S. Kim, M.-W. Shih, J. Lee, C. Choi, Y. Shin, T. Kim, B. B. Kang, and D. Han, "OpenSGX: An open platform for SGX research," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, Feb. 2016, pp. 1–16.

[95] B. Ngabonziza, D. Martin, A. Bailey, H. Cho, and S. Martin, "Trustzone explained: Architectural features and use cases," in *Proc. IEEE 2nd Int. Conf. Collaboration Internet Comput. (CIC)*, Nov. 2016, pp. 445–451.
[96] K. Rubinov, L. Rosculete, T. Mitra, and A. Roychoudhury, "Automated partitioning of android applications for trusted execution environments," in *Proc. 38th Int. Conf. Softw. Eng. (ICSE)*, May 2016, pp. 923–934.
[97] M. A. Rahman, A. T. Asyhari, S. Azad, M. M. Hasan, C. P. C. Munaiseche, and M. Krisnanda, "A cyber-enabled mission-critical system for post-flood response: Exploiting TV white space as network backhaul links," *IEEE Access*, vol. 7, pp. 100318–100331, 2019.
[98] L. Zhang. *Intel SGX and Blockchain: The IEXEC End-To-End Trusted Execution Solution*. Accessed: 2018. [Online]. Available: https://medium.com/iex-ec/iexec-end-to-end-sgx-solution-fee1e63297b2
[99] Z. Hua, J. Gu, Y. Xia, H. Chen, B. Zang, and H. Guan, "VTZ: Virtualizing ARM trustzone," *Proc. 26th Secur. Symp.*, Vancouver, BC, Canada, 2017, pp. 541–556.
[100] X. Ge, H. Vijayakumar, and T. Jaeger, "Sprobes: Enforcing kernel code integrity on the trustzone architecture," *CoRR*, vol. abs/1410.7747, pp. 1–10, Oct. 2014.
[101] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *J. Netw. Comput. Appl.*, vol. 126, pp. 45–58, Jan. 2019.
[102] T. Mizrahi. *Securing the Blockchain With Multi-Party Computation*. Accessed: 2019. [Online]. Available: https://www.unboundtech.com/securing-blockchain-multi-party-computation/
[103] U. Maurer, "Secure multi-party computation made simple," *Discrete Appl. Math.*, vol. 154, no. 2, pp. 370–381, 2006.
[104] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C.-Z. Gao, H. Li, and Y.-A. Tan, "Secure multi-party computation: Theory, practice and applications," *Inf. Sci.*, vol. 476, pp. 357–372, Feb. 2019.
[105] K. Patel, "Secure multiparty computation using secret sharing," in *Proc. Int. Conf. Signal Process., Commun., Power Embedded Syst. (SCOPES)*, Oct. 2016, pp. 863–866.
[106] A. Lapets, N. Volgushev, A. Bestavros, F. Jansen, and M. Varia, "Secure MPC for analytics as a Web application," in *Proc. IEEE Cybersecur. Develop.(SecDev)*, Nov. 2016, pp. 73–74.
[107] M. Marwan, A. Kartit, and H. Ouahmane, "Applying secure multi-party computation to improve collaboration in healthcare cloud," in *Proc. 3rd Int. Conf. Syst. Collaboration (SysCo)*, Nov. 2016, pp. 1–6.
[108] H. Kaur, N. Kumar, and S. Batra, "An efficient multi-party scheme for privacy preserving collaborative filtering for healthcare recommender system," *Future Gener. Comput. Syst.*, vol. 86, pp. 297–307, Sep. 2018.
[109] X. Ma, F. Zhang, X. Chen, and J. Shen, "Privacy preserving multi-party computation delegation for deep learning in cloud computing," *Inf. Sci.*, vol. 459, pp. 103–116, Aug. 2018.
[110] A. Lapets, N. Volgushev, A. Bestavros, F. Jansen, and M. Varia, "Secure multi-party computation for analytics deployed as a lightweight web application," OpenBU, Teh. Rep., 2016.
[111] A. Patra and D. Ravi, "On the power of hybrid networks in multi-party computation," *IEEE Trans. Inf. Theory*, vol. 64, no. 6, pp. 4207–4227, Jun. 2018.
[112] K. D. Albab, R. Issa, A. Lapets, A. Bestavros, and N. Volgushev, "Scalable secure multi-party network vulnerability analysis via symbolic optimization," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2017, pp. 211–216.
[113] S. Almuhammadi and C. Neuman, "Security and privacy using one-round zero-knowledge proofs," in *Proc. 7th IEEE Int. Conf. E-Commerce Technol. (CEC)*, Oct. 2006, pp. 435–438.
[114] J. J. Hathaliya, S. Tanwar, S. Tyagi, and N. Kumar, "Securing electronics healthcare records in Healthcare 4.0 : A biometric-based approach," *Comput. Elect. Eng.*, vol. 76, pp. 398–410, Jun. 2019.
[115] T. M. Zaw, M. Thant, and S. Bezzateev, "User authentication in SSL handshake protocol with zero-knowledge proof," in *Proc. Wave Electron. Appl. Inf. Telecommun. Syst. (WECONF)*, Nov. 2018, pp. 1–8.
[116] A. Beydemir and I. Sogukpinar, "Lightweight zero knowledge authentication for Internet of Things," in *Proc. Int. Conf. Comput. Sci. Eng. (UBMK)*, Oct. 2017, pp. 360–365.
[117] I.-H. Chuang, B.-J. Guo, J.-S. Tsai, and Y.-H. Kuo, "Multi-graph Zero-knowledge-based authentication system in Internet of Things," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–60
[118] A. Chiesa, M. Forbes, T. Gur, and N. Spooner, "Spatial isolation implies zero knowledge even in a quantum world," in *Proc. IEEE 59th Annu. Symp. Found. Comput. Sci. (FOCS)*, Oct. 2018, pp. 755–765.
[119] A. Broadbent, Z. Ji, F. Song, and J. Watrous, "Zero-Knowledge Proof Systems for QMA," in *Proc. IEEE 57th Annu. Symp. Found. Comput. Sci. (FOCS)*, Oct. 2016, pp. 31–40.
[120] U. Ruhrmair, J. Martinez-Hurtado, X. Xu, C. Kraeh, C. Hilgers, D. Kononchuk, J. J. Finley, and W. P. Burleson, "Virtual proofs of reality and their physical implementation," in *Proc. IEEE Symp. Secur. Privacy*, May 2015, pp. 70–85.
[121] B. Li, Y. Wang, P. Shi, H. Chen, and L. Cheng, "FPPB: A fast and privacy-preserving method based on the permissioned blockchain for fair transactions in sharing economy," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Aug. 2018, pp. 1368–1373.
[122] A. Pannu, "Artificial intelligence and its application in different areas," *Artif. Intell.*, vol. 4, no. 10, pp. 79–84, 2015.
[123] F. Al-Turjman, "Intelligence and security in big 5G-oriented IoNT: An overview," *Future Gener. Comput. Syst.*, vol. 102, pp. 357–368, Jan. 2020.
[124] A. Kumari, S. Tanwar, S. Tyagi, and N. Kumar, "Verification and validation techniques for streaming big data analytics in Internet of Things environment," *IET Netw.*, vol. 8, no. 3, pp. 155–163, May 2019.
[125] A. Kumari, S. Tanwar, S. Tyagi, N. Kumar, M. Maasberg, and K.-K.-R. Choo, "Multimedia big data computing and Internet of Things applications: A taxonomy and process model," *J. Netw. Comput. Appl.*, vol. 124, pp. 169–195, Dec. 2018.
[126] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "Tactile-Internet-based telesurgery system for healthcare 4.0: An architecture, research challenges, and future directions," *IEEE Netw.*, vol. 33, no. 6, pp. 22–29, Nov. 2019.
[127] W. Oscar. *AI on Blockchain—What's the catch?* Accessed: 2018. [Online]. Available: https://hackernoon.com/how-cortex-brings-ai-on-the-blockchain-86d08922bb2a
[128] C. Blum and R. Groß, *Swarm Intelligence in Optimization and Robotics*. Berlin, Germany: Springer, 2015., pp. 1291–1309.
[129] D. Kim, D. Shin, and D. Shin, "Unauthorized access point detection using machine learning algorithms for information protection," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Aug. 2018, pp. 1876–1878.
[130] N. Senavirathne and V. Torra, "Approximating robust linear regression with an integral privacy guarantee," in *Proc. 16th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2018, pp. 1–10,
[131] X. Li and T. Zhang, "An exploration on artificial intelligence application: From security, privacy and ethic perspective," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Apr. 2017, pp. 416–420.
[132] G. Natesan and J. Liu, "An adaptive learning model for k-anonymity location privacy protection," in *Proc. IEEE 39th Annu. Comput. Softw. Appl. Conf.*, vol. 3, Jul. 2015, pp. 10–16.
[133] Y.-M. Ye, C.-C. Pan, and G.-K. Yang, "An improved location-based service authentication algorithm with personalized k-anonymity," in *Proc. China Satell. Navigat. Conf. (CSNC)*, vol. 1, J. Sun, J. Liu, S. Fan, and F. Wang, Eds. Singapore: Springer, 2016, pp. 257–266.
[134] S. Tyagi, S. Tanwar, and N. Kumar, "Learning automata-based coverage oriented clustering in HWSNs," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Eng.*, May 2015, pp. 78–83,.
[135] S. Tyagi, S. Tanwar, S. Gupta, N. Kumar, S. Misra, J. Rodrigues, and S. Ullah, "Bayesian coalition game-based optimized clustering in wireless sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 3540–3545.
[136] B. Feizizadeh, M. S. Roodposhti, T. Blaschke, and J. Aryal, "Comparing GIS-based support vector machine kernel functions for landslide susceptibility mapping," *Arabian J. Geosci.*, vol. 10, p. 122, Mar. 2017.
[137] A. K. Patel, S. Chatterjee, and A. K. Gorai, "Development of machine vision-based ore classification model using support vector machine (SVM) algorithm," *Arabian J. Geosci.*, vol. 10, p. 107, Mar. 2017.
[138] S. Sathyadevan and R. R. Nair, "Comparative analysis of decision tree algorithms: ID3, C4.5 and random forest," in *Computational Intelligence in Data Mining*, vol. 1, L. C. Jain, H. S. Behera, J. K. Mandal, and D. P. Mohapatra, Eds. New Delhi, India: Springer, 2015, pp. 549–562.

[139] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient kNN classification algorithm for big data," *Neurocomputing*, vol. 195, pp. 143–148, Jun. 2016.
[140] M. Kuhkan, "A method to improve the accuracy of k-nearest neighbor algorithm," *Int. J. Comput. Eng. Inf. Technol.*, vol. 8, no. 6, pp. 90–95, 2016.
[141] E. Hodo, X. Bellekens, A. Hamilton, P.-L. Dubouilh, E. Iorkyase, C. Tachtatzis, and R. Atkinson, "Threat analysis of IoT networks using artificial neural network intrusion detection system," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, May 2016, pp. 1–6.
[142] H. Mekki, A. Mellit, and H. Salhi, "Artificial neural network-based modelling and fault detection of partial shaded photovoltaic modules," *Simul. Model. Pract. Theory*, vol. 67, pp. 1–13, Sep. 2016.
[143] R. S. J. Charlier, *Neural Networks and Tensors for Smart Contracts and Cryptocurrencies Monitoring*. Accessed: 2017. [Online]. Available: http://2018.ds3-datascience-polytechnique.fr/wp-content/uploads/2018/06/DS3-066
[144] W. Oscar. *AI Smart Contracts—The Past, Present, and Future*. Accessed: 2018.[Online]. Available: https://hackernoon.com/ai-smart-contracts-the-past-present-and-future-625d3416807b
[145] *ACCESSWIRE*. Accessed: 2019. [Online]. Available: https://finance.yahoo.com/news/blockchain-startup-fetch-ai-releases-122500373.html
[146] S. Kaneriya, M. Chudasama, S. Tanwar, S. Tyagi, N. Kumar, and J. J. P. C. Rodrigues, "Markov decision-based recommender system for sleep apnea patients," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
[147] *CoinMarketCap*. Accessed: 2019. [Online]. Available: https://coinmarketcap.com/currencies/matrix-ai-network/
[148] *NPTEL*. Accessed: 2018. [Online]. Available: https://nptel.ac.in/content/storage2/nptel_data3/html/mhrd/ict/text/106105184/lec55.pdf
[149] *Linkedin*. Accessed: 2017. [Online]. Available: https://in.linkedin.com/company/deepbrain-chain
[150] J. Rodriguez. *Why Decentralized AI Matters Part III: Technologies*. Accessed: 2018. [Online]. Available: https://medium.com/datadriveninvestor/why-decentralized-ai-matters-part-iii-technologies-930c3c9d10d
[151] J. Vora, D. Vekaria, S. Tanwar, and S. Tyagi, "Machine learning-based voltage dip measurement of smart energy meter," in *Proc. 5th Int. Conf. Parallel, Distrib. Grid Comput. (PDGC)*, Dec. 2018, pp. 828–832.
[152] J. Rodriguez. *A Decentralized Kaggle: Inside Algorithmia's Approach to Blockchain-Based AI Competitions*. Accessed: 2018. [Online]. Available: https://towardsdatascience.com/a-decentralized-kaggle-inside-algorithmias-approach-to-blockchain-based-ai-competitions-8c6aec99e89b
[153] G.-H. Hwang, P.-H. Chen, C.-H. Lu, C. Chiu, H.-C. Lin, and A.-J. Jheng, "InfiniteChain: A multi-chain architecture with distributed auditing of sidechains for public blockchains," in *Blockchain—ICBC*, S. Chen, H. Wang, and L.-J. Zhang, Eds. Cham, Switzerland: Springer, 2018, pp. 47–60.
[154] X. Boyen, C. Carr, and T. Haines, "Graphchain: A blockchain-free scalable decentralised ledger," in *Proc. 2nd ACM Workshop Blockchains, Cryptocurrencies, Contracts (BCC)*, New York, NY, USA: ACM, 2018, pp. 21–33.
[155] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling Byzantine agreements for cryptocurrencies," in *Proc. 26th Symp. Operating Syst. Princ. (SOSP)*. New York, NY, USA: ACM, 2017, pp. 51–68.
[156] S. Tikhomirov, E. Voskresenskaya, I. Ivanitskiy, R. Takhaviev, E. Marchenko, and Y. Alexandrov, "SmartCheck: Static analysis of Ethereum smart contracts," in *Proc. 1st Int. Workshop Emerg. Trends Softw. Eng. Blockchain (WETSEB)*, May 2018, pp. 9–16.
[157] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Gener. Comput. Syst.*, to be published.
[158] M. Brandenburger, C. Cachin, R. Kapitza, and A. Sorniotti, "Blockchain and trusted computing: Problems, pitfalls, and a solution for hyperledger fabric," *CoRR*, vol. abs/1805.08541, pp. 1–13, May 2018.
[159] R. Norvill, B. Fiz, R. State, and A. Cullen, "Standardising smart contracts: Automatically inferring ERC standards," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, Seoul, South Korea, May 2019, pp. 192–195.

RAJESH GUPTA is currently a full-time Research Scholar with the Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, India, under the supervision of S. Tanwar. His research interests include fog computing in healthcare, blockchain, machine larning, and device-to-device communication for 5G.

SUDEEP TANWAR received the B.Tech. degree from Kurukshetra University, India, in 2002, the M.Tech degree (Hons.) from Guru Gobind Singh Indraprastha University, Delhi, India, in 2009, and the Ph.D. degree with specialization in wireless sensor network, in 2016. He is currently an Associate Professor with the Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad, India. He is also a Visiting Professor with Jan Wyzykowski University, Polkowice, Poland, and the University of Pitesti, Pitesti, Romania. He has authored or coauthored more than 100 technical research articles published in leading journals and conferences from the IEEE, Elsevier, Springer, Wiley, and so on. Some of his research findings are published in top cited journals, such as the IEEE TRANSACTIONS ON TVT, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, *Applied Soft Computing*, the *Journal of Network and Computer Application*, *Pervasive and Mobile Computing*, the *International Journal of Communication System*, *Telecommunication System*, *Computer and Electrical Engineering*, and the IEEE SYSTEMS JOURNAL. He has also published three edited/authored books with International/National Publishers. He has guided many students leading to M.E./M.Tech. and guiding students leading to Ph.D. His current interests include wireless sensor networks, fog computing, smart grid, the IoT, and blockchain technology. He has been awarded best research paper awards from IEEE GLOBECOM 2018, IEEE ICC 2019, and Springer ICRIC-2019. He was invited as the Guest Editors/Editorial Board Members of many International Journals, invited for keynote Speaker in many International Conferences held in Asia and invited as a Program Chair, a Publications Chair, a Publicity Chair, and a Session Chair in many International Conferences held in North America, Europe, Asia, and Africa. He is also an Associate Editor of IJCS, Wiley and Security and Privacy Journal, and Wiley.

FADI AL-TURJMAN received the Ph.D. degree in computer science from Queen's University, Kingston, Ontario, Canada, in 2011. He is currently a Professor with Near East University, Nicosia, Cyprus. He is a leading authority in the areas of smart/cognitive, wireless, and mobile networks architectures, protocols, deployments, and performance evaluation. His publication history spans over 250 publications in journals, conferences, patents, books, and book chapters, in addition to numerous keynotes and plenary talks at flagship venues. He has written and edited more than 25 books about cognition, security, and wireless sensor networks deployments in smart environments, published by Taylor & Francis, Elsevier, and Springer. He has received several recognitions and best papers awards at top international conferences. He also received the prestigious Best Research Paper Award from Elsevier *Computer Communications* Journal, from 2015 to 2018, in addition to the Top Researcher Award for 2018 at Antalya Bilim University, Turkey. Prof. Al-Turjman has led a number of international symposia and workshops in flagship communication society conferences. He also serves as the Lead Guest Editor for several well reputed journals, including *Computer Communications* (Elsevier), MONET (Springer), and *IET Wireless Sensor Systems* journals.

PRIT ITALIYA is currently pursuing the bachelor's degree from Nirma University, Ahmedabad, India. His research interests include blockchain, big data analytics, fog computing, and cloud computing.

ALI NAUMAN received the B.E. degree in electrical (telecommunication) engineering from COMSATS University Islamabad, Pakistan, in 2013, and the M.S. degree from the Institute of Space Technology Islamabad, Pakistan, in 2016. He is currently pursuing the Ph.D. degree with the Wireless Information Networking Laboratory (WINLab), Department of Information and Communication Engineering, Yeungnam University, Gyeongsang, South Korea. From August 2013 to July 2014, he was an Engineer with Inbox Business Technologies, Pakistan. From August 2014 to March 2017, he was an Operations and Maintenance Manager at Mobiserve, Pakistan. From April 2017 to August 2018, he was a Researcher with the Institute of Space Technology, Islamabad, Pakistan. His research interests are focused in the area of wireless multimedia communication in the Internet of Things, wireless multimedia sensor networks and wireless communication via high altitude platforms, resource management, software defined networks, and machine learning.

SUNG WON KIM received the B.S. and M.S. degrees from the Department of Control and Instrumentation Engineering, Seoul National University, South Korea, in 1990 and 1992, respectively, and the Ph.D. degree from the School of Electrical Engineering and Computer Sciences, Seoul National University, South Korea, in August 2002. From January 1992 to August 2001, he was a Researcher with the Research and Development Center of LG Electronics, South Korea. From August 2001 to August 2003, he was a Researcher with the Research and Development Center of AL Tech, South Korea. From August 2003 to February 2005, he was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, USA. In March 2005, he joined the Department of Information and Communication Engineering, Yeungnam University, Gyeongsang, South Korea, where he is currently a Professor. His research interests include resource management, wireless networks, mobile computing, performance evaluation, and machine learning.