Bima Journal of Science and Technology, Vol. 8(2B) July, 2024 ISSN: 2536-6041

DOI: 10.56892/bima.v8i2B.717

Enhancing Byzantine-Fault-Tolerant Consensus in Permissioned Blockchain-Empowered Vehicle-to-Everything (V2X) Network Via Multitask Learning: A Review

Dan'azumi Hussaini[1]*, Fatima Umar Zambuk[2], Badamasi Iman Ya'u and Ajiya Auwal

[1]M.I.S Unit, Computer Science Department, Abubakar Tatari Ali Polytechnic, Bauchi, Bauchi State Nigeria

[2]Department of Mathematical Science, Abubakar Tafawa Balewa University, Bauchi, Bauchi State Nigeria

Corresponding Author: hussainidanazumi@gmail.com

ABSTRACT

This research work discusses the improvement of a specific consensus mechanism called Byzantine Fault-Tolerant (BFT) within a permissioned blockchain-based network designed for Vehicle-to-Everything (V2X) communication. The main objective is to make the consensus protocols of this network more robust in the face of attacks from malicious participants. A method to achieve this improvement by using a robust adversarial training technique known as the Fast Gradient Sign Method is suggested. This technique aims to counteract the negative effects of malicious nodes within the V2X network. The research work also delves into the recent advancements in the fields of distributed systems, blockchain technology, and consensus mechanisms. This exploration is conducted to improve and strengthen the security and reliability of communication networks used in V2X scenarios. The significance of ensuring the resilience and ability to withstand adversarial actions in environments involving vehicles, where the stability and resistance to malicious behavior are of utmost importance is emphasized. Our work has open future direction for interested researchers in this area.

Keywords: Intelligent Transportation, Distributed Systems, V2X Networks, Byzantine Fault Tolerance, Blockchain, Consensus Algorithms, Fast Gradient Sign Method (FGSM), Adversarial Attack,

INTRODUCTION

Intelligent transportation refers to the integration of advanced technologies, such as artificial intelligence, sensors, and communication networks, to create a more efficient, safe, and sustainable transportation system. This involves real-time data collection and analysis to optimize traffic flow, reduce congestion, enhance safety, and improve the overall user experience. Intelligent transportation systems (ITS) enable vehicles, infrastructure, and users to interact and share information, facilitating better decision-making and adaptive management of transportation networks (Njoku, et, al., 2023).

Distributed systems are a collection of independent computers that work together as a single coherent system to achieve a common goal. These systems offer improved reliability, scalability, and fault tolerance by distributing tasks and data across multiple nodes. Blockchain technology, as a solution within distributed systems, provides a decentralized and secure method of recording transactions and information. Its immutable and transparent ledger ensures data integrity and trustworthiness, making it ideal for applications requiring high levels of security and trust, such as financial services, supply chain management, and, increasingly, intelligent transportation systems (Vatter,, et, al., 2023).

DOI: 10.56892/bima.v8i2B.717

Vehicle-to-Everything (V2X) communication is a cutting-edge technology that enables vehicles to communicate with each other (V2V), with infrastructure (V2I), with pedestrians (V2P), and with networks (V2N). V2X aims to enhance road safety, traffic efficiency, and energy savings by providing real-time information exchange and coordinated responses among all elements of the transportation ecosystem. By incorporating blockchain technology into V2X networks, it is possible to enhance security, data integrity, and consensus among participants, addressing challenges such as data tampering and malicious attacks. This integration supports the development of robust, scalable, and efficient intelligent transportation systems that can better adapt to the dynamic needs of modern urban environments (Rehman, et, al., 2023).

This study contributes to the field by proposing an innovative method to enhance Byzantine-Fault-Tolerant (BFT) consensus in permissioned blockchain-empowered V2X networks through robust adversarial training using the Fast Gradient Sign Method (FGSM). By addressing the limitations of existing BFT solutions and improving fault detection and security in dynamic and complex distributed systems, the research enhances the reliability and safety of V2X communication networks. Additionally, the study's findings offer broader implications for improving consensus mechanisms in various permissioned blockchain applications, highlighting the potential of integrating advanced machine learning techniques to bolster blockchain security and performance.

Other section of the research includes: Section 2, Consensus Mechanism in Distributed System, Section 3, Methodology, Section 4, Related Works and Section 5, Conclusion and Future Direction.

CONSENSUS MECHANISM IN DISTRIBUTED SYSTEMS

In the field of computer science, a consensus algorithm serves as a mechanism to ensure unanimous agreement among disparate processes or systems regarding a specific data value. The primary aim of these algorithms is to establish reliability within a network that involves numerous users or nodes. This challenge, often referred to as the consensus problem, holds significant importance in distributed computing and multi-agent systems, including those found in Bitcoin blockchain networks (Awati, 2022).

Consensus is a fundamental concept not limited to blockchain but applies to various distributed systems. It arises in situations where multiple processes or nodes must converge on a single state for a data object. There are two main types of blockchain: permissioned and permission less. In permission less blockchain, nodes remain anonymous, and the addition of a modified block of transactions can lead to a fork. Forking occurs when a legitimate transaction conflicts with an invalid one. In contrast, the primary objective of the consensus algorithm is to achieve unanimity among nodes in a permissioned blockchain, where nodes are known entities and not anonymous (Chaudhry and Yousaf, 2018). It provides a categorization of different consensus algorithms used in distributed systems, which find extensive applications in decentralized computer networks, with blockchain being one of the most popular implementations. Consensus Distributed Algorithm can be either Distributed Consensus algorithm or Blockchain Consensus Algorithm. Figure 1 depicts the features of concensus algorithms. The figure illustrates, describes a novel classification system for consensus algorithms based on how they determine the order of system state changes. Figures illustrated this classification and the tradeoffs in scalability, decentralization, and security. The analysis revealed a common "choose-two" tradeoff among these concerns and identified two main types of consensus algorithms: leader-based and voting-based. The classification was applied to various distributed ledgers, including blockchains and DAGs, offering a framework for selecting suitable consensus algorithms for different distributed applications.

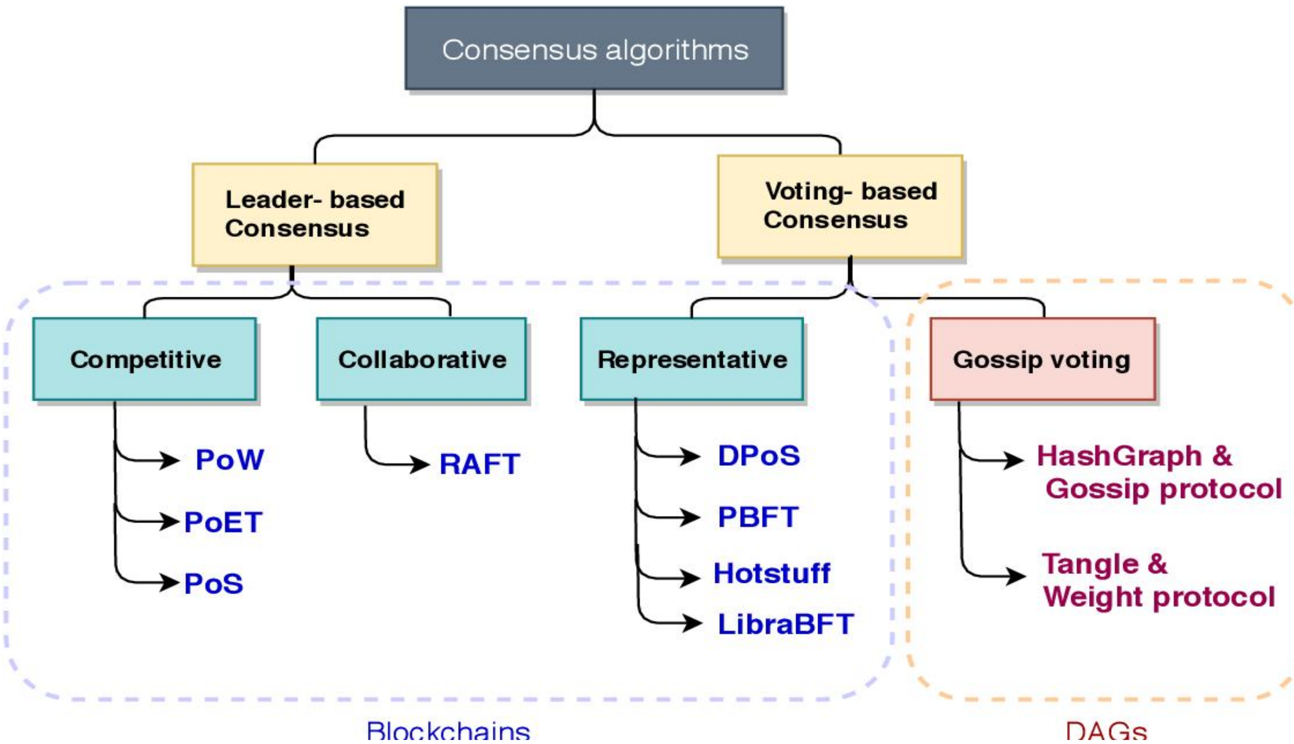

Figure 1: The features of the consensus algorithm in Distributed systemSource: (Altarawneh, et al., 2023).

Concensus Algorithm

Consensus algorithm in general is framed as a decision-making process where a group of people express their individual opinions to construct the decision which provides a best estimate of a process or system. Each member of the group expresses their opinion to support the decisions taken for a course of action. In simple terms, it is just a method to decide any event to occur within a group. Every one present in the group can suggest an idea, but the majority will be in favor of the one that helps them the most. Others have to deal with this decision whether they liked it or not. Byzantine Fault Tolerance (BFT), a problem of Byzantine General, is a system with a particular event of failure. One can experience best the aforementioned situation (BFT) with a distributed computer system. Many times, there can be malfunctioning consensus systems. These components are responsible for the further conflicting information. Consensus systems can only work successfully if all the elements work in harmony. However, if even one of the components in this system malfunctions the whole system could break down as seen in figure 1. These Blockchain consensus models are just the way to reach an agreement. However, there can’t be any decentralized system without common consensus algorithms. It won’t even matter

DOI: 10.56892/bima.v8i2B.717

whether the nodes trust each other or not. They will have to go by certain principles and reach a collective agreement. In order to do that, it is required to check out all the Consensus algorithms. It can be stated that versatility of blockchain networks is due to consensus algorithms. However, blockchain consensus algorithm may have pros and cons as shown in Table 1 below which can always alter the perfection of the algorithm (Rashmi et al.,2020).

Blockchain Consensus Algorithm

The underlying technology that forms the basis of Bitcoin is known as blockchain. This technology has been garnering increasing interest in various industries due to Bitcoin's rising value and its consistent and trustworthy operation. Blockchain is characterized by its decentralized nature, stability, security, and immutability. As a result, the network structure may evolve over time. The consensus algorithm plays a crucial role in upholding the security and efficiency of the blockchain system (Alazzam,, et al. 2023).

Table 1: The Blockchain Consensus Algorithm

Authors	**Algorithm**	**Designing Goal**	**Benefits**	**Drawback**
Jakobsson and Juels, (1999).	PoW	Sybil Proof	High-level security, decentralized verification	51% attack risks, time-consuming, resource consumption
King, and Nadal (2012).	PoS	Energy Efficiency	Energy efficiency, fast and cheap transaction process	Not really secured, large holdings validators have excessive influence on transaction verification, wary of 51% attack
Larimer, (2017)	DpoS	Organize PoS effectively	Good protection against double-spending, more democrats, required less power	The concentration of voting power, censorship resistance, delegate costs, etc
Castro and Liskov, (1999)	PBFT	Removing software error	Transaction finality, Energy efficiency, Low rewards variants	Low Scalability, liable to Sybil attack, compromising
Dziembowski, et al., (2015)	PoC	Less energy than PoW	30×more energy efficient than POW, save half time consume by POW and its can be use in general purpose	Storing data of hashes have a lot of free space, which make it hard to detect malicious computation storage.
Bentov, et al., 2014	PoA	Benefits of both PoS and PoW	Combines PoW hashing and PoSs digital signing, more	High energy consumption, takes long time to mine
Zheng, et al., 2017	Dbft	Faster PBFT	More efficient than pBFT, suitable for permission blockchain, effective countering unreliable participants	It requires regulated blockchain, which include a level centralization
NEM, (2018)	Pol	Improve PoS	Fast and power-efficient, scalable and safe	limits the amount of mining
Bitcoin-Wiki. (2018)	PoB	N/A	More efficient than PoW	Not friendly, slow verification process and investment risk

Distributed Consensus Algorithm

A distributed consensus algorithm is a protocol used in distributed systems to achieve agreement on a single data value among multiple processes or nodes. This is crucial for ensuring consistency and reliability in systems such as blockchain networks, distributed databases, and multi-agent systems. Consensus algorithms help maintain a unified state across all nodes despite potential failures or malicious actors. Common types include Proof of Work, Proof of Stake, and Byzantine Fault Tolerance, each with its own approach to validating transactions and securing the network. The choice of algorithm impacts the system's scalability, security, and decentralization, often requiring tradeoffs between these factors to meet specific application needs (Zhong, et al. 2023).

Gap in the Literature

The use of Reinforcement Learning (RL) for Byzantine-Fault-Tolerant (BFT) consensus in (Kim and Ibrahim, 2022) may introduce vulnerabilities to adversarial attacks. Adversarial Agents could exploit weaknesses in the RL-based consensus mechanism to manipulate the decision-making process, potentially causing disruptions or malicious behavior in the V2X network. If the RL model used in the consensus mechanism is not adequately trained or lacks robustness, it may become susceptible to exploitation, leading to biased decisions or unauthorized changes in the blockchain. In a permissioned blockchain-empowered V2X network, ensuring data privacy and confidentiality is crucial. If the RL-based consensus mechanism does not properly handle private information or secure communication channels, sensitive data could be exposed to unauthorized entities. Again, RL-based consensus mechanisms can be computationally expensive, potentially impacting the scalability and performance of the V2X network. Slow consensus times could lead to delays in transaction validation and block confirmation, affecting the overall efficiency of the system. Also, Permissioned blockchains may have a controlled set of participants, but the RL-based consensus mechanism should be resistant to Sybil attacks, where adversaries create multiple identities to gain influence or control over the consensus process. Finally, the presence of Byzantine nodes in a permissioned blockchain can disrupt consensus and compromise the integrity of the network. Ensuring robustness against Byzantine nodes is crucial to maintain the trust and reliability of the V2X network.

Fault Tolerance

Comparatively, to uniprocessors, fault tolerance investigates how a system responds to an unexpected hardware or software failure; distributed systems are difficult to isolate errors in. Failure detection and recovery are the two core elements that make up fault tolerance. When a problem occurs or if any of the system's components get disconnected or malfunction, it can be challenging to keep dispersed systems running. In the distributed paradigm, there are various fault tolerance strategies available, including retry, replication, checkpointing, and message logging, among others (Reghenzani, et, al., 2023). The communication model, the failure model, the dynamics model, and the timing model are the main issues to consider while modeling a distributed system. The communication model explains the interactions between the parts of a distributed system. The failure model outlines the types, frequency, and severity of system failures that could happen, such as when components abruptly stop functioning or behave irrationally or maliciously. The dynamics model describes how the distributed system changes over time in terms of the number of its components. Last but not least, the timing model connects system events to

DOI: 10.56892/bima.v8i2B.717

the passage of time, for instance by setting upper limits on computation time. Fault tolerance has four distinct phases as mentioned in (Indhumathi, et, al., 2023), which include; Error detection, damage confinement, error recovery, Fault treatment, and continued system service.

The ability of a distributed system to perform perpetually despite some component failure is a perfect selling point for the systems.

Faut Tolerance Techniques

Zhong, et al., (2023), categorized fault tolerance techniques into four (4), which includes; replication-based fault tolerance technique, Checking point/Roll back technique process level redundancy technique and fusion based technique. They discussed three techniques and summarized in the Table 2.

Table 2: Comparative Analysis of Different Fault-Tolerant Techniques in a Distributed System (Jiang and Yu, 2012)

Major Factors	Replication Based Technique	Checking Point/Roll Back Technique	Fusion Based Technique	Process Level Redundancy Technique
Working	Redirected to replica	State saved on stables to rage used for recovery	Back up machine	A set of processes redundant
Consistency	Some criterion; Linearizability	Avoiding orphan messages	Among backup machines	Not a major issue
Multiple Faults Handling	Depend upon number of replica.	Depend upon Check pointing scheduling	The back machine's effectiveness relies on the number of occurrences.	It relies on a collection of duplicate procedures.
Performance	As the number of replicas increases, reductions take place.	As the frequency increases, the size and frequency of the checkpoint decrease.	Reduced occurrence of faults due to the higher cost of recovery.	The frequency of faults decreases, resulting in their apparent disappearance.
N-Faults	N replicas ensure n-1 faults	Uncoordinated and pessimistic N-level disks are scarcely utilized, mainly reserved for N-1 fault tolerance.	To manage additional N faults, N backup machines are necessary.	Scaling up the process count while utilizing Majority voting
Multiple Failure Detector	Dependable, Precise, Flexible	Dependable, Precise, Flexible	Dependable, Precise, Flexible	Dependable, Precise, Flexible

Byzantine

The "Byzantine Generals Problem," a fictitious scenario that highlights the difficulties of reaching consensus in a network of unreliable or malicious nodes, inspired the name of the technique known as "Byzantine Fault Tolerance" (BFT), which is used in distributed computing systems to ensure the system remains operational and consistent even in the presence of faulty components or malicious actors. Nodes may provide erroneous data, transmit inconsistent signals, or fail to reply at all as examples of byzantine errors. The Practical Byzantine Fault Tolerance (PBFT) method was one of the first to achieve Byzantine fault tolerance. It was

DOI: 10.56892/bima.v8i2B.717

developed by Castro and Liskov and popularized in 1999 as a fundamental method for creating Byzantine fault-tolerant systems. (Zhang,et, al., 2020).

Ohri(2021) The classification divides fault tolerance into two main categories: hardware fault tolerance and software fault tolerance. Hardware fault tolerance, which includes methods like built-in self-test (BIST), Triple Modular Redundancy (TMR), and Circuit Breaker, is generally considered more straightforward to manage compared to software fault tolerance techniques. On the other hand, software fault tolerance involves more complex approaches, such as N-version programming, Recovery block, Checkpointing and roll-back recovery, Failure-Oblivious computing, and Recovery shepherding.

Table 3: The Early Fault Detection Techniques

Author	Technique	Process	Benefit	Limitation
Siewiore and Swarz, (1998).	Duplication	Copy of a component. In failure it compares the copies and detect the fault	Detects all faults except that of comparison elements	It requires more redundancy
Postma, (1998)	Error Detecting code	Code words with redundant bits that enable the sets of code word	Detects a mutilation of code words	
Postma, (1998)	Checksum	Form by adding *s* in modulo-*n*	Cheap in required redundancy	In take long time to detect fault
Postma, (1998)	Safe checking and fail-safe logic	Self-testing and fault-secure	Low redundancy	Relies on one gate in the circuit
Postma, (1998)	Watch dog timers and bus timeout	Time limit set for responds,	Effective methods since task fail in an infinite loop	Not every failure is detected
Postma, (1998)	Consistency and capability checks	Check for input result to be on a valid range	Use by most computers	Access to object is limited
Postma, (1998)	Processor monitoring	3 phases; Control flow, checking techniques and assertion checking	Success of the methods depends on the invariant of the application	Variable values in a set increase or decrease monotonically.

For decades, the practical application of fault tolerance detection methods in distributed systems has been lacking. However, recent advancements in Distributed techniques such as Grid Computing, Cloud Computing, Internet of Things, and blockchain technologies have expanded the scope of distributed techniques significantly. Notably, the most prominent Byzantine fault detection technique was introduced by Haeberlen et al. in 2006, leading to the evolution of various fault tolerance intrusion detection methods, particularly focused on Byzantine fault tolerance. Table 3 provides an overview of some of the latest innovative fault intrusion detection approaches.

Byzantine Fault Tolerance

Byzantine fault tolerance (BFT) is a crucial technique used to ensure the security and reliability of distributed systems. Error detection is an essential aspect of BFT, as it allows for the identification and removal of faulty nodes or components that may cause a failure in the system. There are several error detection techniques used in BFT systems, including redundancy, voting, and signature-based schemes (Zhong, et al., 2023). Redundancy involves duplicating critical components in the system to provide backups in case of failure. Voting schemes involve having multiple nodes independently verify the output of a component and reach a consensus on its correctness. Signature-based schemes use cryptographic signatures to verify the authenticity and integrity of messages and transactions (Smith and Poon, 2016). Despite the use of error detection techniques in BFT systems, they still face challenges in identifying and isolating Byzantine faults, particularly in large-scale and complex systems. The detection of Byzantine faults may also be complicated by malicious nodes that intentionally disrupt the system (Winter,, et, al., 2023).

To increase the precision and effectiveness of Byzantine fault detection in distributed systems, researchers have looked at the use of machine learning techniques like deep neural networks. These methods have produced encouraging results, but further study is required to confirm their efficacy and scalability in practical settings.

In addition to fail-stop and Slowdown classes of processor (node) faults, another significant failure that can occur is the Byzantine fault caused by a processor attempting to disrupt the calculation by either running slowly or at normal speed (Misra, et, al,2023). This fault is known as the Byzantine Mistake, and it refers to a computer system's ability to continue functioning even when some of its node's malfunction or behave maliciously. The origin of this term lies in the Byzantine Generals Problem, a logical puzzle centered around a group of Byzantine generals who must make a collective decision to attack or retreat. If they all agree, they will succeed in their mission, but if communication errors or acts of treachery cause some generals to attack while others retreat, the endeavor will fail. Such issues fall under the category of Byzantine faults.

The byzantine fault tolerance techniques are summarized with their characteristics, advantages, and disadvantages in Table 4.

Table 4: Summary of some Byzantine Fault Tolerance Techniques (Nasreen, et al., 2018)

Techniques	Criteria	Pron	Cons
Local Broadcast	The proportion of byzantine nodes in each neighborhood	A simple and efficient algorithm	Does not work on the grid
Explorer	The connectivity of a network with 2k + 1 Byzantine nodes.	Criteria on the topology of the network	Tolerates at most 1 byzantine failure on the grid
Control Zone	Byzantine node surrounded by a control zone	Tolerate many Byzantine'sfailurese with high probability	This necessitates a comprehensive understanding of worldwide information.
Trigger	Distance among Byzantine nodes.	Endure numerous Byzantine failures even in the absence of topology information.	Weaker performance than control zone
Fractal	Byzantine rate λ	As the diameter increases, the likelihood of communication being maintained also increases.	Only works for $\lambda > 10^{-5}$

Byzantine Fault Tolerance is significant from a computing standpoint because it ensures that a system can continue to run even if some of its components fail. Anything that employs a computing system, like an aircraft or a spacecraft, must be able to function even when some of its nodes are not operating at full capacity (Dren, 2022)

Fast Gradient Sign Method

Fast Gradient Sign Method (FGSM) Attack, (Hong, et al., 2024), This attack uses the gradient of the model's cost function to determine the direction in which the input data needs to be modified resulting in erroneous predictions. FGSM tends to be a relatively simple attack but is quite effective.

The information above is just a few examples of attacks where we chose the Fast Gradient Sign Method (FGSM) Attack as an Adversarial attack that has many potential implications and impacts(Jagadeesha, 2022), especially when applied to critical AI systems such as autonomous vehicles, security systems, or medical decisions and Vehicle to Everything Network . Therefore, research and efforts in understanding and countering adversary attacks continue to improve the security and reliability of AI systems and Network.

REVIEW OF RELATED LITERATURE

The threat of Denial-of-Service attacks on vehicular networks has been well-documented (Hashbullah et al.,2021). These attacks have the potential to congest the RF spectrum, block access to vital RF resources for vehicles, disrupt the transmission of crucial safety-related data between vehicles, and prevent vehicles from connecting to road-side units (RSUs), making them one of the most perilous threats to vehicular networks. Extensive research has focused on studying DoS attacks within the framework of DSRC. An evaluation conducted in (Pu'nal et al.,2020) assessed the performance of DSRC-based vehicular networks when subjected to jamming attacks. DoS attacks known as jamming attacks involve jammers disrupting legitimate signals to hinder receivers from properly demodulating them. The research examined the impacts of constant, periodic, and reactive jamming on DSRC devices within an anechoic chamber were reactive jamming only occurs if the attacker senses that an energy threshold on a certain band has been exceeded (Natasa et al.,2020).

CONCLUSION

The review explains the concept of the existing system and proposes a mechanism to enhance its security context. The existing system utilizes a Fabric network with RL-based algorithms for channel selection and online learning. The proposed mechanism incorporates adversarial training using the FGSM method to improve the model's robustness against adversarial attacks in the consensus mechanism. The study highlights the benefits and characteristics of distributed systems, such as scalability, fault tolerance, and transparency. Fault tolerance strategies, including retry, replication, checkpointing, and message logging, are discussed. The research contributes to the field of blockchain by exploring novel methods to enhance consensus mechanisms and demonstrates the potential of machine learning techniques in improving blockchain security. The proposed mechanism has potential applications beyond V2X networks, showcasing the versatility of the adversarial training approach in other permissioned blockchain scenarios. The proposed model will be measured using the Benchmarked perfomance metrics like the latency, block dissemination rate, convergence, scalability, and regret.

Future Work

Our work opens several directions for further research, including exploring the application of machine learning techniques to further enhance the security and fault tolerance of the V2X network, building on the concept of multitask learning. Investigating novel consensus algorithms and mechanisms that can offer improved scalability and efficiency for V2X networks. Incorporating secure and privacy-preserving data sharing mechanisms

DOI: 10.56892/bima.v8i2B.717

into the blockchain-enabled V2X network for enhanced We are currently investigating methods to reduce the effects of Sybil attacks and other advanced threats on the V2X communication system while focusing on data protection.

REFERENCES

Alharbi, S., Khan, M. K., Gupta, B. B., and Choo, K. K. R. (2021). A comparative study of Byzantine fault tolerance consensus algorithms for blockchain networks. Journal of Parallel and Distributed Computing, 151, 137-151.

Alharby, A., Rehman, M. H. U., and Muhammad, K. (2020). Distributed Byzantine fault tolerance in blockchain networks: A comprehensive survey. Journal of Network and Computer Applications, 166, 102687. https://doi.org/10.1016/j.jnca.2020.102687

Alharby, et al. (2021) "Deep reinforcement learning for blockchain: A review", Journal of Parallel and Distributed Computing, Vol. 147, pp. 59-72.

Ali, M., Islam, M. T., Hossain, M. S., and Alam, M. (2020). A Dynamic and Adaptive Reinforcement Learning Algorithm for Byzantine Fault Tolerance in Permissioned Blockchain-Empowered V2X Networks. IEEE Transactions on Intelligent Transportation Systems, 22(6), 3653-3663.

Bhatia, R., Sivakumar, D., and Shetty, S. (2020). Reinforcement Learning Based Byzantine Fault Tolerant Consensus in Blockchain. In 2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON) (pp. 53-59). IEEE.

Bhutta, M. N. M., Khwaja, A. A., Nadeem, A., Ahmad, H. F., Khan, M. K., Hanif, M. A., ... and Cao, Y. (2021). A survey on blockchain technology: Evolution, architecture and security. *Ieee Access*, *9*, 61048-61073.

Chen, X., Zhang, Y., Liu, Y., Liu, Z., and Vasilakos, A. V. (2020, November). A reinforcement learning approach for blockchain consensus. In Proceedings of the 11th ACM Symposium on Cloud Computing (pp. 563-569).

Dren, H., (2022, September 8). *What is BFT? Byzantine Fault Tolerance Explained*. Crypto Academy. https://crypto-academy.org/what-is-bft-byzantine-fault-tolerance-explained/

Huang, X., and Wu, Y. (2021). Blockchain Byzantine Fault Tolerance Consensus Algorithm Based on Improved Reinforcement Learning. IEEE Access, 9, 44872-44884.

Kim, S., and Ibrahim, A. S. (2022). Byzantine-Fault-Tolerant Consensus via Reinforcement Learning for Permissioned Blockchain-Empowered V2X Network. IEEE Transactions on Intelligent Vehicles.

Li, X., Zheng, Z., Chen, Y., Sun, X., and Wu, L. (2019). An Improved Byzantine Fault Tolerance Algorithm Based on Reinforcement Learning for Blockchain. IEEE Access, 7, 49705-49715.

Liu, S., Yu, Y., Ren, Y., and Zhang, Y. (2021). Dynamic weight adjustment-based practical Byzantine fault tolerance algorithm for V2X networks. International Journal of Distributed Sensor Networks, 17(4), 1550147721100

Liu, X., Shen, X., and Sun, X. (2019). A survey on reinforcement learning for blockchain networks. IEEE Access, 7, 52491-52503.

Lu, M., Chen, X., Wang, X., and Zhang, J. (2020). Dynamic Consensus: A Novel Byzantine Fault-Tolerant Consensus Algorithm for Blockchain Networks. IEEE Transactions on Services Computing, 13(1), 55-69.

DOI: 10.56892/bima.v8i2B.717

Park, J., Lee, S., and Kim, S. (2021). A distributed machine learning approach for Byzantine fault tolerance in blockchain networks. Future Generation Computer Systems, 117, 130-142.

Rezvy, A. M. A., Biswas, N. K., and Haque, S. (2018). On efficient Byzantine fault tolerance for vehicular ad-hoc networks. In Proceedings of the International Conference on Information and Communication Technology for Sustainable Development.

Rosebrock, A. (2021, March 1). *Adversarial attacks with FGSM (Fast Gradient Sign Method)*. PyImageSearch. https://pyimagesearch.com/2021/03/01/adversarial-attacks-with-fgsm-fast-gradient-sign-method/

TensorFlow, (2023). *Adversarial example using FGSM | Tensor Flow Core*. https://www.tensorflow.org/tutorials/generative/adversarial_fgsm

Wang, R., Liu, C., and Li, Q. (2021). A Review of Byzantine Fault Tolerance Consensus Mechanisms for Blockchain. IEEE Access, 9, 106287-106304.

Wu, R., Chen, J., and Zhang, J. (2021). Distributed Byzantine Fault Tolerance Algorithm for Permissioned Blockchain-Based V2X Networks. IEEE Transactions on Vehicular Technology, 70(2), 1421-1430.

Xu, J., Zhang, S., Wang, Y., and Li, W. (2020). A Deep Reinforcement Learning Approach to Byzantine Fault Tolerance in Blockchain Systems. IEEE Transactions on Network and Service Management, 17(2), 962-974.

Xu, K., Zhang, K., and Wang, J. (2021). A secure and private V2X system based on blockchain. IEEE Access, 9, 21638-21649.

Y. Zhang, et al. (2019). Empirical analysis of blockchain-based systems: A systematic mapping study. IEEE Access, 7, 69176-69190.

Yang, Y., Zhang, J., Li, J., and Chen, Y. (2020). A blockchain-based V2X network architecture for intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems, 21(8), 3313-3325.

Zhang, B., Zheng, K., and Xiang, W. (2021). Optimizing Byzantine Fault Tolerance in Vehicular Networks with Permissioned Blockchain. IEEE Transactions on Vehicular Technology, 70(2), 1378-1392.

Zhang, J., Wu, X., Zhang, Y., and Li, X. (2020). A Survey on Byzantine Fault Tolerance in Blockchain Systems. IEEE Access, 8, 144165-144184`

Zhang, X., Zhang, J., Li, J., Ma, H., and Zhang, W. (2020). A Novel Adaptive Byzantine Fault Tolerant Consensus Algorithm Based on Reinforcement Learning in Blockchain Network. IEEE Access, 8, 223756-223768.

Gao, S., Yu, T., Zhu, J., and Cai, W. (2019). T-PBFT: An EigenTrust-based practical Byzantine fault tolerance consensus algorithm. *China Communications*, *16*(12), 111-123.

Dhinesh Kumar et al,.2023: revolutionizing intelligent transportation systems with cellular vehicle-to-everything (c-v2x) technology: current trends, use cases, emerging technologies, standardization bodies,industry analytics and future directions**:** https://doi.org/10.1016/j.vehcom.2023.100638.

Altarawneh, A., Herschberg, T., Medury, S., Kandah, F., and Skjellum, A. (2023, January). Buterin's scalability trilemma viewed through a state-change-based classification for common consensus algorithms. In *2020 10th Annual*

DOI: 10.56892/bima.v8i2B.717

Computing and Communication Workshop and Conference (CCWC) (pp. 0727-0736). IEEE.

Alazzam, F. A. F., Salih, A. J., Mohd Amoush, M. A., and Khasawneh, F. S. A. (2023). The nature of electronic contracts using blockchain technology–currency bitcoin as an example. *Revista De Gestão Social E Ambiental*, *17*(5), e03330-e03330.

Zhong, W., Yang, C., Liang, W., Cai, J., Chen, L., Liao, J., and Xiong, N. (2023). Byzantine Fault-tolerant consensus algorithms: a survey. *Electronics*, *12*(18), 3801.

Reghenzani, F., Guo, Z., and Fornaciari, W. (2023). Software fault tolerance in real-time systems: Identifying the future research questions. *ACM Computing Surveys*, *55*(14s), 1-30.

Indhumathi, R., Amuthabala, K., Kiruthiga, G., Yuvaraj, N., and Pandey, A. (2023). Design of task scheduling and fault tolerance mechanism based on GWO algorithm for attaining better QoS in cloud system. *Wireless Personal Communications*, *128*(4), 2811-2829.

Winter, L. N., Buse, F., De Graaf, D., Von Gleissenthall, K., and Kulahcioglu Ozkan, B. (2023). Randomized testing of byzantine fault tolerant algorithms. *Proceedings of the ACM on Programming Languages*, *7*(OOPSLA1), 757-788.

Misra, A., and Kshemkalyani, A. D. (2023, January). Byzantine fault-tolerant causal ordering. In *Proceedings of the 24th International Conference on Distributed Computing and Networking* (pp. 100-109).

Hong, D., Chen, D., Zhang, Y., Zhou, H., and Xie, L. (2024). Attacking Robot Vision Models Efficiently Based on Improved Fast Gradient Sign Method. *Applied Sciences*, *14*(3), 1257.

H. Hasbullah, I. A. Soomro, and J. L. Ab Manan, "Denial of service (DOS) attack and its possible solutions in VANET," *World Academy of Science, Engineering and Technology*, vol. 65, no. 5, pp. 411–415, 2021.

O. Pu ´ nal, C. Pereira, A. Aguiar, and J. Gross, "Experimental charac- ˜ terization and modeling of RF jamming attacks on VANETs," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 524–540, Feb 2020.

Natasa Trkulja et al.,2020: Denial-of-Service Attacks on C-V2X Networks

N. Lyamin, D. Kleyko, Q. Delooz, and A. Vinel, "Real-Time Jamming\ DoS Detection in Safety-Critical V2V C-ITS Using Data Mining," *IEEE Communications Letters*, vol. 23, no. 3, pp. 442–445, mar 2019.

Jagadeesha, N. (2022). Facial Privacy Preservation using FGSM and Universal Perturbation attacks. 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), 46–52.

Rashmi et al.,2020: Consensus Algorithm: College of Computing and Information Technology, University of Bisha, Bisha, Saudi Arabia Mohammad Ayoub Khan