

BLOCKCHAIN-BASED AND ENERGY-AWARE ADAPTIVE TRUST MECHANISMS FOR IOT

ARDA HACİFEVZİOĞLU

ÖZYEGİN UNIVERSITY

JUNE, 2025

BLOCKCHAIN-BASED AND ENERGY-AWARE ADAPTIVE TRUST MECHANISMS FOR IOT

by
Arda Hacifevzioğlu

Thesis
Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in
Computer Science

Advisor: Asst. Prof. Kübra Kalkan Çakmakçı

Graduate School of Science and Engineering
Özyeğin University
İstanbul

June, 2025

BLOCKCHAIN-BASED AND ENERGY-AWARE ADAPTIVE TRUST MECHANISMS FOR IOT

Approved by:

Asst. Prof. Kübra Kalkan Çakmakçı, Advisor
Department of Computer Science
Özyeğin University

Asst. Prof. İsmail Arı
Department of Computer Science
Özyeğin University

Prof. Fatih Alagöz
Department of Computer Science
Boğaziçi University

Approval Date: May 13, 2025

To my mom and dad, thank you for always believing in me.



DECLARATION OF ORIGINALITY

I hereby declare that I am the sole author of this thesis and that this is the true copy of my thesis, including the final revisions, approved by my thesis committee. All data and information have been obtained, produced, and presented in accordance with the rules of research ethics and principles of academic honesty. As required by these rules, to the best of my knowledge I have acknowledged ideas, thoughts, and any copyrighted material in accordance with the standard referencing rules. I certify that any part of this thesis has not been submitted for a degree or diploma in another educational institution.

Arda Hacifevzioğlu

ABSTRACT

Modern Internet of Things (IoT) increasingly rely on distributed decision-making, computation, and communication between independent nodes. As these systems expand and diversify, particularly when central coordination is not possible or not acceptable, decentralization becomes the primary concept. Decentralized systems eliminate dependence on central authority or trusted third parties by distributing decision-making and verification tasks among independent members of a network. While this shift promises greater scalability and fault tolerance, it introduces a critical challenge: establishing trust among resource-constrained and anonymous devices with no prior relationship or shared context. Blockchain technology has emerged to create a fundamental infrastructure to address this challenge by enabling transparent, auditable, and tamper-resistant management of data and consensus management in decentralized environments. Its distributed ledger mechanism ensures that transactions and state changes are immutably recorded, ensuring accountability in trustless networks. However, the presence of blockchain alone does not guarantee trustworthiness, especially when consensus mechanisms fail to account for the behavior, honesty, or sustainability of participating nodes. In this context, trust management becomes essential to maintaining the security and reliability of decentralized IoT networks. Trust can generally be broadly defined as the confidence in an entity's behavior based on past interactions, feedback, and reliability in service delivery. Modern trust models, ranging from reputation-based to computational and voting-based schemes, attempt to quantify this notion, yet most fall short in dynamic and malicious conditions. Despite extensive research, many existing trust models rely on static scores, stake-based voting, or unweighted historical data, making them vulnerable to adaptive adversaries and resource exhaustion. Especially for large-scale and heterogeneous IoT networks, lack of

adaptive and behavior-aware trust evaluation diminishes the overall security and reliability. To address these limitations, this thesis introduces a series of progressively enhanced trust-aware models, each developed based on the insights gained from its predecessors:

- TrustVista introduces a layered trust evaluation model that calculates trust at the node, group, and community perspectives. It emphasizes recent behavior by using a weighted average that gives more importance to recent interactions, and applies statistical techniques to detect abnormal or potentially malicious behavior. By evaluating trust not just individually, but in a collaborative and structural context, TrustVista improves the speed and accuracy of identifying harmful participants. During this stage, it was observed that mechanisms like Delegated Proof of Stake (DPoS) could be a viable consensus mechanism for IoT, yet lacked any awareness of nodes' energy capacity.
- Based on this insight, EDPoS was developed to extend classical DPoS by incorporating energy-awareness. EDPoS enhances the classical DPoS mechanism by introducing energy-awareness into the consensus process. It dynamically adjusts how many nodes can become validators, how much data is included in each block, and which nodes are eligible to participate, based on how much energy nodes have and how efficiently they've used it in the past. This reduces the likelihood of block validation failures due to node dropouts and extends the overall system lifetime in IoT environments where resources are limited.
- Building on EDPoS, TEDPoS further integrates behavioral trust into the energy-aware DPoS framework. TEDPoS proposes a dynamic validator scoring method that combines three key factors: a node's trust level based on its behavior, its available energy, and the level of community support it receives through voting. Unlike traditional DPoS, TEDPoS adapts its selection process based on real-time network

conditions, favoring nodes that are both trustworthy and resource-sufficient. This enables robust, context-aware, and attack-resilient delegate selection in decentralized IoT environments.

- As a supplementary use-case, TEMBUV, presented in Appendix A, presents how trust and blockchain can jointly support secure operations in post-disaster communication networks. While not simulated in OMNeT++, this model offers a mathematical abstraction of trust-based decision-making in UAV and VANET-integrated domains.

All proposed models are implemented and tested against various adversarial conditions, including camouflage, bad-mouthing, and self-promotion attacks. Experimental results demonstrate clear improvements in malicious node detection, energy efficiency, and consensus reliability compared to baseline models. Collectively, the frameworks presented in this thesis contribute to the development of adaptive, secure, and scalable trust mechanisms for decentralized and heterogeneous IoT systems.

Keywords: Blockchain, Trust Management, Consensus Mechanism, Energy Efficiency, Adaptive Consensus

ÖZET

Modern Nesnelerin İnterneti (IoT), giderek daha fazla şekilde, bağımsız düğümler arasında dağıtık karar alma, hesaplama ve iletişime dayanmaktadır. Bu sistemler genişledikçe ve çeşitlendikçe, özellikle merkezi koordinasyonun mümkün ya da tercih edilir olmadığı durumlarda, merkeziyetsizlik temel bir kavram hâline gelir. Merkeziyetsiz sistemler, karar alma ve doğrulama görevlerini ağdaki bağımsız üyeler arasında dağıtarak merkezi otoriteye veya güvenilir üçüncü taraflara olan bağımlılığı ortadan kaldırır. Bu dönüşüm daha yüksek ölçeklenebilirlik ve hata toleransı vaat etse de, önceden ilişkisi veya ortak bağlamı olmayan, kaynak kısıtlı ve anonim cihazlar arasında güven oluşturma gibi kritik bir sorunu da beraberinde getirir. Bu sorunu ele almak amacıyla, blockchain teknolojisi şeffaf, denetlenebilir ve kurcalamaya karşı dayanıklı veri yönetimi ile merkeziyetsiz ortamlarda mutabakat sağlama imkânı sunarak temel bir altyapı olarak ortaya çıkmıştır. Kriptografik olarak birbirine bağlı bloklara, dağıtık mutabakat mekanizmalarına ve değiştirilemez defterlere dayalı yapısıyla blockchain, işlemlerin ve durum değişikliklerinin geri döndürülemez biçimde kaydedilmesini sağlayarak merkezi olmayan ağlarda hesap verebilirliği mümkün kılar. Ancak, blockchain'in varlığı tek başına güvenilirlik garantisi sağlamaz. Özellikle de mutabakat mekanizmaları, katılımcı düğümlerin davranışlarını, dürüstlüğü veya sürdürülebilirliğini dikkate almadığında bu güvenilirlik zayıflar. Bu bağlamda, güven yönetimi, merkeziyetsiz IoT ağlarının güvenliği ve sürekliliği için vazgeçilmez hâle gelmiştir. Güven genel olarak, bir varlığın geçmiş etkileşimleri, geri bildirimleri ve hizmet sunumundaki güvenilirliği temel alınarak davranışına duyulan inanç olarak tanımlanabilir. Modern güven modelleri; itibar temelli, hesaplama dayalı veya oylama odaklı yöntemler kullanarak bu kavramı nicel hâle getirmeye çalışır. Ancak, çoğu model dinamik ve kötü niyetli koşullarda yetersiz kalmaktadır. Geniş çaplı ve heterojen IoT ağları özelinde, bu modellerin çoğu statik puanlara, hissedarlık (stake) temelli

oylamalara veya ağırlıksız tarihsel verilere dayanmakta olup, uyarlanabilir düşmanlara ve kaynak tüketimine karşı savunmasızdır. Uyarlanabilir ve davranışa duyarlı güven değerlendirmesinin eksikliği, genel güvenlik ve sistem güvenilirliğini ciddi şekilde zayıflatır. Bu sınırlamaları aşmak için, bu tezde kademeli şekilde geliştirilen ve birbirlerinden elde edilen içgörüler doğrultusunda tasarlanan üç farklı güven odaklı model sunulmuştur:

- TrustVista, düğüm, grup ve topluluk düzeylerinde güveni hesaplayan çok katmanlı bir güven değerlendirme modeli sunar. Son zamanlardaki davranışlara daha fazla ağırlık veren bir ortalama yöntemi kullanır ve anormal ya da potansiyel olarak kötü niyetli davranışları tespit etmek için istatistiksel teknikler uygular. Güveni yalnızca bireysel olarak değil, aynı zamanda işbirlikçi ve yapısal bir bağlamda değerlendiren TrustVista, zararlı katılımcıların daha hızlı ve doğru şekilde tespit edilmesini sağlar. Bu aşamada, Delegated Proof of Stake (DPoS) gibi mekanizmaların IoT için uygulanabilir olabileceği, ancak enerji farkındalığı taşımadığı gözlemlenmiştir.
- Buna dayanarak geliştirilen EDPoS, klasik DPoS mekanizmasını enerji farkındalığı ile genişletmektedir. EDPoS, hangi düğümlerin doğrulayıcı olabileceğini, her blokta ne kadar veri işleneceğini ve katılım uygunluğunu, düğümlerin mevcut enerji seviyeleri ve önceki verimliliklerine göre dinamik olarak ayarlamaktadır. Bu yapı, doğrulayıcı düğümlerin aniden devre dışı kalmasından kaynaklanan blok doğrulama hatalarının önüne geçer ve kaynak kısıtlı IoT ortamlarında sistem ömrünü uzatır.
- TEDPoS, EDPoS üzerine inşa edilmiş ve davranışa dayalı güveni enerji farkındalığı ile bütünleştiren bir yapı sunar. TEDPoS, bir düğümün davranışına dayalı güven puanını, sahip olduğu enerjiyi ve topluluktan aldığı oyları birleştirerek dinamik bir doğrulayıcı puanlama yöntemi önerir. Geleneksel DPoS'tan farklı olarak, TEDPoS seçme sürecini gerçek zamanlı ağ koşullarına göre uyarlar ve hem güvenilir hem de

kaynak açısından yeterli olan düğümleri önceliklendirir. Bu yaklaşım, merkeziyetsiz IoT ortamlarında bağlamsal farkındalık, saldırı dayanıklılığı ve sürdürülebilir doğrulama sağlar.

- Ek olarak, Appendix A bölümünde sunulan, TEMBUV adlı model, hava ve kara birimlerinin ayrı blokzincirlerle koordine edildiği çift katmanlı bir mimariyi önerir. Güven ve davranış skorlarına dayalı kredi bazlı bir oylama sistemi üzerinden tanık seçimi gerçekleştirilir. Bu model, domain tabanlı ağ ayrışması için alternatif bir yapı sunmaktadır. OMNeT++ ortamında simüle edilmeyen TEMBUV, UAV ve VANET entegre yapılarında güvene dayalı karar alma sürecini matematiksel bir soyutlamayla açıklamaktadır.

Sunulan tüm modeller, kamufraj, karalama (bad-mouthing) ve kendini yükseltme (self-promotion) gibi çeşitli saldırı senaryolarına karşı OMNeT++ ortamında uygulanmış ve test edilmiştir. Deneysel sonuçlar, zararlı düğümlerin tespiti, enerji verimliliği ve muhtabakat güvenilirliği gibi ölçütlerde temel modellerin ötesinde net iyileştirmeler ortaya koymuştur. Bu tez kapsamında geliştirilen sistemler, merkeziyetsiz ve heterojen IoT sistemleri için güvenli, ölçeklenebilir ve uyarlanabilir güven mekanizmaları oluşturma konusunda önemli katkılar sunmaktadır.

Anahtar Kelimeler: Blokzincir, Güven Yönetimi, Konsensüs Mekanizması, Enerji Verimliliği, Uyarlanabilir Konsensüs

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my thesis advisor, Assistant Professor Kübra Kalkan Çakmakçı, for her constant support, guidance, and encouragement during my graduate studies. She is the person who first introduced me to the fields of blockchain and trust management, and more importantly, helped me find a sense of purpose and passion within them. Her insightful advice and constant encouragement allowed me to grow not only as a researcher but also as a person. From idea generation to implementation, from paper writing to academic presentations, she has taught me how to conduct meaningful research with academic rigor and integrity. I am truly thankful for her trust in me, her tireless support, and her belief in my potential, which continuously motivated me to aim higher and persevere even during the most difficult times.

I would also like to sincerely thank the members of my thesis committee, Assistant Professor İsmail Arı and Professor Fatih Alagöz, for their valuable time, feedback, and insightful comments, which have helped shape the final version of this work.

Lastly, and most profoundly, I owe everything to my family. I would like to express my heartfelt thanks to my mother, whose unwavering emotional support, care, and love have given me the strength to stand tall during all phases of my life. She has always been the one to remind me of my worth, especially in moments of self-doubt. Her encouragement has been a constant source of comfort. To my father, there are no words that can fully describe the depth of my gratitude. He stood by me every step of the way, not only as a parent, but as someone who genuinely carried this burden with me. His extraordinary support, wisdom, and sacrifices made this thesis possible in ways beyond explanation. He is not only the backbone of this work, but also of everything I have achieved in life. This thesis is as much his as it is mine.

TABLE OF CONTENTS

DECLARATION OF ORIGINALITY	iv
ABSTRACT	v
ÖZET	viii
ACKNOWLEDGEMENTS	xi
LIST OF TABLES	xv
LIST OF FIGURES	xvi
LIST OF ACRONYMS AND ABBREVIATIONS	xviii
1 INTRODUCTION	1
1.1 Motivation and Problem Statement	2
1.2 Scope and Contributions	3
1.3 Thesis Organization	5
2 BACKGROUND AND RELATED WORKS	7
2.1 Internet of Things (IoT)	7
2.2 Blockchain	8
2.3 Trust	10
2.4 Literature Review	11
2.5 Trust and Energy Related Models	14
2.6 Trust Models: Centralized, Semi-Centralized, Decentralized	16
2.6.1 Centralized Trust Models	16
2.6.2 Semi-Centralized Trust Models	17
2.6.3 Decentralized Trust Models	18
2.7 Summary of Gaps in the Literature	20
3 TRUSTVISTA: NODE, GROUP AND COMMUNITY BASED PERSPECTIVE, BLOCKCHAIN UTILIZED TRUST MECHANISM FOR IOT	22
3.1 Overview of TrustVista Architecture	24

3.2	System Model: Node, Group, and Community-Based Trust Modeling	25
3.2.1	Node Perspective	26
3.2.2	Group Perspective	27
3.2.3	Community Perspective	29
3.3	Trust Calculation and Evaluation	30
3.3.1	Trust Score Computation Framework	30
3.3.2	Node Trust Score Calculations.....	33
3.3.3	Multi-Factor Dynamic Decay Function	34
3.3.4	Group Trust Score Calculation.....	35
3.3.5	Community Trust Score Calculation	36
3.4	Trust Assessment and Decision Making	36
3.5	NTDM: Node Trust Decision Mechanism	39
3.6	Simulation Setup	44
3.7	Simulation Results and Evaluation	45
3.7.1	Attack Evaluations	45
3.7.2	Decay Evaluations.....	49
3.7.3	System Evaluations.....	50
4	EDPOS: ENERGY-AWARE DELEGATED PROOF OF STAKE RESOURCE- CONSTRAINED DEVICES.....	54
4.1	Overview of EDPoS Architecture	54
4.2	Adaptive Block Size Adjustment	57
4.3	Adaptive Energy Threshold.....	59
4.4	Adaptive Delegate Count	61
4.5	Historical Energy Consumption Analysis.....	65
4.6	Simulation Setup	66
4.7	Simulation Results and Evaluation	70
4.7.1	Adaptive Delegate Count.....	70
4.7.2	Adaptive Energy Threshold	72
4.7.3	Adaptive Block Size Adjustment	74

5	TEDPOS: TRUST AND ENERGY AWARE DELEGATED PROOF OF STAKE	78
5.1	System Model: TEDPoS Architecture	80
5.1.1	Network Assumptions and Architecture	80
5.1.2	Trust and Energy Computation Framework	83
5.1.3	Node Evaluation and Behavior Profiling	88
5.1.4	Delegate Selection Process in TEDPoS	90
5.2	Simulation Setup	94
5.3	Simulation Results and Evaluation	94
6	CONCLUSION AND FUTURE WORK	110
6.1	Summary of Contributions	112
6.2	Suggestions for Future Research	113
	REFERENCES	114
	APPENDICES	120
	APPENDIX A — TEMBUV: TRUST ENSURING MECHANISM BASED ON BLOCKCHAIN USING UAVS AND VANETS	121

LIST OF TABLES

Table 2.1: Analysis of various trust models	11
Table 3.1: Simulation Metrics for TrustVist	44
Table 4.1: Simulation Metrics for EDPoS Consensus Mechanism	67
Table 5.1: Simulation Metrics	95
Table 5.2: Consensus Models: Performance Summary	106
Table A.1: Simulation Parameters for TEMBUV Evaluation	125

LIST OF FIGURES

Figure 3.1: Node-Centric Interaction in TrustVista	27
Figure 3.2: Group perspective in TrustVista	29
Figure 3.3: Life cycle of a malicious node from initialization to termination in TrustVista.	39
Figure 3.4: General structure and trust management workflow. (a) Block addition process via <i>NTDM</i> . (b) Node <i>U</i> 's removal due to malicious behavior. ..	40
Figure 3.5: Comparison of model responses when a node experiences two consecutive camouflage attacks from the same malicious node in TrustVista... ..	46
Figure 3.6: Comparison of system-wise accuracy <i>ACC</i> under different attack scenarios (a), (b), (c), (d) and MNRs for TrustVista, PeerTrust [1] and Semi-Centralized Trust Management Model [2].	47
Figure 3.7: Impact of varying λ rates on <i>NTS</i> in regular, recovery, attack and no transaction periods in TrustVista.	49
Figure 3.8: Comparison of the decay functions' impact on the <i>NTS</i> in an attack-free environment between TrustVista and Semi-Centralized Trust Management Model [2].	50
Figure 3.9: Comparison of attack reduction during bad-mouthing attacks, from system initialization to termination.	51
Figure 3.10: Malicious node elimination over number of transactions for a 50% MNR under a bad-mouthing attack scenario.	52
Figure 3.11: Impact of malicious node reduction on transaction speed under bad-mouthing attack.	52
Figure 4.1: EDPoS structure	55
Figure 4.2: Dynamic Round Robin structure and illustration of node replacement ..	55
Figure 4.3: Number of Blocks Added for EDPoS Tests	70
Figure 4.4: System Longevity for EDPoS Tests	71
Figure 4.5: Average Remaining Energy (%) Over Rounds for EDPoS Tests	72
Figure 4.6: Variance of Energy Consumption per Round for EDPoS Tests	73

Figure 4.7: Average Throughput per Round for EDPoS Tests	74
Figure 4.8: Number of Transactions per Block for EDPoS Tests	75
Figure 4.9: Block Validation Failure Rate for EDPoS Tests.....	77
Figure 5.1: TEDPoS Workflow.....	82
Figure 5.2: Average trust score evolution of honest and malicious nodes over time in 50% MNR environment.	98
Figure 5.3: Malicious delegate ratio across 150 rounds in 50% MNR environment.	99
Figure 5.4: Detection rates of camouflage (50% camouflage rate), self-promotion, and bad-mouthing attacks across varying MNR, comparing TEDPoS, EDPoS 4 and DPoS.	100
Figure 5.5: TECS comparison of TEDPoS, Semi-Centralized [2], EDPoS 4 and DPoS under different attack types.	103
Figure 5.6: Average remaining energy levels over 150 rounds, showing the trade- off between energy efficiency and security across consensus models....	104
Figure 5.7: Standard deviation of node energies over time, showing load balance and fairness in energy consumption across consensus models.	105
Figure 5.8: Block validation failure rate over time, highlighting sustainability and load-balancing efficiency across consensus models.....	105
Figure 5.9: Block finality delay trends over time, showing latency convergence and performance optimization across consensus models.	107
Figure 5.10: System longevity.....	108
Figure A.1: System Model of TEMBUV	122
Figure A.2: System comparison among different camouflage ratios	127
Figure A.3: System comparison among different malicious supportive and dis- crediting attack ratios.....	128
Figure A.4: System comparison among different non-discrediting malicious sup- portive attack ratios.....	130
Figure A.5: System comparison among different malicious collaborative attack collectives number	130

LIST OF ACRONYMS AND ABBREVIATIONS

PoW	Proof of Work
PoS	Proof of Stake
EDPoS	Energy-Aware Delegated Proof of Stake
MNR	Malicious Node Ratio
ACC	Accuracy
SWR	Secure Witness Ratio
TS	Trust Score
NTDM	Node Trust Decision Mechanism
EWMA	Exponentially Weighted Moving Average
NTS	Node Trust Score
GTS	Group Trust Score
CTS	Community Trust Score
DyRT	Dynamic Removal Threshold
P2P	Peer-to-Peer
CS	Consensus Score
IoT	Internet of Things
CPS	Cyber-Physical Systems
RSU	Road Side Unit

1. INTRODUCTION

The increasing reliance on distributed infrastructures in modern digital systems has shifted the basic research thrust towards decentralization. From blockchain platforms or large-scale peer-to-peer networks, the absence of central control mechanisms offers both potential and risk. Decentralized systems offer improved scalability, resilience, and autonomy but, simultaneously, raise a very critical question: how to establish trust and reliability in an environment that comprises independently acting and largely unknown nodes. Blockchain has become widely adopted as a key technology, ensuring immutability and verifiability in decentralized networks. Through secure consensus among untrusted parties through distributed ledgers and cryptographic protocols, transforming how trust is built in systems. However, the effectiveness of any blockchain is closely tied to the design of its consensus protocol. Among existing methods, Delegated Proof of Stake (DPoS) stands out for its performance and scalability. DPoS solutions, however, are purely voting- or token ownership-based, with no consideration for the behavioral, energy, or reliability state of involved nodes. The notion of trust, therefore, becomes central to the efficiency and integrity of decentralized ecosystems. Trust has been widely described as the anticipation of a participant to act honestly and dependably in a particular encounter, derived from previous experience. Most systems present today have attempted to model trust via static reputation scores or rule-based heuristics. However, in dynamic environments, where the actions of nodes may vary over time, or coordinated malicious activity is possible, such static approaches are insufficient. Delayed identification of malicious nodes, unfair selection of block validators, and inefficient use of network resources are just a few consequences of outdated trust models.

1.1 Motivation and Problem Statement

The increasing adoption of decentralized architectures has highlighted the need for trust evaluation mechanisms out of static assumptions and stake-based voting into prominence. In dynamic networks with autonomous, heterogeneous nodes, robust consensus is dependent not only on participation but also on the quality, consistency, and sustainability of the behavior of each individual node. Existing systems cannot distinguish between malicious and honest participants in a timely manner, leading to degraded performance, biased validator choice, and security risks. These problems necessitate trust-aware and adaptive consensus methods that can sustain performance under adversarial and resource-constrained settings. Traditional consensus protocols—most notably Delegated Proof of Stake (DPoS)—are primarily based on token-based voting or static reputation scores. These mechanisms assume that participants with more votes or historical credibility will act honestly and efficiently. However, in real-world decentralized networks, this assumption fails due to several key limitations:

- **Lack of real-time trust assessment:** Nodes are often selected based on outdated or unverified metrics, making the system vulnerable to malicious actors who manipulate their standing through techniques like self-promotion or bad-mouthing.
- **No energy-awareness:** Validator roles are assigned without considering the node's energy status or historical efficiency, resulting in frequent block production failures or early node dropouts in constrained environments.
- **Static or one-dimensional evaluation:** Trust is usually evaluated from a single perspective (e.g., direct interaction or total votes), ignoring the broader social context or group behavior patterns.

As a result, unreliable or malicious nodes may remain active in the network for extended periods, degrading trustworthiness, slowing convergence, and weakening the resilience

of the consensus process. There is a critical need for consensus models that integrate dynamic trust evaluation, multi-layered behavior analysis, and resource-aware filtering to ensure that only capable and trustworthy nodes participate in critical consensus functions.

1.2 Scope and Contributions

This thesis addresses the design and analysis of novel blockchain-enabled trust and consensus models for dynamic decentralized networks. Specifically, the study focuses on environments that require node participation, trustworthy communication, and resilient decision-making mechanisms in the absence of pre-fixed trust relationships. First, it explores the use of multi-perspective trust evaluation, embracing node, group, and community-level behavior, can be leveraged to improve network resilience. Furthermore, it looks at integrating behavioral feedback, vote dynamics, and energy consumption into efficient but lightweight consensus protocols. It presents a comparative simulation-based modeling of the proposed models under various attack scenarios, with a focus on detection accuracy, secure node election, and resilience to adversarial behaviors. The key contributions of this thesis are as follows:

- **TrustVista**

- A three-perspective trust model for localized and overall trust assessment.
- Employs Exponentially Weighted Moving Average (EWMA) for prioritizing recent interactions in trust updates.
- Implements Z-score-based anomaly detection to identify and isolate malicious or suspicious nodes using statistical deviation from group behavior.

- **EDPoS**

- Enhances the DPoS protocol with energy-aware mechanisms including adaptive block size, dynamic delegate count, and candidacy thresholds.

- Introduces historical energy trend analysis to improve validator reliability and extend network lifetime.
- Prevents low-energy nodes from participating in consensus roles, reducing block validation failure rates.

- **TEDPoS**

- Proposes a dynamic delegate scoring mechanism that jointly considers trust levels, energy status, and voting support for validator selection.
- Introduces an adaptive scoring strategy that adjusts node evaluation based on current network conditions, improving consensus reliability and sustainability.
- Includes a temporal trust update function and energy-based scaling logic to reflect recent behavior and operational capacity. Also, Trust-Energy Composite Score (TECS) is presented, a unified metric that enables fair comparison across models by jointly quantifying security and sustainability.

- **Comparative Simulation and Evaluation**

- Implements all proposed models in a discrete simulation environment using OMNeT++ and evaluates them under various adversarial conditions such as camouflage, bad-mouthing, and self-promotion.
- Demonstrates improvements in malicious node detection accuracy, fairness in delegate selection, and energy efficiency over DPoS and reputation-based models.

- **TEMBUV – Supplementary Trust Coordination Architecture**

- Presents a dual-layer blockchain architecture (air and ground domains) as an appendix contribution.

- Uses a credit-based voting system and domain-separated trust scoring to coordinate nodes across different communication layers.
- Offers a flexible design for layered or hierarchical networks requiring domain-specific trust isolation.

Collectively, these contributions are designed to provide an adaptive and secure infrastructure to be deployed in unpredictable, resource-constrained, and mission-critical post-disaster data exchange networks.

1.3 Thesis Organization

This dissertation is structured organized in chapters, each of which builds upon the previous to progressively address the trust and consensus challenges in UAV-aided, decentralized networks for disaster scenarios.

- **Chapter 1** introduces the overall problem, discusses the motivation for the research, and states the scope and contributions of the thesis.
- **Chapter 2** provides a comprehensive background on trust in decentralized systems and blockchain-based architectures. It also includes a detailed literature review of existing trust models and consensus mechanisms, highlighting their limitations in dynamic, large-scale, and heterogeneous environments.
- **Chapter 3** introduces TrustVista, a hierarchical trust management framework that evaluates node behavior at multiple levels, such as node, group, and community, while using statistical decay and adaptive consensus.
- **Chapter 4** describes EDPoS mechanism, which substitutes trust with a dual focus on energy level and vote count to ensure lightweight yet robust delegate selection.

This chapter outlines the delegate election and block validation and examines the behavior of the system under different adversarial attacks.

- **Chapter 5** presents TEDPoS. In order to improve the sustainability and reliability to validator election for decentralized blockchain networks, TEDPoS is a novel consensus mechanism. TEDPoS combines three essential aspects, trust, energy, and voting, as opposed to conventional DPoS systems that solely utilize voting or stake. The model evaluates the recent behavior of each node based on a dynamically calculated trust score, considers its residual energy capacity for long-term contribution, and incorporates community vote support. TEDPoS does not employ static thresholds but adapts dynamically to network conditions and removes untrusted or energy-drained nodes. The resilience of the network against adversarial nodes is drastically improved by TEDPoS, which also increases the efficiency of block validation by giving priority to honest and resource-efficient participants in the consensus process.
- **Chapter 6** summarizes the thesis by summarizing the key findings and suggesting directions for future work, such as real-world deployment issues and incorporation of trust models with physical-world sensing capability.

2. BACKGROUND AND RELATED WORKS

The growing popularity of decentralized technologies such as blockchain, dynamic trust systems, and distributed consensus has led to numerous solutions attempting to address the problem of trust and security in large-scale, heterogeneous networks. In the context of operating in environments where the nodes to participate may be unknown, dynamic, or resource-constrained, trusting assessment and consensus remains an imperative requirement for system integrity and resilience.

This chapter provides the background required to understand the models addressed in this thesis. It overviews key concepts like trust management in distributed systems, blockchain-based architecture, and common trust and consensus mechanisms. Additionally, it provides a literature overview of key research in the subject, describes their limitations towards reacting to dynamic and adversarial settings, and rationalizes the need for more adaptive, trust-aware consensus models proposed in this work.

2.1 Internet of Things (IoT)

The Internet of Things (IoT) [3] is a rapidly expanding model of computing in which common physical objects are embedded with sensing, processing, and communication capabilities so that they can communicate and interact with one another and with centralized or distributed facilities on their own. From smart cities [4] and industrial automation [5] to precision agriculture and disaster response, IoT is revolutionizing data gathering, analysis, and action in both civilian and mission-critical uses. By current estimates, billions of devices [6] are being networked worldwide, often operating

in resource-limited, heterogeneous, and highly dynamic environments. While this distributed intelligence promises unprecedented automation and scalability, it poses fundamental challenges towards system coordination, energy efficiency, data integrity, trustworthiness, and resilience. Among these challenges, energy efficiency is a critical issue [7]. Most of the IoT nodes are powered either by batteries or energy harvesting, and therefore it is essential that any communication or consensus protocol be as energy-constrained as feasible without compromising on performance and security. At the same time, data privacy [8] is also a comparable threat because IoT devices are often handling sensitive personal or operational data, loss of which can lead to catastrophic confidentiality violations or even physical safety. Traditional centralized models typically don't apply or work in such environments due to connectivity constraints, single points of failure, and a lack of scalable, context-aware trust management. In addition, unlike traditional network architectures, the majority of IoT deployments lack static infrastructure or central control, particularly in ad-hoc, edge, or disaster recovery contexts, where devices must coordinate independently in uncertain settings. The decentralized IoT environment further amplifies the risk of malicious activity [9], including impersonation, data poisoning, and Sybil attacks. Enabling devices to make trustworthy, privacy-preserving, and energy-efficient decisions despite intermittent connectivity, adversarial attacks, and constrained local resources calls for novel models of secure communication, distributed consensus, and trust evaluation tailored to IoT's operating context.

2.2 Blockchain

The growing necessity of autonomous coordination, trustworthy data sharing, and trust establishment among infrastructure-less and resource-limited IoT devices has made blockchain technology one of the primary enablers of decentralized systems. Initially proposed as the underlying infrastructure of cryptocurrency [10], blockchain technology

has been developed as a universal instrument for safe, tamper-evident, and open record-keeping for distributed systems. Its core design, based on cryptographically linked blocks, distributed consensus mechanisms, and immutable ledgers, offers strong guarantees for data integrity and auditable system behavior without centralization. In the context of IoT environments, blockchain offers [11] several advantages that balances its operational constraints. Firstly, the distributed ledger avoids the requirement of a central controller, and thus devices can locally verify and record transactions even in partially connected or dynamic topologies. Secondly, its inherent tamper-resistance and auditability make it suitable for ensuring data origins and accountability between multiple possibly untrusted participants. This applies in particular to the use case of heterogeneous IoT devices, which may likely be owned, operated, trust-based, and failed separately. Third, by combining smart contracts or light-weight consensus protocols, blockchain can facilitate automated negotiation, and access control. Classic blockchain protocols like Proof of Work (PoW) [12] or even generic Proof of Stake (PoS) [13] are generally not feasible for IoT networks because they consume significant power, introduce high latency, and have computation overhead. In order to offset these limitations, literature has offered IoT-forged consensus protocols such as Delegated Proof of Stake (DPoS) [14] [15], Byzantine Fault Tolerant (BFT) variants, and reputation-based systems that ease the consensus overhead while security is maintained. These innovations aim to strike a balance between scalability, energy efficiency, and decentralization, a critical requirement for mission-critical or large-scale IoT applications. Nevertheless, while blockchain introduces consensus integrity and data immutability, it does not inherently provide trust among nodes. In other

words, although the network can necessarily come to an agreement [16] on "what happened" it cannot necessarily evaluate [17] "who can be trusted" under adversarial environments for malicious behavior, data poisoning, or selfishness. This limitation becomes critical in IoT environments, where node conduct varies dramatically based on resource limitations, changing roles, or conflicting objectives. Therefore, trust management emerges as a necessary complement to blockchain [18], augmenting its structural guarantees with behavior-aware and context-sensitive evaluation of node reliability.

2.3 Trust

There are various definitions and perspectives to define the concept of trust. These definitions vary depending on the application domain, the nature of the interacting entities, and the specific goals of the system, ranging from security assurance to behavioral expectation. For instance, [19] indicates trust as a subjective expectation or belief regarding the ability to provide secure services. On the other hand, [20] considers trust not only as a security measure but also as confidence, belief, expectation, reliability, and dependability. In [21], researchers define trust from another perspective, stating it reduces uncertainty among nodes while the network is connected to the internet. However, in [22] defines trust as confidence in the behavior during data exchange. In [23] trust defined as a reputation-based abstraction that reflects how much confidence one user has in another's reliability based on their past behavior and ratings within electronic communities. [24] explained trust as the subjective expectation that a node will receive positive outcomes from its interactions with another node in an Internet of Things (IoT)/Cyber-Physical Systems (CPS) environment. In the paper [25], trust is defined as the belief that an entity can perform a specific action under a particular context, formed based on direct interactions or recommendations from others. In [26], trust is defined as the authorization to perform security-critical actions, established through credentials and policies rather

than identity, enabling decentralized and scalable decision-making in distributed systems. Moreover, trust is considered in [16] as the authorization to perform specific actions in a decentralized system, established through credentials and policies, without relying on a central identity or authority. In [27], trust is defined as a subjective probability with which an agent believes that another agent will perform a particular action, even in the absence of monitoring, and in a context where that action affects the trusting agent's own behavior.

2.4 Literature Review

Table 2.1: *Analysis of various trust models*

	Architecture			Trust Model		Decay Functionality			Attack Immunity			Elimination of Maliciousness		Immutable Trust Logs	
	Centralized	Semi-centralized	Decentralized	Reputation	Computational	Time-dependent Decay	Event-dependent Decay	No Decay	Bad-Mouthing	On-Off	Collaboration	Malicious Nodes	Malicious Messages	Blockchain	No Blockchain
[1], 2004			✓	✓		✓			✓		✓		✓		✓
[2], 2023		✓		✓		✓			✓	✓	✓	✓	✓	✓	
[24], 2011			✓	✓		✓			✓	✓		✓			✓
[28], 2007			✓	✓				✓	✓		✓	✓			✓
[29], 2002	✓			✓				✓	✓	✓	✓		✓		✓
[30], 2008			✓	✓				✓	✓		✓	✓			✓
[31], 2015			✓		✓			✓	✓	✓	✓	✓	✓		✓
[32], 2012		✓		✓			✓		✓	✓		✓			✓
[33], 2015		✓			✓	✓			✓	✓	✓	✓			✓
[34], 2020		✓		✓			✓		✓	✓	✓	✓		✓	
[35], 2012			✓	✓				✓	✓	✓	✓	✓			✓
[36], 2003			✓	✓				✓	✓	✓	✓	✓			✓
[37], 2007			✓		✓	✓			✓			✓			✓
[38], 2019			✓	✓		✓			✓	✓	✓	✓		✓	
[39], 2020			✓	✓		✓			✓	✓	✓	✓	✓	✓	
[40], 2019			✓	✓		✓						✓	✓	✓	
[41], 2019		✓		✓		✓			✓	✓	✓	✓	✓	✓	

Reputation-based trust models have played a central role in managing trust in decentralized environments such as IoT, peer-to-peer, and ad-hoc networks. These models typically assess the trustworthiness of a node by its history of behavior and interaction with other nodes. Despite their widespread adoption, challenges such as feedback manipulation, dishonest ratings, and limited scalability persist. A prominent example is [1],

which computes trust by aggregating weighted feedback from different nodes in peer-to-peer systems. Their framework accommodates negative ratings for malicious content and includes mechanisms to handle dynamic node behavior management and secure trust data processing mechanisms. Another relevant work is [2], which introduces a semi-centralized blockchain-based trust architecture. The authors employ a weighted trust computation that accounts for both direct and indirect inputs, and they consider storage requirements in their design. Although the model performs well under varying Malicious Node Ratio (MNR), its semi-centralized nature introduces a vulnerability to single points of failure and offers a limited range of attack simulations. For reputation-based approaches, the ability to withstand several attacks is essential. The work in [36] uses global reputation scores and a probabilistic model to increase resilience to malicious behavior, and also suggests a reward system. Similarly, [29] enhances reputation modeling through beta distributions to probabilistically update trust values; however, it is designed for centralized settings. The PowerTrust model in [28] presents a reputation system with “power nodes”, a select group of high-reputation nodes identified via distributed ranking. This significantly lowers computation costs in large-scale networks, and the model outperforms [1] and [36] in scalability and node churn. While many studies such as [1], [36], and [28] utilize overlays like Chord or CAN, [30] introduces DHT with bloom filters and dynamic power node adjustments, enabling efficient reputation storage and reduced latency. Several works concentrate on trust in IoT-specific environments. For instance, [42] introduces a machine learning-based trust model using features such as goodness, usefulness, and perseverance features, using incremental SVM to respond to evolving malicious behavior. Their evaluation includes diverse attack strategies. The [24] framework integrates fuzzy logic to manage uncertainty and introduces energy-aware isolation for improved flexibility. Though comprehensive, it lacks evaluation against camouflage attacks. In contrast, [25] proposes the RKM model for the Social IoT, combining

reputation, experience, and knowledge as trust metrics. It is particularly designed for large-scale deployments, leveraging the propagative nature of reputation. Computational trust models aim to improve trust management through the application of algorithmic and mathematical formulations, which enable scalability, adaptability, and defense strength. These models commonly utilize optimization techniques, artificial intelligence, and formal logic to evaluate trust dynamically. For example, [31] applies genetic programming to evolve trust models capable of detecting various attacker types, including collaborative and recommendation-based ones. The system dynamically adapts to attack strategies such as pseudonym-based and hypocritical attacks. In [37], Schmidt et al. propose a flexible trust assessment framework that includes credibility adjustment, weighted aggregation, and defuzzification, primarily suited for business-oriented environments. While their method is highly customizable, it lacks detailed simulation against specific attack vectors and field testing. Certain trust models are designed with the explicit purpose of defending against specific attacks. For instance, [39] proposes a blockchain-based approach intended to defend vehicular location data in VANETs through an RSU-based anonymization approach. The model is evaluated for its resilience against Sybil, bad-mouthing, whitewashing, and on-off attacks. Similarly, [34] by Zhang et al. integrates the TrustRank algorithm in a Bayesian trust model that resolves on-off and collaborative attacks through local and global observations combined. The enhanced EigenTrust++ mechanism in [35] strengthens its predecessor with increased resistance to dishonest feedback and collusion. It includes evaluation against camouflage, Sybil, and coordinated attack testing. Another model, [33], employs user-based collaborative filtering in order to assess the trustworthiness of recommendations, achieving robustness against multiple adversarial methods. In the healthcare domain, the Retruster framework [32] targets sensor-based medical networks. It adheres to lightweight design principles and accommodates direct, indirect, and recommendation trust models. Its evaluation focuses on defending

against bad-mouthing and on-off attacks, though its application is domain-specific.

2.5 Trust and Energy Related Models

In [43], they proposed AICons which selects block validators through a decentralized federated learning process, where each node trains a local machine learning model using its historical data. These local models are then shared and aggregated into a global ML model, which ranks nodes based on their contributions. The node with the highest rank is selected as the winner to create a new block. It evaluates nodes' contributions by using Shapley value approach in [44], considering ML model accuracy and historical data. In contrast, our model dynamically selects delegates based on energy states, prioritizing energy efficiency over static contribution metrics, which makes it more suitable for resource-constrained environments. [45] proposes an energy-efficient model for securing IoT devices by integrating Code Division Multiple Access (CDMA) algorithm into blockchain. This approach multiplexes data from various IoT devices into a single blockchain transaction, reducing the number of transactions and lowering energy consumption by up to 90%. [46] proposes an optimized practical byzantine fault tolerance protocol, called S-PBFT, designed for energy-constrained IoT networks. The protocol uses verifiable random function for secure leader selection and applies a "closeness" centrality-based communication optimization to reduce message transmission overhead. S-PBFT improves energy efficiency by selecting nodes with sufficient energy levels at the initiation of the environment as consensus participants. Besides, [47] introduced an energy-aware data aggregation framework for distributed battery-less IoT networks, combining an application-aware task and energy manager and a vector synchronization-based data aggregator. LSTM model to predict the energy levels of IoT devices and dynamically schedules data collection tasks based on available energy. This approach reduces data loss by 98% and packet delays by 95%, it lacks consensus mechanism to adaptively

select nodes based on their live energy status and no blockchain application. Wadhwa et al. in [48] proposed an energy-efficient consensus mechanism for blockchain in IoT networks by offloading mining tasks to edge devices. A single miner is selected based on specifications such as RAM, CPU and bandwidth, reducing energy consumption by 21% and memory utilization by 24%. FedChain [49] integrates federated learning tasks into the mining process to reduce energy consumption but without dynamic network-wide energy management. PoFEL [50] uses federated learning models to select the leader node and implements cryptographic mechanisms; however, it focuses on optimizing leader selection rather than managing energy across the entire network. Proof of Team Sprint (PoTS) [51] adopts a team-based approach to replace individual competition in traditional PoW, reducing energy consumption by distributing computational tasks among groups, but it does not account for the active energy monitoring of all nodes. For IoT data exchange, the work in [2] suggests a semi-centralized trust management model that determines trust by combining direct and indirect trust values with decay functions and recommendation credibility to identify malicious nodes. While suitable for dynamic behavior monitoring and scalability, their model primarily focused on service quality and reputation, without considering energy availability or decentralized voting mechanisms. [52] developed a blockchain-inspired trust and incentive mechanism for task offloading security in Mobile Edge Computing (MEC), by combining bidirectional trust evaluation with a Stackelberg game-based model. While, scope remains limited to task-oriented trust and incentives alone, without addressing general consensus dilemmas like delegate selection or malicious behaviors network-wide. The GLSBIoT framework proposed in [53], a hybrid trust-based consensus using PoW and PoS, enhanced with Grey Wolf Optimization (GWO) to enable side-chaining for improved scalability and energy efficiency in IoT environment. Other works, such as [54,55], employ clustering or hierarchical trust structures to detect reputation anomalies. These approaches offer valuable insight for adaptive trust

profiling but often operate in environment without energy or voting constraints. Earlier reputation-based trust models in [54] and [55] use clustering or hierarchical structures for anomaly detection in mobile ad hoc and sensor networks. While these models are effective in profiling unreliable nodes, they are designed for static or semi-static environment and are not integrated into modern blockchain consensus frameworks. In [56], an energy-adaptive version of the Raft consensus algorithm is introduced. It allows IoT nodes to adjust their consensus roles based on real-time energy phases. While the model enhances leader availability, it focuses on intra-cluster coordination and lacks stakeholder-driven voting or behavior-based filtering. The EA-PoA (Energy-Aware Proof-of-Authority) protocol proposed in [57] selects cluster leaders based on residual battery capacity and utilizes a multi-layer blockchain to improve network longevity. Despite these benefits, the model does not support trust scoring or adversarial behavior detection, and is primarily designed for static, WSN-style topologies. In [58], a dual-layer blockchain framework is introduced for UAV data sharing under 6G networks. The model includes dynamic consensus that reacts to resource conditions. However, it lacks delegate-based consensus and does not integrate behavioral trust metrics with energy profiling.

2.6 Trust Models: Centralized, Semi-Centralized, Decentralized

2.6.1 Centralized Trust Models

The Beta Reputation System [29] introduces a reputation model based on beta probability density functions, providing a statistically grounded approach to combining positive and negative feedback. It defines flexible mechanisms for discounting feedback, and combining ratings, with a focus on formal mathematical soundness. While it emphasizes probabilistic trust computation, the model is primarily centralized and lacks integration with real-time, distributed decision-making or behavioral adaptation mechanisms.

2.6.2 *Semi-Centralized Trust Models*

Liu et al. propose a semi-centralized trust management model in [2] that leverages blockchain to facilitate data exchange in IoT systems. Their architecture introduces a two-layer design, cloud and device layers, where cloud servers coordinate trust evaluations using a rotation-based blockchain consensus protocol. The model calculates trust by aggregating direct and indirect trust, a decay function based on Newton’s cooling law, and recommendation credibility mechanisms. While it effectively mitigates malicious behavior across multiple domains, the reliance on a semi-centralized architecture introduces potential single points of failure and reduces adaptability in decentralized environments. AATMS [34] proposes a trust management scheme for VANETs, using Bayesian inference for local trust and a TrustRank-inspired algorithm for global trust. It incorporates adaptive forgetting and decay factors to counter on-off and newcomer attacks, and considers social factors for selecting seed vehicles. While resilient against various attacks, AATMS depends on trusted authorities, limiting its scalability and flexibility in fully decentralized environments. ReTrust [32] is a lightweight, attack-resistant trust model for Medical Sensor Networks, using a two-tier architecture where low-power sensor nodes report to resource-rich master nodes for trust computations. It employs a sliding time window and modified beta distribution for trust calculation, with defenses against on-off attacks and bad-mouthing via weight adjustments. While effective in healthcare settings, its reliance on master nodes and centralized decision-making limits its suitability for decentralized environments. ART [33] proposes an attack-resistant trust management scheme for VANETs, evaluating data trust and node trust separately but interdependently. Node trust combines functional and recommendation trust, using direct observations and collaborative filtering, while data trust leverages multi-vehicle sensing and Dempster-Shafer Theory for evidence fusion. Despite strong precision, recall, and low communication overhead, ART relies on semi-trusted RSUs and lacks system-wide dynamic decay for

adaptive node elimination.

2.6.3 *Decentralized Trust Models*

PeerTrust [1] is a reputation based trust model for peer-to-peer networks that computes trust using weighted feedback, transaction volume, and feedback credibility. It operates in a fully decentralized manner and introduces mechanisms like personalized similarity and trust vector models. While effective against basic manipulation, it lacks dynamic trust decay or statistical detection mechanisms for evolving attack strategies. GossipTrust [30] introduces a decentralized reputation aggregation system for unstructured P2P networks using randomized gossip protocols. It computes global trust scores through iterative message exchanges and matrix-vector multiplications, achieving convergence in $\mathcal{O}(\log^2 n)$ steps. The model uses Bloom filters to store and rank peer reputations efficiently and employs power nodes to mitigate collusion. However, it lacks adaptive decay or statistical anomaly detection mechanisms, which limits its responsiveness to evolving malicious behaviors. PowerTrust [28] presents a decentralized system designed for large-scale P2P computing. It leverages a Trust Overlay Network and power-law distribution of peer feedback to dynamically select a small number of power nodes. Key mechanisms include look-ahead random walks, and periodic re-election of power nodes. While the model demonstrates low overhead, and strong resistance to collusive and non-collusive malicious peers, it lacks statistical trust decay. EigenTrust [36] proposes a decentralized reputation system for P2P networks that computes global trust values using a distributed eigenvector-based aggregation of local feedback. Each peer assigns scores based on successful transactions, and the trust matrix is iteratively computed using power iteration methods. While EigenTrust achieves fast convergence, it does not include adaptive trust decay or statistical node evaluation, limiting its responsiveness to evolving behavioral anomalies. EigenTrust++ [35] enhances the original EigenTrust [36] for P2P systems,

improving resilience against dishonest feedback, and collusion. It introduces, local trust computation, feedback credibility weighting based on rating similarity, and threshold-based probabilistic trust propagation to similar peers. It outperforms EigenTrust in resisting independent, collaborative, camouflage, and spy-based attacks but lacks dynamic decay or context-aware trust attenuation. TRM-IoT [24] introduces a decentralized trust and reputation model tailored for IoT. The model computes both direct and indirect trust values through observed and recommended interactions. It combines packet forwarding ratio, energy consumption, and delivery ratio. Local and global trust values are continuously updated based on behavioral consistency over time, with heavier weight given to recent interactions. TRM-IoT is optimized for resource-constrained IoT nodes, but lacks a decay mechanism and node removal strategy found in more recent dynamic trust models. Schmidt et al. [37] propose a fuzzy logic, based trust evaluation model for multi-agent systems, designed to mimic human reasoning in uncertain and decentralized e-commerce environments. The model integrates weighted trustworthiness computation, credibility adjustment, and post-interaction review. While the model supports agent-specific trust policies, it lacks decay mechanisms that adapt to global system behavior or interaction frequency trends. GenTrust [31] introduces a genetic programming-based trust model for decentralized P2P systems, evolving trust formulas from direct interactions and recommendations. Peers independently compute trust using satisfaction, file weights, and neighbor feedback, encoded as genetic trees. It resists individual, collaborative, and pseudospoofing attacks, minimizing malicious interactions. Among the reviewed works, PeerTrust [1], EigenTrust++ [35], and the Semi-Centralized Trust Model [2] are the closest to our approach in terms of architecture and purpose. PeerTrust offers a decentralized approach using experience over time and source reputation but lacks any temporal decay mechanism, which may cause outdated trust data to persist indefinitely. EigenTrust++

enhances its predecessor by merging credibility-weighted propagation with attack resistance but does not incorporate time-based trust decay. The Semi-Centralized Trust Model employs an exponential decay inspired by Newton's law, but the decay rate is fixed and does not adapt based on changing network behavior. None of them incorporate a statistically adaptive decay that evolves in real time based on activity patterns and node behavior shifts.

2.7 Summary of Gaps in the Literature

Despite the extensive body of research in trust management for decentralized systems, several critical limitations persist. The majority of reputation models rely on static thresholds and fixed decay mechanisms, which makes them not sufficiently adaptive to rapidly changing node behaviors or context-dependent trust fluctuation. Additionally, computational models, being more robust, are likely to be overly complex, not scalable, or lacking evaluation under different types of attacks. A considerable proportion of such models are either limited to simulation environments or fail to address real-world constraints like energy efficiency, compatibility with consensus protocols, or heterogeneity among participating nodes.

Moreover, although some studies utilize blockchain to improve trust integrity, most overlook the realistic incorporation of trust mechanisms into communication infrastructure like UAVs and VANETs, especially in disaster scenarios where the infrastructure is unreliable. There are extremely limited models holistically combine behavior monitoring, adaptive trust scoring, and lightweight consensus mechanisms in a unified framework that is resistant to advanced threats such as camouflage, collusion, and on-off attacks. These limitations highlight the need for a multi-layered, resilient, and context-aware trust management system, one that this thesis aims to fulfill by incorporating blockchain-based models with energy, behavior, and role-aware assessment techniques in mission-critical

areas.



3. TRUSTVISTA: NODE, GROUP AND COMMUNITY BASED PERSPECTIVE, BLOCKCHAIN UTILIZED TRUST MECHANISM FOR IOT

The exponential growth of Internet of Things (IoT) devices has introduced new dimensions of automation, data collection, and intelligent decision-making in different applications. However, the resource-constrained and decentralized nature of IoT systems, such as Unmanned Aerial Vehicles (UAVs), Vehicular Ad-Hoc Networks (VANETs), and Mobile Ad-Hoc Networks (MANETs), presents severe security threats. In dynamic systems like disaster relief operations or smart city infrastructure, insecure devices have the potential to significantly compromise major communication, data integrity, and real-time coordination [59]. Trust management protocols dynamically analyze and quantify the reliability of nodes on the basis of experienced behavior, history, and network feedback. They enable systems to recognize trusted participants from malicious ones, thereby preserving system health. As IoT networks expand and diversify, maintaining adaptive trust mechanisms has become a critical concern for ensuring resilience against improving malicious strategies. Blockchain technology has been becoming more integrated with trust management systems because of its fundamental properties of decentralization, and immutability. Its distributed ledger architecture exclude the need for a centralized authority, allowing secure and immutable storage of transactional data. Blockchain alone, however, does not judge node behavior or detect behavior-based attacks such as collaboration attacks or camouflage behavior. This gap limits its effectiveness in real-time IoT security contexts where behavior analysis is essential. Different blockchain trust management frameworks have been proposed [1, 2]. Most of the existing models typically apply static thresholding, fixed-valued trust decay rates, or semi-centralized assessment structures. These compromise their response to dynamic attacks such as attacks, camouflage, and

on-off attacks, involving real-time and context-sensitive adjustment. Static decay functions, where trust scores reduce constantly with time regardless of network conditions, and are not able to capture evolving node behaviors in dynamic networks. To address these limitations, we propose TrustVista, a multi-perspective trust management system that evaluates node behavior from the node, group, and community perspectives. By aggregating trust insights across multiple perspectives, our model enables more scalable, and adaptive trust scores without calling for centralization. We propose a Multi-Factor Dynamic Decay function. Unlike the semi-centralized model in [2], which uses fixed-rate decay and centralized evaluation, our approach enables more responsive and distributed trust adjustment through context-aware dynamic decay. Decay functions in traditional models only decrease trust scores over time to account for outdated behavior, at fixed exponential or linear rates. However, our dynamic decay mechanism adjusts trust score decrease rate as a function of interaction frequency, system-wide activity levels, and time elapsed since last transaction. This provides for more context-dependent trust updates so that trust scores accurately reflect actual behavior patterns. We employ a Z-score, based statistical trust assessment approach to dynamic malicious node identification and removal. This statistical methodology allows trustworthiness assessments to capture deviations from group and network norms, allowing for the more rapid and accurate identification of aberrant nodes. The main contributions of this section are as follows, we introduce a trust evaluation framework, TrustVista, which captures behavioral context across multiple perspectives (node, group, community) that facilitates an extensive and large-scale trust model in decentralized networks. This grouping abstraction is non-hierarchical but a statistical abstraction for the purpose of increasing trust scoring granularity without centralization. Also, we introduce a Multi-Factor Dynamic Decay function that enables context-aware trust score updating for improved responsiveness and stability compared to traditional decay models. We apply a Z-score-based statistical trust evaluation technique

to dynamic malicious node detection and removal. The mechanism enables the system to react in real time to behavioral anomalies, improving its resilience against adaptive malicious activities. Additionally, we implement and test our approach through large-scale OMNeT++ [60] simulations over variety of attack scenarios, demonstrating improved attack detection rates, trust stability, and system robustness compared to baseline models.

3.1 Overview of TrustVista Architecture

In decentralized networks, the term service refers to distinct functionalities executed by participating nodes, such as data sharing, computation, or transaction validation. They are constantly evaluated by receiving nodes based on a set of quality metrics, which are then used to compute the TS of the providing node. Effective trust management ensures the general integrity and soundness of a network. In the absence of a centralized authority, however, ensuring secure and consistent interactions becomes a complex challenge, especially in large-scale, dynamic environments. The traditional trust management models are generally not effective in meeting the requirements of scalability, flexibility, and resilience needed by such systems.

In response to these limitations, a group-based hierarchical trust model is proposed, TrustVista, that evaluates and maintains trust from three related perspectives: node-level, group-level, and community-level. At the node level, the trust rating for every node is calculated depending on its own performance and peer feedback. At the group level, nodes are grouped into manageable clusters and collective behavior of every cluster contributes to a Group Trust Score (GTS). Also finally, at the community level, all group scores are integrated into a Community Trust Score (CTS) that reflects the overall trust of the network.

The proposed architecture is built around the following three main objectives:

- **Scalability:** To enable efficient trust management in huge networks by segmenting

the network into smaller, manageable trust domains while ensuring global trust consistency.

- **Security:** To detect and isolate malicious nodes through trust evaluations at different hierarchical levels, improving resilience against localized or coordinated attacks.
- **Dynamism:** In order to allow a ripple effect in trust propagation, whereby an action of a node not only influences its own trust value but also that of its peers and group, hence enabling faster response to suspicious activity.

By introducing this multi-perspective approach, the system not only ensures highly efficient trust management model, but also gives a flexible framework that adapts dynamically to the changes in the network such as attacks or behavior changes. The proposed trust management model is referred to as "Dynamic", hence every node is responsible from every other. That ripple effect allows us to detect malicious activity and nodes quicker, by spreading the malicious activity throughout the system and the malicious nodes are marked more precisely.

3.2 System Model: Node, Group, and Community-Based Trust Modeling

The TrustVista framework models trust as a hierarchical and interdependent structure, where evaluations are conducted across three primary levels: nodes, groups, and the overall community. At the node level, trust is calculated individually by evaluating each node's service performance and behavior during interactions. This includes metrics such as service availability, response time, and data integrity.

3.2.1 Node Perspective

Nodes form the foundational layer of the TrustVista framework and autonomous entities with the ability to initiate opening of interactions, measuring trust, and carrying out validation of transactions and voting operations. Each node not only serves its own computational or data-oriented tasks but also contributes actively to the distributed trust management of the entire system. When the node is served, such as data sharing, message forwarding, or computation, the node evaluates the service provider based on multiple predefined metrics and assigns a trust score (TS). This score is then broadcast to the network and recorded on the blockchain ledger, to generate transparency, immutability, and verifiability. The calculation of trust at the node perspective is grounded in three core evaluation metrics:

- **Response Speed:** This metric measures the response speed of the provider node.
- **Activity:** It reflects how frequently the node do transactions, such as sending and receiving data, and its overall contribution to the network.
- **Service Integrity:** This metric evaluates whether the serviced data has been fully and successfully transmitted.

Although each node operates independently in terms of decision-making and connectivity, they establish direct Peer-to-Peer (P2P) links with neighboring nodes. These links form the foundation of localized trust evaluation and allow the emergence of overlapping groups. The initiation of a transaction begins when a node sends a service request to one of its directly connected peers. The recipient node either accepts or declines this request. In cases where the request is declined, the requesting node may reduce the provider's TS in future interactions due to poor responsiveness or unwillingness to cooperate. Once the service is successfully provided, the requester evaluates the interaction and assigns a TS

to the service provider based on the defined quality metrics. This TS is then submitted to the smart contract for further steps. As shown in Figure 3.1, the trust evaluation process

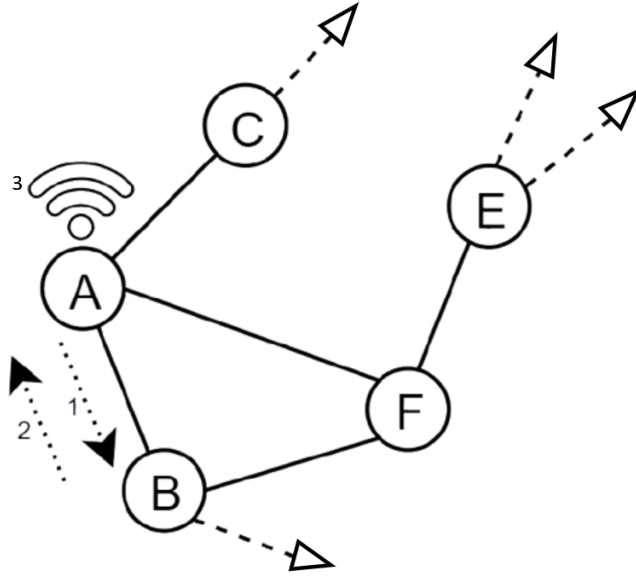


Figure 3.1: *Node-Centric Interaction in TrustVista*

starts when a node e.g., Node A, initiates a service request to a directly connected peer, e.g., Node B. Node A is directly connected to nodes B, C and F. In this figure Node A interacts with Node B. Step 1 shows the request phase as Node A ask for a service from Node B. Afterwards, in step 2 Node B provides the service. Then, in step 3, Node A assigns a TS to Node B based on the defined quality metrics. Following, this score is submitted to the smart contract, initiating the trust update and verification process.

3.2.2 Group Perspective

In decentralized environments, nodes naturally form localized trust contexts based on their direct connections. Rather than pre-defined or statically assigned structures, the proposed model allows each node to dynamically construct its own trust neighborhood, a conceptual group composed of all directly connected peers. These emergent group

structures are not hierarchical layers but reflections of P2P interactions and trust evolution within local scopes of the network. In the proposed TrustVista architecture, the group-level trust is quantified by the Group Trust Score (*GTS*), which is recorded on the blockchain. The *GTS* not only reflects the collective reliability of the group but also plays a critical role in identifying potentially malicious nodes. *GTS* is the metric to decide a node's future existence and improves the ability of malicious node detection. Unlike centralized systems, no node holds special authority or control over a group. Instead, a node influences both its own *GTS* and those of its peers by participating in multiple peer-defined groupings simultaneously. This overlapping structure increases fault tolerance by ensuring that the removal or failure of a single node does not monopolize the overall trust evaluation across the network. When a new node establishes direct connection with a peer, it automatically forms a new group and the new node designated as the leader. Subsequently, a request for group creation is sent to the blockchain, where the group is registered under the leader's node index (e.g., Group A). On the other hand, when an existing node connects to another node, both nodes add each other to their respective groups. System evaluates each node according its own group's *GTS*. Although, there is no privilege of being a leader but every node should check frequently the connectivity with each own group member nodes. Figure 3.2 illustrates the grouping model and overlapping structure of the proposed system model, highlighting the role of leader concept and nodes interactions with directly connected nodes.

In Figure 3.2, each node has at least one connection so, they have at least one membership and a formed group. In the sample structure presented in Figure 3.2, Node A's group showed in the figure by 1. Node A act as a group leader and directly affects own *GTS* threshold. But also acts as a member of Node B's group, Node F's group and Node C's group so, its indirect influence stems from Node A's presence during the evaluation of *GTS* in these groups, as its trust score contributes to the overall trust threshold. Group

of Node B represented as 2 and the group contains members Node A-F-G, because its direct connections. Connected nodes are indirectly affecting Node B's group threshold,

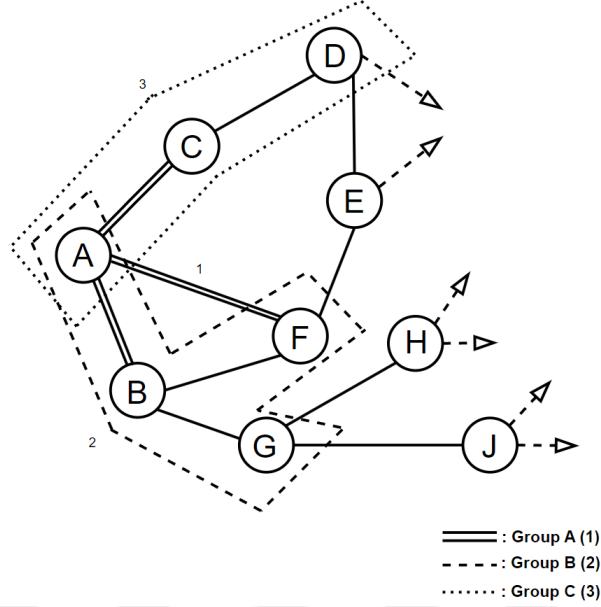


Figure 3.2: *Group perspective in TrustVista*

however Node B directly influences own *GTS*. Meanwhile, group 3 is formed under Node C, the group includes Node D, Node C, and Node A. Besides Node C's direct influence, group C is indirectly affected by Node A and Node D, both participate in Node C's *GTS* calculations.

3.2.3 Community Perspective

The community perspective captures the aggregated trust state of the entire network. Every node is a member of the community, and their behaviors individually contribute to overall system-wide trust dynamics. This perspective does not imply a hierarchical layer but represents a network-wide evaluation of overall reliability. To quantify this, a metric called the Community Trust Score (*CTS*) is introduced. Computed on the blockchain, *CTS* reflects the average trustworthiness of all participating nodes, based on

their *NTS*. The *CTS* continuously adapts to real-time changes in node behavior, offering a comprehensive indicator of the system's health and stability. While *NTS* and *GTS* manage individual and group behavior individually, *CTS* offers a general view that ensures coherence across the decentralized structure. This trust model includes multiple perspectives from broader to smaller by proposing a abstract grouping strategy.

3.3 Trust Calculation and Evaluation

In decentralized networks, trust relies profoundly on the interactions and evaluations between nodes. Each node evaluates and stores information about its peers as its the nature of blockchain and decentralized architecture, forming an individual perspective on the network's overall reliability. In 1968, Virginia Held described trust as "In short, it seems to me that trust is most required exactly when we least know whether a person will or will not do an action." in [61]. Building on this perspective, the unpredictability of node behavior necessitates the formulation of trust evaluations as a mathematical expression within a measurable framework. The thesis implemented a trust score *TS* mechanism in the framework, where nodes are assigned a score after each interaction based on their performance. This approach ensures that trust is no longer abstract but quantified, allowing the network to continuously assess the reliability of its participants. Metrics such as response speed, activity and service integrity are essential for the evaluation of the service provider node.

3.3.1 Trust Score Computation Framework

In this subsection, trust calculations are performed at three levels: node, group, and community. Each level builds on *TS* as its foundation, with *NTS* reflecting individual interactions, *GTS* aggregating individual node scores, and *CTS* is used for broader evaluation of network. Together, these levels provide a scalable and comprehensive trust

management approach. This paper proposes trust score calculations based on metrics, which builds on the basis of TS s. Suppose Node A requests a data service from Node B , and after receiving the service, Node A evaluates Node B based on several key performance metrics. These metrics are normalized because they collectively form the TS that Node A assigns to Node B for this interaction. Over time, the trust scores from multiple interactions accumulate, forming the hierarchical trust evaluations.

3.3.1.1 Response Speed

Response speed reflects how fast Node B responds to a data request $req(A \rightarrow B, data)$ initiated by Node A . To calculate the normalized response speed, Normalized Response Time is defined which is defined as $NorRT$. Accordingly, also Normalized Response Speed is defined which is denoted as $NorRS$. The normalized response speed formulations are shown below in Equation 3.1 and Equation 3.2.

$$NorRT_{B \rightarrow A} = \frac{T_i}{RT_{\max}} \quad (3.1)$$

In Equation 3.1 T_i represents the response time for the i -th request and $RT_{\max}$ indicates maximum response time in the network.

$$NorRS_{B \rightarrow A} = \frac{1}{NorRT_{B \rightarrow A}} \quad (3.2)$$

In Equation 3.2 the speed formulation is applied to find the normalized response speed.

3.3.1.2 Activity

Activity measures the extent of Node B 's activeness across the network, focusing on its interaction frequency and contribution to data exchanges, which is key for network strength and the system remains up to date. Specifically, this metric evaluates how frequently Node B performs transactions with Node A over a given time period. By incorporating this metric, the influence to the network of Node B as perceived by Node A .

This provides a quantitative assessment of how active a node is in the network.

$$RAc_{B \rightarrow A} = \frac{|Tr_{B \rightarrow A}|}{t_{\text{current}} - t_{\text{last}}(B \rightarrow A)} \quad (3.3)$$

In Equation 3.3, raw activity score $RAc_{B \rightarrow A}$ is calculated using Equation 3.3, where $Tr_{B \rightarrow A}$ denotes the total number of transactions Node B has performed for Node A . The value is then divided by the time elapsed since last transaction.

$$NorAc_{B \rightarrow A} = \frac{RAc_{B \rightarrow A}}{RAc_{\text{max}}} \quad (3.4)$$

To ensure uniformity in TS calculation, the normalized activity score $NorAc_{B \rightarrow A}$, is obtained by dividing $RAc_{B \rightarrow A}$ by the maximum activity score across the network, as expressed in Equation 3.4.

3.3.1.3 Service Integrity

Service integrity is a metric used to evaluate the integrity of data packets transmitted in a single transaction from Node B to Node A . It measures the success rate of data packets transmitted, more specifically service integrity is defined as the ratio of the number of successfully delivered packets to the total number of packets sent from from Node B to Node A . The formula is expressed in Equation 3.5 as follows:

$$SerI_{B \rightarrow A} = \frac{|Ser_{B \rightarrow A}|}{|Ser_{\text{total}}|} \quad (3.5)$$

Equation 3.5 provides the success rate, extend insight into how reliable Node B is in sending accurate data to Node A .

$$NorSerI_{B \rightarrow A} = \frac{SerI_{B \rightarrow A}}{SerI_{\text{max}}} \quad (3.6)$$

In Equation 3.6, normalization scales all service integrity values between 0 and 1 for comparison among nodes. Specifically, $NorSerI_{B \rightarrow A}$ represents the normalized service integrity, where $SerI_{B \rightarrow A}$ is the service integrity for Node B 's provided data packets to Node A , and $SerI_{\text{max}}$ denotes the maximum service integrity value in the network.

3.3.1.4 Trust Score Calculation

Trust Score (TS) is used to evaluate the reliability of Node B's transaction to Node A, based on response time, activity level and service integrity shown in Equation 3.7.

$$TS_{A \rightarrow B} = f(NorRS_{B \rightarrow A}, NorAc_{B \rightarrow A}, NorSerI_{B \rightarrow A}) \quad (3.7)$$

In the calculation of TS , each of these metrics can be assigned a weight based on its importance to the system's goals. The trust score calculation formulation shown below in the Equation 3.8 equation:

$$TS_{A \rightarrow B} = w_1 \times NorRS_{B \rightarrow A} + w_2 \times NorAc_{B \rightarrow A} + w_3 \times NorSerI_{B \rightarrow A} \quad (3.8)$$

The weights w_1 , w_1 , and w_1 are set to 0.33 each in an equilibrium system where all metrics are considered equally important. However, these weights can be adjusted based on the system's real-time requirements and priorities. This adaptability allows the trust evaluation to evolve in response to changing conditions, ensuring that the system remains responsive and effective in varying contexts.

3.3.2 Node Trust Score Calculations

In the proposed trust model, each node is assigned a Node Trust Score (NTS), and it is constantly updated following every transaction. On receiving service from a node (Node B), the recipient of the service (Node A) evaluates the quality of this transaction and provides a TS. This score reflects the most recent behavioral assessment of the provider and is used to update the node's overall trustworthiness.

To effectively model evolving behavior and maintain sensitivity to recent interactions, the Exponentially Weighted Moving Average (EWMA) method is employed for updating the NTS. As given in Equation (3.9), the new trust score NTS_B^t at time t is computed by weighting the acquired trust score $TS_{A \rightarrow B}^t$ and the previous score NTS_B^{t-1} ,

and a dynamic decay factor DeP_B^t :

$$NTS_B^t = (w_B^t \times TS_{A \rightarrow B}^t) + ((1 - w_B^t) \times NTS_B^{t-1} \times DeP_B^t) \quad (3.9)$$

The weight factor w_B^t , defined in Equation (3.10), determines the relative influence of the new interaction versus the historical performance. It is calculated based on the average normalized values of three key behavioral metrics, Response Speed ($NorRS_B$), Activity ($NorAc_B$), and Service Integrity ($NorSerI_B$):

$$w_B^t = \frac{f(\overline{NorRS}_B, \overline{NorAc}_B, \overline{NorSerI}_B)}{NorRS_{\max} + NorAc_{\max} + NorSerI_{\max}} \quad (3.10)$$

Maximum normalization constants are set to 1. If $w_B^t \in [0.0, 0.5]$, the model favors historical trust; otherwise, it emphasizes recent interaction.

3.3.3 Multi-Factor Dynamic Decay Function

To further enhance the adaptability of the trust model, "Multi-Factor Dynamic Decay" function is implemented. As shown in Equation (3.11), the decay factor DeP_B^t reduces the influence of outdated trust scores based on time elapsed and node activity:

$$DeP_B^t = e^{-\left(\lambda \times \frac{\Delta t_{B \rightarrow A, last}}{\frac{1}{|N|} \sum_{i \in N} t_i^{last}}\right)} \times e^{-(\lambda \times \ln(n))} \quad (3.11)$$

This formulation integrates two decay dimensions:

- **Temporal decay:** Reflects how long it has been since Node B 's last interaction relative to the network average. While $\frac{1}{|N|} \sum_{i \in N} t_i^{last}$ is used to calculate the average of the last transaction elapsed times across all nodes.
- **Interaction volume decay:** Adjusts trust based on the number of interactions n .

The base e in this function represents the natural exponential decay behavior. Its use ensures a smooth and continuous reduction of trust contribution from outdated data, modeling how real-world trust tends to reduce over time or inactivity.

The decay sensitivity is governed by the parameter λ , computed dynamically as computed in Equation (3.12):

$$\lambda = \frac{1}{\ln\left(\frac{1}{|N|} \sum_{i \in \mathcal{N}} \frac{\sum_{j=1}^{m_i} t_{i,j}}{m_i}\right) + \ln(T)} \quad (3.12)$$

The coefficient λ is determined based on the average transaction processing time across all nodes and the total number of system transactions. Here, T represents the total number of transactions in the system, and $t_{i,j}$ represents the timestamp of the j^{th} transaction of node i . The greater the T results in smaller λ values, preserving long-term trust. Conversely, lower activity increases λ , causing the model to favor recent observations, enabling faster detection of malicious behavior.

Traditional trust decay mechanisms often model time-based attenuation using simplified assumptions. For instance, prior studies such as [2] adopt Newton's law of cooling, where decay is modeled as a function of time elapsed since a rating was given and the number of prior positive responses. These such models offer intuitive behavior for time-window-constrained evaluations, they often fail to incorporate the broader dynamics of system-wide interactions and real-time adaptability.

3.3.4 Group Trust Score Calculation

In the calculation of GTS , every single node of the group are responsible on its own GTS , so every node is considered. The average of their NTS 's is then used to determine the overall trust at the group level.

$$GTS_A^t = \frac{\sum_{i \in A} NTS_i^t}{|M_A|} \quad (3.13)$$

Suppose, Group A 's GTS_A is calculated in Equation 3.13 according Node A , Node B , Node C and Node F . It is computed by summing up the NTS_i of all nodes (i) in Group A and then dividing by the total number of nodes in the group, M_A , which is denoted as GTS_A .

3.3.5 Community Trust Score Calculation

CTS evaluates the entire network status, which is calculated by averaging the NTS of all nodes in the community. The formula for CTS is Equation 3.14 as follows:

$$CTS_C^t = \frac{\sum_{i \in C} NTS_i^t}{|M_C|} \quad (3.14)$$

CTS_C^t represents the CTS at time t , which assess the overall trust level of the community at time t , while M_C represents the total number of nodes in the community C . Meanwhile, NTS_i^t denotes the NTS of Node i at time t .

3.4 Trust Assessment and Decision Making

Maintaining the integrity of decentralized systems depends heavily on the timely identification and mitigation of underperforming or malicious nodes. Within the proposed framework, a novel approach is proposed which involves statistically-driven trust evaluation and decision-making mechanism that operates based on dynamic assessment of node behavior. The trust assessment procedure identifies under performing or malicious nodes, keeping reliable ones active in the network. The mechanism enables the network to isolate or remove nodes that significantly deviate from expected performance levels, thereby guaranteeing system reliability and resilience. We propose a Z-score-based evaluation mechanism paired with a dynamically adaptive removal threshold called the Dynamic Removal Threshold (DyRT). $DyRT$ is used to quantify how far a node's performance deviates from the average behaviors of its own group and the entire community. It is defined as follows:

$$DyRT_A^t = \frac{|\mu_{G(A)}^t - \mu_{C(A)}^t|}{\sigma_{G(A)}^t + \sigma_{C(A)}^t} \quad (3.15)$$

In Equation (3.15), $\mu_{G(A)}^t$ and $\mu_{C(A)}^t$ denote the average NTS values of Node A 's group and the entire community at time t , respectively, while $\sigma_{G(A)}^t$ and $\sigma_{C(A)}^t$ represent their

corresponding standard deviations. The numerator captures the deviation between localized and overall trust behavior, while the denominator ensures normalization with respect to behavioral variability. Fixed thresholds fail to capture the evolving behavior patterns and statistical distributions of trust in dynamic and decentralized networks. By contrast, a dynamic threshold adapts to real-time changes in network conditions so that the system to account for varying degrees of behavioral anomalies without being too sensitive or permissive. This is particularly important in environments with non-stationary behavior, where trust values change over time as a result of network fluctuations, transaction variability, or temporary performance degradation. By tuning the threshold with current statistical analysis, the model reduces false positives and false negatives, thereby increasing robustness, fairness, and adaptability of the trust assessment process.

$$\mathcal{Z}_{G(A)} = \frac{NTS_A - \mu_{G(A)}}{\sigma_{G(A)}} \quad (3.16)$$

The Z-Score from the group perspective, denoted as $\mathcal{Z}_G(A)$ in Equation 3.16, quantifies the deviation of Node A 's trust score from the mean trust score $\mu_{G(A)}$ of its corresponding group. This metric provides a statistical measure of how significantly Node A 's behavior diverges from the expected behavior within its local trust environment. The aim is to detect whether Node A is performing better or worse compared to the group. Another measure for $\mathcal{Z}_{G(A)}$ is $\sigma_{G(A)}$, which represents how much the performance of nodes within the group deviates from the group's average, known as the Standard Deviation.

$$\mathcal{Z}_{C(A)} = \frac{NTS_A - \mu_{C(A)}}{\sigma_{C(A)}} \quad (3.17)$$

The community perspective in Equation 3.17 uses metrics like $\mu_{C(A)}$ and $\sigma_{C(A)}$, representing the average and standard deviation for the community that Node A is part of. The general form of standard deviation is expressed as:

$$\sigma_X = \sqrt{\frac{1}{|M_X|} \sum_{i \in \mathcal{X}} (NTS_i - \mu_X)^2} \quad (3.18)$$

where $X \in \{G, C\}$ and M_X denotes the number of nodes in the selected domain. Based on these scores and the computed $DyRT$, each node's status is classified into one of three categories based on its statistical deviation from group and community norms:

$$NS_A^t = \begin{cases} \text{Removal} & \begin{cases} Z_{\text{group}(A)} < DyRT_A^t \\ \text{and} \\ Z_{\text{community}(A)} < DyRT_A^t \end{cases} \\ \text{Isolation} & \begin{cases} Z_{\text{group}(A)} < DyRT_A^t \\ \text{or} \\ Z_{\text{community}(A)} < DyRT_A^t \end{cases} \\ \text{Active} & \text{otherwise} \end{cases} \quad (3.19)$$

- **Removal:** If both the group-level and community-level Z-Scores fall below the computed $DyRT_A^t$ threshold, the node is considered to be exhibiting consistently malicious behavior. This double violation indicates a systemic deviation rather than a temporary fluctuation. Therefore, the node is permanently removed from the network.
- **Isolation:** If either Z_{group} or $Z_{\text{community}}$ is below the dynamic threshold, but not both, the node is considered partially inconsistent with expected trust behavior. In this intermediate state, the node is marked as isolated and they are allowed to engage in transactions and maintain connectivity within the network, but are excluded from voting in critical processes.
- **Active:** Nodes whose Z-Scores remain above the $DyRT$ threshold in both perspectives are considered trustworthy. These nodes participate in all network activities.

Furthermore, once a node is removed, all of its prior trust scores are disregarded in future

evaluations, thereby preventing historical data from contaminating ongoing trust computations. This ensures that the trust landscape reflects only the active and relevant node behavior, promoting long-term integrity in the environment.

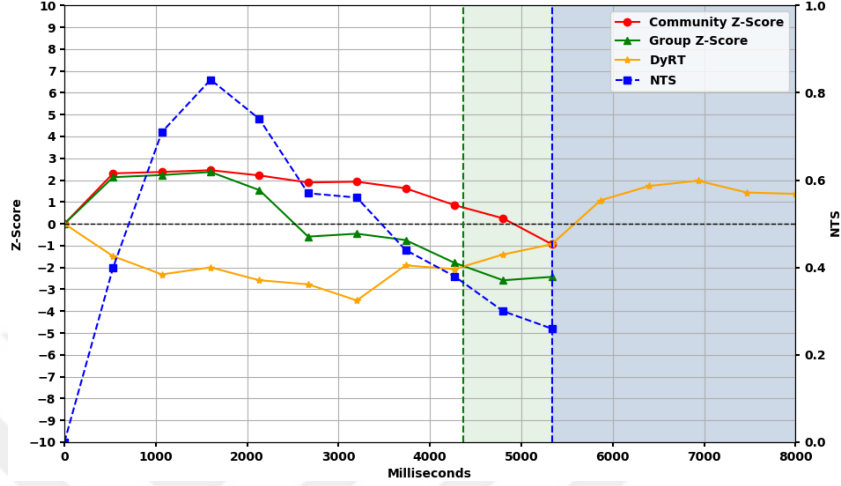


Figure 3.3: *Life cycle of a malicious node from initialization to termination in TrustVista.*

Figure 3.3 illustrates a time-series plot of the behavior of a malicious node during its lifespan. In the early stages, the node maintains high trust score, resulting in increasing Z-Scores and NTS. As the node’s behavior begins to deviate from group’s activity, its Group Z-Score falls below the $DyRT$, and it is isolated. This is reflected at 4360 ms when the node is no longer allowed to participate in voting processes but can still execute transactions. Progressive reduction causes to the Community Z-Score also dropping below $DyRT$ at 5340 ms, at which point the node is removed from the system. Post-removal, Z-Scores are no longer computed on the node, and its NTS is set to zero. The $DyRT$, being a system-level metric, continues to evolve in response to ongoing trust dynamics in the remaining network.

3.5 NTDM: Node Trust Decision Mechanism

The proposed trust management process includes a evaluation mechanism called the Node Trust Decision Mechanism (NTDM). As visualized in Figure 3.4, this protocol

governs two core operations: (a) the validation of transactions and block addition, and (b) the removal of malicious nodes.

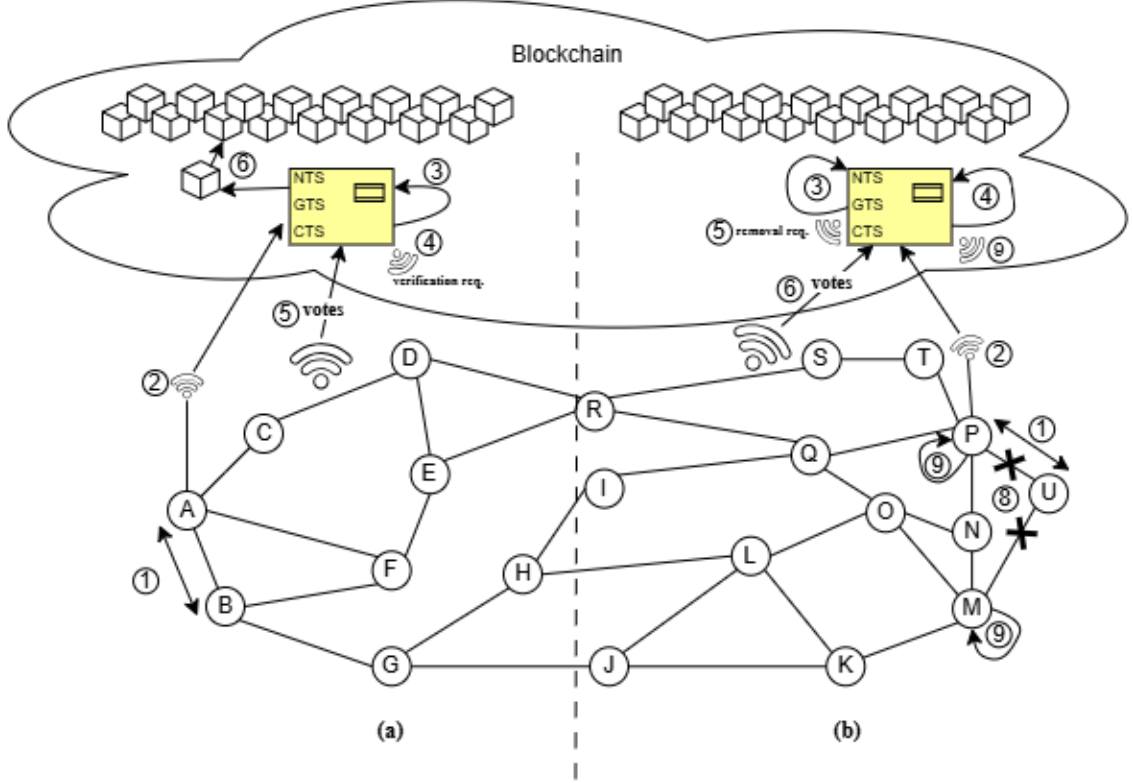


Figure 3.4: General structure and trust management workflow. (a) Block addition process via NTDM. (b) Node U's removal due to malicious behavior.

(a) Block Addition Process

- ① Node A sends a request $req(A \rightarrow B, data)$ to Node B for a service. Node B responds, after which Node A assigns a TS .
- ② Node A broadcasts the TS to the blockchain layer.
- ③ The smart contract receives this TS and updates Node B's NTS . Additionally, it recalculates the GTS values of all directly connected groups and the global CTS .
- ④ The smart contract initiates a $NTDM$ round by broadcasting the calculated TS , NTS , GTS , and CTS values to all active nodes.

- ⑤ Each active node independently recalculates the same trust metrics using their own records and verifies whether the smart contract's values are consistent. If the values align, the node sends an "Accept" vote; otherwise, it abstains or rejects.
- ⑥ If all active nodes accept the verification, the updated trust values are appended to a new block and committed to the blockchain.

Algorithm 1 *NTDM* Block Addition

Input: TS, NTS, GTS, CTS from Smart Contract

Output: Block addition decision

Init: $t = 0, M_{\text{accept}} = 0, A_{\text{active}} = \text{active nodes}$

```

1: for each  $TS$  in  $[T_{\text{start}}, T_{\text{max}}]$  do
2:   SC computes  $NTS, GTS, CTS$ 
3:   SC broadcasts trust values to active nodes
4:   for each active node  $X_j$  do
5:      $X_j$  recomputes all trust values
6:     if  $\text{match}(\text{SC}, X_j)$  then
7:       vote = Accept
8:        $M_{\text{accept}} += 1$ 
9:     else
10:      vote = Reject
11:    end if
12:  end for
13:  if  $M_{\text{accept}} == A_{\text{active}}$  then
14:    SC appends block
15:  else
16:    discard block
17:  end if
18:  if  $t \geq T_{\text{max}}$  then
19:    next round
20:  end if
21: end for

```

As shown in Algorithm 1, the proposed NTDM block addition process operates in batch mode, where all transactions within the interval $[T_{\text{start}}, T_{\text{max}}]$ are evaluated collectively. At the beginning of each round, nodes are classified as either active or isolated

based on NTS , GTS , and group/community-level Z-scores. Only active nodes participate in the verification process. After a new TS is received, the smart contract computes updated trust metrics (NTS , GTS , CTS) and broadcasts them to all active nodes. Each node independently recalculates these values and compares them with the received ones. If the values match, the node casts an “Accept” vote. Once all votes are collected, the system checks if $M_{\text{accept}} == A_{\text{active}}$. If so, the block is appended to the blockchain by the smart contract. Otherwise, the block is discarded. The protocol also enforces a timeout: if T_{max} is exceeded before agreement is reached, the system proceeds to the next round.

The node removal process involves several steps starting with evaluating node performance and ends with the node’s disconnection and network updates. Shown in Figure 3.4 (b).

(b) Node Removal Process

- ① Node P initiates a service request to Node U and completes the interaction. After the service, Node P evaluates Node U and assigns a TS .
- ② Node P broadcasts the TS to the blockchain layer.
- ③ The smart contract updates Node U ’s NTS and recalculates GTS and CTS values based on this interaction.
- ④ The smart contract evaluates Node U ’s behavior by comparing its NTS , Z_{group} , and $Z_{\text{community}}$ values against a dynamic removal threshold ($DyRT$).
- ⑤ If Node U is flagged as anomalous, the smart contract issues a NTDM removal request to all active nodes.
- ⑥ Each active node recomputes the trust metrics and verifies whether Node U satisfies the removal conditions. If the number of "Accept" votes reaches the majority of active nodes, the removal is approved, and the smart contract initiates block preparation.

- ⑦ Node U is disconnected from all its peers and becomes ineligible for future transactions.
- ⑧ The smart contract propagates updated GTS values to affected nodes and updates the global CTS accordingly.
- ⑨ The final removal result, including updated GTS and CTS , is broadcast to the network by the smart contract.

Algorithm 2 *NTDM Node Removal*

Input: TS, NTS, GTS, CTS for Node X_k

Output: Decision on removal

Init: $M_{\text{accept}} = 0, A_{\text{active}} = \text{active node count}$

```

1: SC evaluates  $X_k$  using  $NTS, Z_G, Z_C$ , and  $DyRT$ 
2: if  $X_k$  violates threshold then
3:   SC sends removal proposal to active nodes
4:   for each active node  $X_j$  do
5:      $X_j$  verifies  $X_k$ 's anomaly status
6:     if confirmed then
7:       vote = Accept
8:        $M_{\text{accept}} += 1$ 
9:     else
10:      vote = Reject
11:    end if
12:  end for
13:  if  $M_{\text{accept}} \geq \lceil A_{\text{active}}/2 \rceil$  then
14:    SC removes  $X_k$  and updates  $GTS, CTS$ 
15:    SC broadcasts removal result
16:  end if
17: end if

```

Algorithm 2 outlines the NTDM process workflow for node removal, which is invoked whenever a node's behavior significantly deviates from expected trust behaviors. All nodes are initially tagged as active or isolated based on recent trust activity. $DyRT$ is calculated by the smart contract using Z-scores at the group and community levels. If the node X_k shows extremely low Z_{group} and $Z_{\text{community}}$ values, both below its $DyRT$, it is

flagged to be removed. The smart contract then broadcasts a removal request to all active nodes. Each active node checks the trust metrics independently and votes. If the number of "Accept" votes is equal to or exceeds the majority of active nodes, the node is removed from the network and disconnected from its peers. The decision is then committed to the blockchain, and the updated trust state is broadcast to ensure global state consistency.

3.6 Simulation Setup

We conducted simulations using the OMNeT++ discrete event simulator [60]. Each graph represents the average of ten different runs, in each run, OMNeT++ selected pairs and TS s according to the calculations. Technical simulation details shown in Table 3.1. During the simulations we have tested our model and others against self-promoting, bad-mouthing, camouflage and collaboration attacks.

Table 3.1: *Simulation Metrics for TrustVist*

Number of Nodes	100
Each run duration	60,000
Episode	10 simulation runs
Simulation Tool	OMNeT++ 6.0+
Compiler	GCC 11.3.0
Operating System	Ubuntu 22.04 LTS (Linux Kernel 5.15+)
Framework	INET Framework 4.4

The proposed novel consensus protocol tested against multiple attack scenarios, and they listed as follows:

- **Self-promoting Attack:** A self-promoting attack is a very commonly used strategy

by the attackers, which aims to promote attacker owned nodes. Thus, attackers can introduce themselves as trusted nodes to manipulate the system.

- **Bad-mouthing Attack:** In bad-mouthing attack, attackers are collaboratively target a single node at the same time, to reduce its rating. This is an organized attack strategy for eliminating the specific node.
- **Camouflage Attack:** Attackers can behave partially trusted to deceive the trust mechanism. In the camouflage attack, attackers are act normally in specific percentage while they pursue their attacks.
- **Collaboration Attack:** In collaboration attack multiple malicious nodes are collaboratively attack to the system and manipulate the system in terms of the performance and TS .

3.7 Simulation Results and Evaluation

In this section our system model is evaluated in multiple categories as: Attack evaluations, decay evaluations, and system evaluations.

3.7.1 Attack Evaluations

In this subsection it is evaluated that the reflex capability of our proposed model against various attack models and compare it with similar models PeerTrust [1] and Semi-Centralized Trust Management Model [2]. It is emphasized that the importance of fast reaction to attacks and the ability to recover quickly after an attack.

Figure 3.5, initially (0–650 ms) all models observe increasing trust scores as nodes engage in regular transactions, including transactions with a malicious node that temporarily camouflages its bad behavior. At 650 ms, the malicious node begins to misbehave, its behavior changes, leading to bad service or misleading data. This causes a

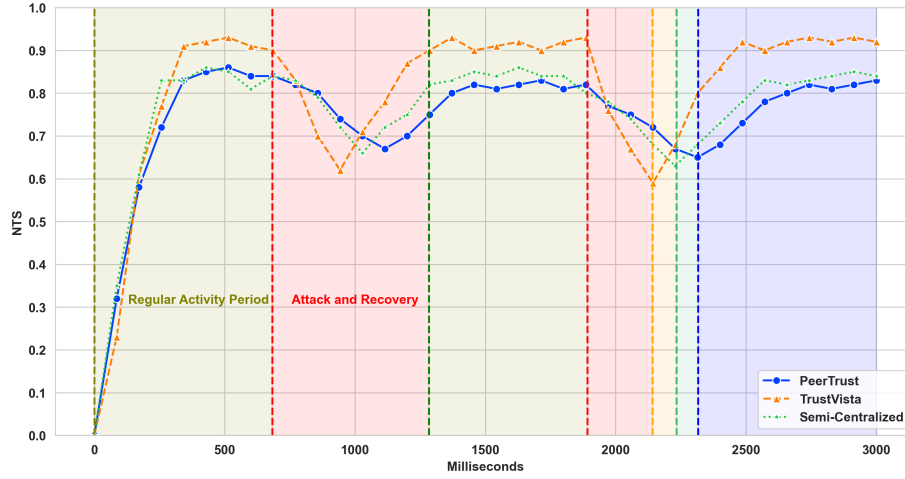


Figure 3.5: Comparison of model responses when a node experiences two consecutive camouflage attacks from the same malicious node in TrustVista.

decline in its trust score, especially TrustVista, which rapidly penalizes recent anomalies, quickly lowering the attacker’s trust score due to its time-sensitive decay mechanism (see Eq. 3.11), which dynamically prioritizes recent trust evidence over older, possibly outdated history. Crucially, our model initiates recovery earlier than others because the decay mechanism accelerates the detection and elimination of malicious nodes. Once the attack is recognized, the node is rapidly excluded from interactions, enabling the system to stabilize sooner. As a result, trust scores start increasing earlier compared to PeerTrust and Semi-Centralized models, which suffer from delayed detection due to cumulative or slower-adapting mechanisms. The observed rise in trust after the attack is not due to direct counteraction or re-engagement with honest nodes alone, but instead stems from our formulation’s ability to reduce the weight of past malicious behavior. This leads to a trust recalibration driven by recent activity and system-wide dynamics. PeerTrust, lacking such decay sensitivity, continues to reflect the impact of prior misbehavior for a longer duration, delaying its recovery.

Figure 3.6 illustrates a thorough comparison of the three models, PeerTrust [1], TrustVista, and a Semi-Centralized approach [2], under four distinct attack scenarios:

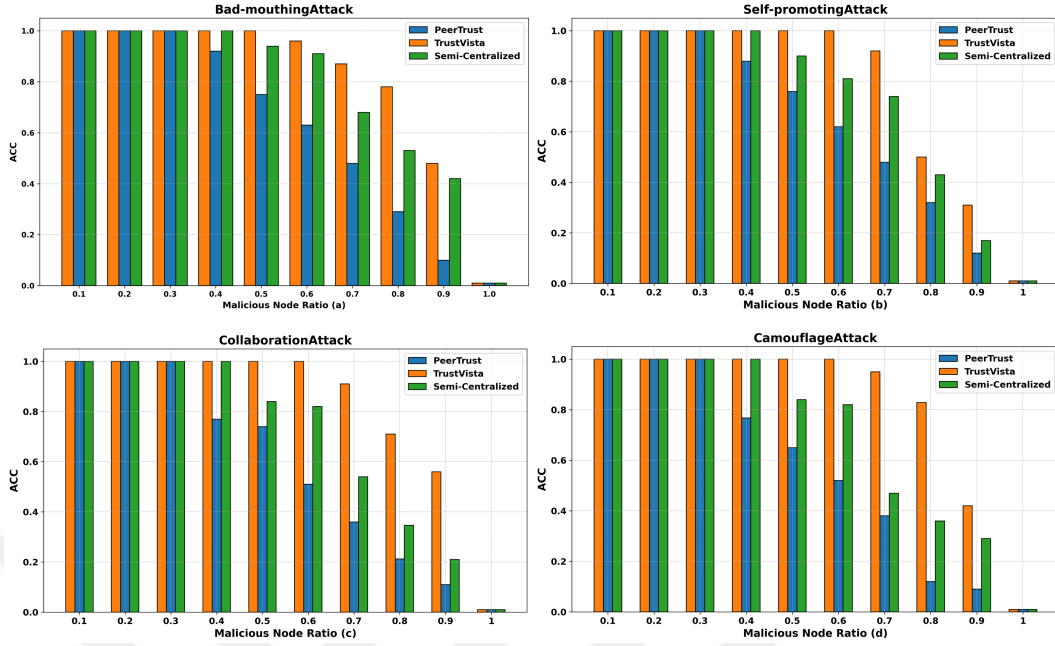


Figure 3.6: Comparison of system-wise accuracy ACC under different attack scenarios (a), (b), (c), (d) and MNRs for TrustVista, PeerTrust [1] and Semi-Centralized Trust Management Model [2].

(a) Bad-mouthing Attack, (b) Self-promoting Attack, (c) Collaboration Attack, and (d) Camouflage Attack. Each subfigure demonstrates system-wide Accuracy (ACC) against various MNR, ranging from 0.1 to 1.0 in increments of 0.1. For each experiment, the simulation is run 10 times for each MNR setting, and the average accuracy is calculated. ACC represents the proportion of correctly identified malicious and honest nodes, which clearly reflects the detection performance under increasing adversarial ratio. Also, computed as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.20)$$

(a) Bad-mouthing Attack: The TrustVista model maintains near-perfect accuracy up to MNR = 0.7 and exhibits graceful degradation beyond that. PeerTrust, on the other hand, begins to degrade after MNR = 0.4 and does not perform well after 0.6. Semi-Centralized performs marginally better than PeerTrust but falls behind our model in withstanding false negative classification. TrustVista’s adaptive trust assessment with decay

withstands the effect of colluded misinformation for longer.

(b) Self-promoting Attack: Malicious nodes cooperatively promote each other to build artificial credibility. TrustVista still sustains near-optimal accuracy up to 0.8 MNR, showing insensitivity to self-bias. PeerTrust and Semi-Centralized degrade more rapidly, especially after $\text{MNR} = 0.6$. The reason is that our system downgrades artificially maintained past trust by decay and thus is more immune to self-promotion attacks.

(c) Collaboration Attack: Malicious nodes collaboratively attack to the network. This is one of the most difficult to detect, since many malicious nodes attack at the same time and it cause high fluctuations. Even so, TrustVista maintains a significantly higher accuracy rate after $\text{MNR} = 0.5$, whereas both PeerTrust and Semi-Centralized collapse rapidly beyond this point. The performance gap widens as TrustVista dynamically suppresses coordinated false trust due to decaying influence from frequent but transient interactions.

(d) Camouflage Attack: Malicious nodes behave honestly in the beginning to gain trust before launching an attack. This attack targets models with high memory retention. Our TrustVista model gives the best result in this case, maintaining over 80% accuracy up to $\text{MNR} = 0.8$, while PeerTrust and Semi-Centralized sharply degrade after 0.5. This indicates that our decay function effectively minimizes the long-term weight of early-stage camouflage behavior, leading to quicker adaptation to behavior change.

Overall, TrustVista performs better than PeerTrust and the Semi-Centralized models for all types of attacks and malicious node ratios. Its recent interaction prioritization and dynamic decay make it highly resistant to deception attacks and responsive to evolving threats. This further enhances the robustness of our trust model to maintain system integrity under highly adverse adversarial environments.

3.7.2 Decay Evaluations

Proposed a novel "Multi-factor Dynamic Decay" considers multiple factors and dynamically adapts to the current system state, it evaluated the performance of our decay function within different metric values. To regulate how much influence past interactions should have on the current trust score, we introduced a dynamic decay sensitivity coefficient, denoted by λ (see Eq. 3.12). This coefficient adjusts the rate at which older trust evidence is discounted based on two core factors: overall system activity and total number of transactions. A higher λ results in faster decay, while a lower λ allows trust to retain historical influence longer. To evaluate the effect of this design, we assess the behavior of our decay model under different λ settings and compare it with the static approach Semi-Centralized Trust Management Model [2]. In Figure 3.7, during the evaluation it indicated how the λ changes affects NTS , since our λ is dynamic it changes during the run. Thus, λ depends on multiple metrics. As T increases over the time of the system's run, λ increases as well. This increase in λ leads to a deceleration of the NTS 's rise and oppositely, accelerates its decline when TS decreases. The longer the system runs, the more sensitive the system becomes in detecting and responding to malicious nodes.

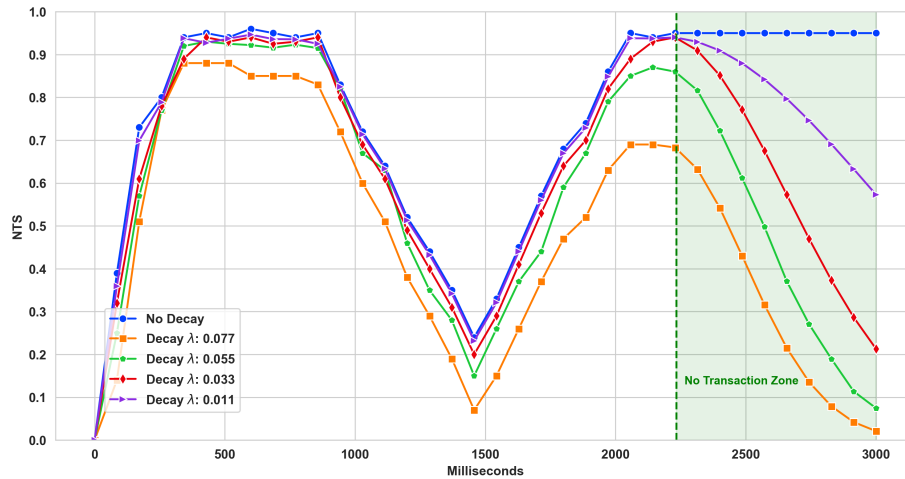


Figure 3.7: Impact of varying λ rates on NTS in regular, recovery, attack and no transaction periods in TrustVista.

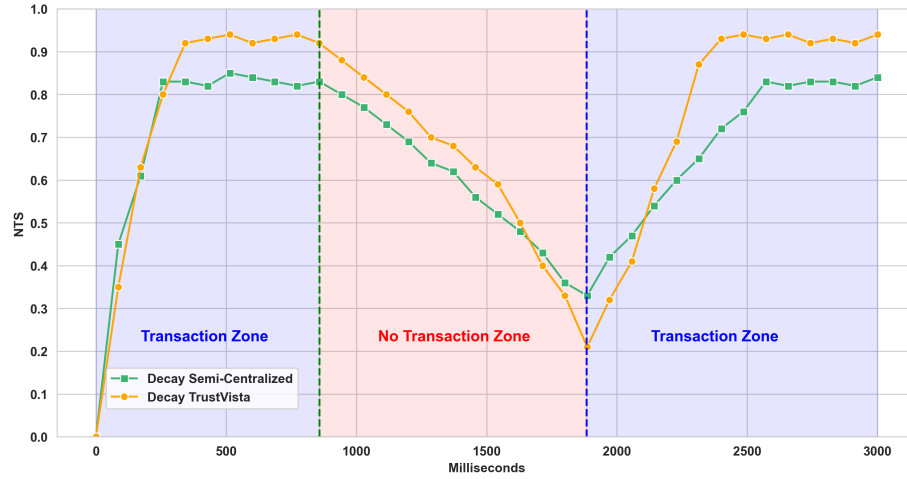


Figure 3.8: Comparison of the decay functions’ impact on the NTS in an attack-free environment between TrustVista and Semi-Centralized Trust Management Model [2].

Figure 3.8 presents a comparison of decay functions between the [2] model and this proposed model within an attack-free environment. Since, Equation 3.11 uses multiple factors NTS decreases more rapidly than in [2]. System evaluates the situation as if there is no transaction, causing the node become useless and should be terminated. In cases of no transactions, our model views this as possible inefficiency or malicious intent, leading to a faster decline in TS . When the transaction zone activated TrustVista sharply rises and stabilizes itself.

3.7.3 System Evaluations

In this section, it is evaluated how our model reduces the malicious activity across the network and compare them also, which discussed model special issues 3.16, 3.17, 3.15. We implemented a 50% MNR under a bad-mouthing attack scenario in Figure 3.9, and recorded the number of successful attacks occurring in each 1/10th of the total simulation time. Attacks started at 5,800 ms. The goal was to assess how effectively each trust model identifies and mitigates malicious behavior over time. Although the Semi-Centralized model initially registers slightly fewer attacks in the early stages, our proposed TrustVista model begins to outperform both baselines after approximately 25,000

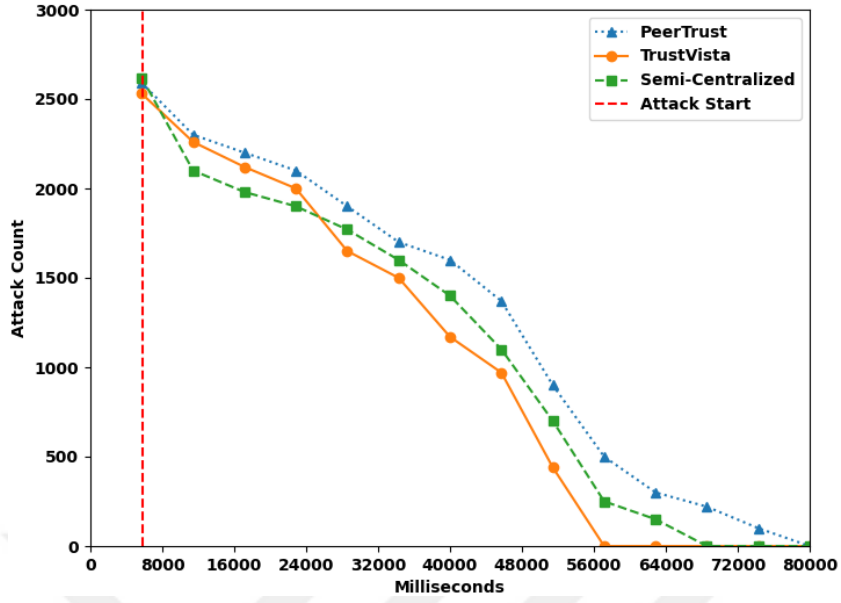


Figure 3.9: Comparison of attack reduction during bad-mouthing attacks, from system initialization to termination.

ms. The attack count under TrustVista steadily reduces and reaches zero, while other models continue to experience residual attack activity. Our malicious node detection and elimination strategy effectively identifies and removes threats, sharply reducing attack attempts over time. TrustVista’s adaptive trust decay and Z-score-based removal mechanisms highlight the value of dynamic trust evaluation and early detection for robust, long-term resilience against persistent threats. Figure 3.10 shows the number of remaining malicious nodes as a function of total transactions under a 50% MNR during a bad-mouthing attack. The TrustVista model consistently removes attackers faster than both PeerTrust and the Semi-Centralized model. TrustVista achieves full elimination significantly earlier, enabling a longer attack-free operational window. In contrast, [1] shows the slowest reaction and weakest detection efficiency, while [2] falls behind TrustVista by approximately 3000 transactions. For each model, the simulation was run to observe how the transaction time changed as the ratio of malicious nodes was reduced, as shown in

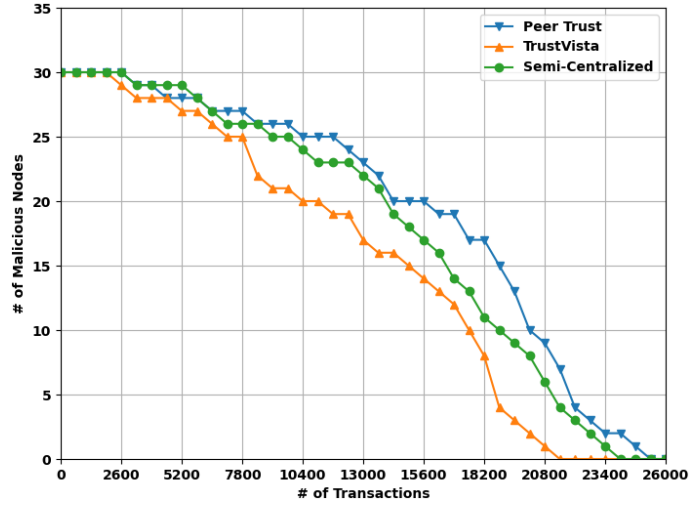


Figure 3.10: *Malicious node elimination over number of transactions for a 50% MNR under a bad-mouthing attack scenario.*

Figure 3.11. The figure illustrates the impact of malicious node elimination on transaction processing time under bad-mouthing attack. The x-axis represents the MNR, which reduces step by step while malicious nodes are detected and removed. At each step,

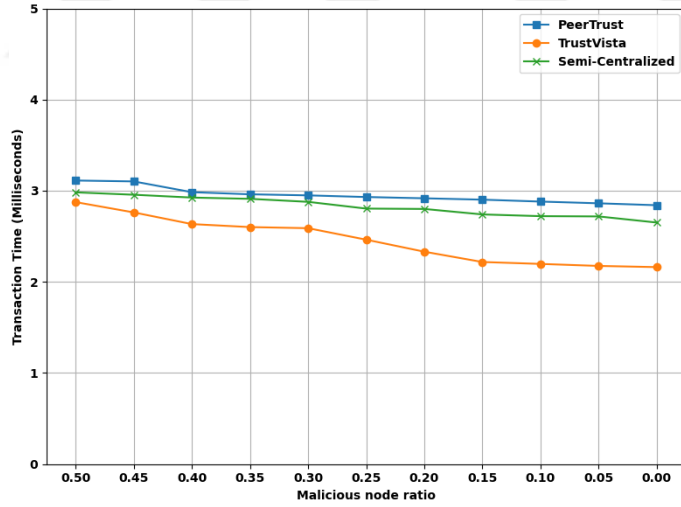


Figure 3.11: *Impact of malicious node reduction on transaction speed under bad-mouthing attack.*

the transaction time is measured to capture the real-time effect of network purification. As shown, all models experience a decrease in transaction time as MNR drops, indicating reduced overhead in trust calculations and consensus validation. Among them, the

TrustVista model consistently maintains the lowest transaction time, completing operations within 2.162 ms when all malicious nodes are removed, indicating its high detection and system stabilization capabilities. This corresponds to a performance gain of approximately 15% over PeerTrust and 13% over Semi-Centralized models in a fully secure environment.



4. EDPOS: ENERGY-AWARE DELEGATED PROOF OF STAKE RESOURCE-CONSTRAINED DEVICES

DPoS consensus mechanism offers significant improvements over traditional consensus algorithms such as Proof of Work (PoW) and Proof of Stake (PoS), especially in terms of scalability and democracy. Nonetheless, DPoS still falls short in environments where resource constraints are critical, such as IoT networks. One of the major drawbacks is that the lack of awareness regarding the energy levels of nodes. Since, electing low energy nodes as delegates can cause to block validation failures, weakening network longevity and sustainability.

To address this limitations, we propose the EDPoS model, which dynamically integrates energy awareness into each phase of the DPoS process. Unlike conventional solutions that rely solely on voting power, our model prioritizes delegates based on both their real-time energy status, network conditions and votes. This chapter proposes EDPoS model in detail, including all components of the model such as, adaptive block size adjustment, dynamic round-robin scheduling, energy-aware delegate filtering, and historical energy consumption analysis.

4.1 Overview of EDPoS Architecture

In the proposed EDPoS architecture, one of the main improvements over traditional DPoS is the dynamic consensus execution mechanism. Unlike DPoS, our model select delegates with the combination of votes and real time energy levels as shown in Figure 4.1. Specifically, the scheme suggests a dynamic round-robin mechanism, which replaces the static block validation cycle with a more adaptable approach that actively responds to fluctuating node energy levels. This dynamic scheduling renders that the consensus process remains continuous and avoids block generation from being blocked by

Traditional DPoS

EDPoS

of Vote

of Vote + % of Energy Level

Figure 4.1: *EDPoS structure*

energy depleted delegates. In each round of consensus, the system monitors the energy

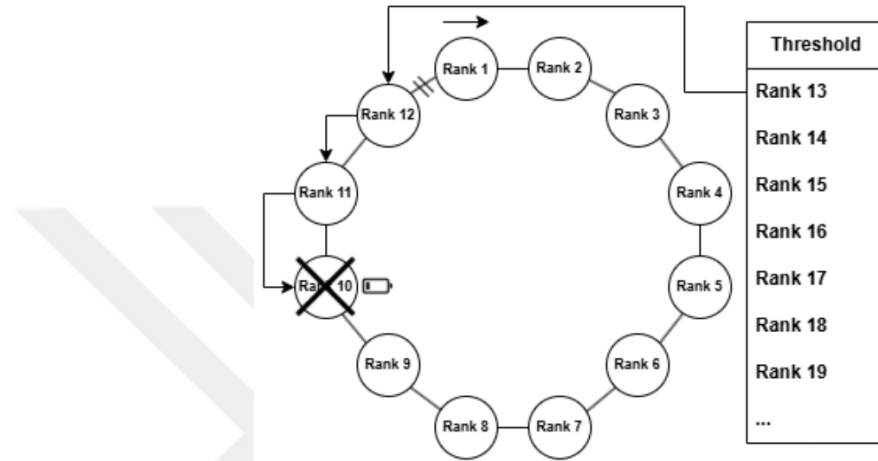


Figure 4.2: *Dynamic Round Robin structure and illustration of node replacement*

level of each elected delegate remaining. If a delegate is critically low on energy before its turn (according to the adaptive threshold), the system automatically replaces such a delegate with a backup node, elected from the candidate pool based on the highest available energy. The replacement mechanism prevents downtime, also eliminates reliance on unstable nodes. Dynamic round robin ensures that the selection of delegates and the validation of blocks are kept energy-efficient and sustainable. As it shown on Figure 4.2, the order of procedure is as follows:

1. **Delegate Selection:** At the beginning of each round, delegates are chosen based on votes and real-time energy levels.
2. **Round-Robin Execution:** Delegates take turns producing blocks based on their election order while their energy levels continue to be monitored.

3. **Delegate Removal:** If a delegate's energy drops below a critical threshold before its turn, it is immediately replaced by a backup node.
4. **List Update at Round End:** After all delegates have completed their turn, a new list of delegates is generated, factoring in any changes in energy levels.

In addition, the model also enables adaptive block size adjustment. Since block size directly influences the computational and energy burden on validator nodes, the system dynamically adjusts the number of transactions per block according to the average energy level of the current delegate set. If the network possesses low available energy, the block size is reduced to ensure successful validation; if there are ample energy resources, the maximum block size is reached to gain maximum throughput.

The internal operation of this combined mechanism can be summarized as follows:

- **Real-Time Monitoring:** During consensus rounds, the energy level of every delegate is monitored in real-time.
- **Delegate Replacement:** Nodes falling below the critical energy threshold are replaced dynamically with the most energy-efficient backup node.
- **Adaptive Block Size Adjustment** The system calculates average energy across delegates and scales block size accordingly to ensure efficiency without loss of validation failure.
- **End-of-Round Updates:** At round-end, pool of delegates gets updated according to fresh energy and voting data.
- **Adaptive Energy Thresholding:** The system updates the minimum energy requirement for candidate eligibility based on real-time network conditions.

- **Adaptive Delegate Count:** The number of active delegates per round is flexibly adjusted to ensure reliable consensus while conserving energy.

This modular and adaptive approach enables the EDPoS model to remain operational under variable energy conditions, improving system resilience and sustainability, especially in constrained and dynamic environments such as UAV-assisted or IoT-based disaster networks.

4.2 Adaptive Block Size Adjustment

In conventional DPoS systems, block size remains static across rounds, regardless of the current energy conditions of the network. This static design may lead to suboptimal performance in environments where nodes operate with limited energy resources. Fixed block sizes usually cause to energy inefficiency in networks, particularly in wide networks. Major problems are listed below:

- A large block size (high transaction count per block) requires more computational resources, which can overload low-energy nodes, accelerating their depletion.
- A small block size results in low throughput, reducing the efficiency of the blockchain.

Energy-Aware block size adjustment mechanism adjusts the number of transactions per block according to network-wide energy levels. It dynamically scales the number of transactions per block based on the system's overall energy level, ensuring that when energy is high, then blocks will expand to hold more transactions for higher throughput, while in cases of low energy, block sizes are reduced to minimize computational load and prevent node failure. The goal is to maintain a balance between energy consumption and network throughput while minimizing validation failures. The dynamic adjustment of block size is formulated in Equation (4.1):

$$T_{\text{block}} = T_{\text{prev}} \times \left(1 + \kappa \times \frac{E_{\text{avg}}^{(t)} - E_{\text{avg}}^{(t-1)}}{E_{\text{avg}}^{(t-1)}} \right) \quad (4.1)$$

As shown in Equation (4.1), it dynamically adjust the block size in terms of system's overall energy, where we introduce sensitivity factor (κ) which decides how aggressively the transaction capacity per block (T_{block}) takes responds to fluctuation in network energy. Here, T_{block} represents the transaction capacity (i.e., block size) for the current round, and T_{prev} denotes the block size used in the previous round. This ensures that, in the case the network's average energy level at the current round t goes down ($E_{\text{avg}}^{(t-1)} > E_{\text{avg}}^{(t)}$), the block size is trimmed down to reduce energy consumption. In contrast, if the network's energy level rises on average ($E_{\text{avg}}^{(t)} > E_{\text{avg}}^{(t-1)}$), the block size expands for adding more transactions per block, hence increasing throughput. To moderate the sensitivity of block size adjustments, a dynamic scaling factor κ is introduced, which is calculated based on the energy variation trend in the network. This factor is defined as Equation (4.2):

$$\kappa = \frac{|E_{\text{avg}}^{(t)} - E_{\text{avg}}^{(t-1)}|}{\sigma_E} \quad (4.2)$$

In Equation (4.2), σ_E is the standard deviation of energy levels across the last several rounds. The standard deviation helps regulate how radically the system reacts to energy fluctuations, if the energy levels have been historically stable, it reacts more aggressively to changes, while in a highly volatile environment, it ensures that block size changes are more controlled. When energy levels have historically fluctuated heavily, σ_E is large, which results in a smaller κ , ensuring more conservative adjustments. Conversely, it leading to gradual changes, while when σ_E is low, κ has a stronger effect, causing sharper block size adjustments. This two-step mechanism ensures that the block size:

- Decreases when average network energy drops, conserving power.
- Increases when more energy is available, improving throughput.

- Remains stable during high fluctuation periods to avoid instability.

However, to prevent excessive reductions in block size that could degrade network performance, we enforce $T_{\min}$ is enforced, which guarantees that even under prolonged low-energy conditions, the system can continue processing transactions. The status of T_{block} should be as $T_{\text{block}} \geq T_{\min}$ to ensure a minimum level of transactions applicable. The calculation of $T_{\min}$ shown in Equation (4.3).

$$T_{\min} = \frac{E_{\min}}{C_{\text{txn}}} \quad (4.3)$$

Where $E_{\min}$ is the minimum allowable energy per delegate, and C_{txn} is the estimated energy cost for processing a single transaction. This formulation guarantees that block size does not exceed the energy constraints of the network and it secures the lower bound so the system does not crash, dynamically adapting to fluctuations in available energy.

4.3 Adaptive Energy Threshold

In energy-sensitive environments such as IoT-based networks or UAV-assisted systems, selecting delegates without considering their energy sufficiency can lead to unstable consensus operations and frequent block validation failures. Conventional blockchain systems operate with fixed delegate thresholds to remove ineffective nodes. To address this, the EDPoS model introduces an Adaptive Energy Threshold mechanism that dynamically determines the minimum energy level required for a node to be eligible as a delegate in each round. By introducing this function we aim to solve following fixed energy threshold problems, such as unnecessary delegate removal:

- If the threshold is too high, many nodes are removed prematurely, leading to centralization and security risks.
- If the threshold is too low, low-energy nodes continue operating, increasing the chance of block validation failures and system instability.

Proposed threshold is not static; it adapts to current network-wide energy conditions and transaction load by incorporating three key parameters:

- The current block size T_{block} .
- The average energy of all active delegates E_{davg} .
- A dynamically computed minimum bound $E_{\text{th}}^{\text{min}}$.

The formula ensures that only nodes with enough energy continue validating blocks while preventing excessive delegate removal when system-wide energy levels are low. It is given as in Equation (4.4):

$$E_{\text{th}} = \max \left(\log(T_{\text{block}} + 1) \times E_{\text{davg}}, E_{\text{th}}^{\text{min}} \times E_{\text{davg}} \right) \quad (4.4)$$

This formula ensures several important properties:

- The logarithmic scaling with T_{block} smooths the impact of transaction load fluctuations on energy requirements. Minor changes in block size result in minimal threshold adjustments, while larger blocks demand proportionally more energy.
- The scaling by E_{davg} reflects system-wide energy availability. When the network is energy-rich, the threshold rises, favoring high-energy nodes. Conversely, during low-energy conditions, the threshold relaxes to prevent over-filtering.
- The use of the maximum function ensures that the threshold never drops below a baseline energy level necessary for reliable participation.

The minimum adaptive threshold component $E_{\text{th}}^{\text{min}}$ is defined as in Equation (4.5):

$$E_{\text{th}}^{\text{min}} = \frac{E_{\text{davg}}^2}{T_{\text{block}} + 1} \quad (4.5)$$

Finally, a minimum threshold value (E_{th}^{min}) is enforced to prevent extreme conditions where the threshold becomes too low. This ensures that even at an extremely low system-wide energy level, a baseline threshold values prevent the election of entirely non-functional delegates, maintaining blockchain continuity. The dynamic adjustment of the model to both system-wide energy conditions and transaction loads, this model enhances energy efficiency, sustainability in a system and stability in a network. If the energy of a delegate falls below the computed threshold ($E_d < E_{th}$), it will then be removed from the network while delegates whose energy is greater than or equal to this threshold ($E_d \geq E_{th}$) value resume their validation tasks. This adaptive model enhances:

- Energy efficiency, by removing weak nodes only when necessary.
- Sustainability, by preserving functionality even during energy scarcity.
- Network stability, by avoiding abrupt changes and centralization risks.

This self-regulating mechanism prevents excessive removal of delegates, maintains stability across the network, and energy efficiency without requiring any manual tuning, which supports the adaptivity.

4.4 Adaptive Delegate Count

In traditional DPoS, delegate selection is solely based on stake-based voting, where nodes with the most votes become block validators. While effective in many blockchain applications, this approach is problematic in resource-constrained environments such as IoT or UAV-enabled networks, where energy availability plays a crucial role in node performance and network reliability. To overcome these limitations, the ED-PoS model integrates an Adaptive Delegate Count mechanism that incorporates energy-awareness directly into the delegate selection and participation process.

$$E_{th}^{cand} = \max(\log(T_{block, prev} + 1) \times E_{davg, prev}, E_{th, prev}^{min} \times E_{davg, prev}) \quad (4.6)$$

However, in EDPoS, we introduce an additional energy-based selection mechanism is introduced such that only nodes with sufficient energy levels participate in the block validation and this prevents low-energy delegates from becoming validators, thereby enhancing the sustainability and efficiency of the blockchain network and shown in Algorithm 3. We calculate the candidacy threshold as in Equation (4.6).

Algorithm 3 Delegate Selection

Input: Candidate nodes $\mathcal{N}$, votes V_n , energy levels E_n .

Output: Selected delegates $\mathcal{D}$, Backup nodes $\mathcal{B}$.

```

1: ① Phase 1: Energy-Filtered Voting
2: for each node  $n \in \mathcal{N}$  do
3:   if  $E_n \geq E_{th}^{cand}$  then
4:     Add  $n$  to  $\mathcal{V}_{eligible}$ 
5:   end if
6: end for
7:  $(\mathcal{D}_{candidates}, \mathcal{B}) \leftarrow \text{SelectDelegates}(\mathcal{V}_{eligible}, V_n, N_{max})$ 
8: ② Phase 2: Delegate Selection and Backup Nodes
9: for each delegate  $d \in \mathcal{D}_{candidates}$  do
10:   compute weighted selection score:

$$S_d = \lambda \times \frac{V_d}{V_{max}} + (1 - \lambda) \times \frac{E_d}{E_{max}}$$

11: end for
12: sort  $\mathcal{D}_{candidates}$  by  $S_d$  in descending order
13: assign top-ranked delegates to  $\mathcal{D}$  (final delegate set)
14: while block production is active do
15:   for each delegate  $d \in \mathcal{D}$  do
16:     if  $E_d < E_{th}^{post}$  then
17:       Remove  $d$  from  $\mathcal{D}$ , select top backup from  $\mathcal{B}$ 
18:     end if
19:   end for
20: end while
21: Return final Delegates  $\mathcal{D}$ , Backup Nodes  $\mathcal{B}$ 

```

The following enumeration provides a overview of the delegate selection process, which is systematically illustrated upon in the subsequent sections as follows:

① **Phase 1:**

- Before the voting process begins, each node's energy level (E_n) is compared against the pre-calculated Candidacy threshold (E_{th}^{cand}). Only nodes meeting or exceeding E_n qualify to participate in voting.
- Token holders votes only for the eligible nodes.
- The top N_{max} nodes with the highest votes move to Phase 2. N_{max} represents the maximum possible number of delegates that can be selected for Phase 2.

② **Phase 2:** Delegates take turns producing blocks based on their election order while their energy levels continue to be monitored.

- Instead of sorting by energy alone, we compute a weighted score for each delegate by calculating delegate selection score ($S_{delegate}$).
- The highest- $S_{delegate}$ delegates finalize the block validation order.
- Backup nodes are selected from the remaining high-voted candidates by their vote count.

Equation (4.7) formalized for EDPoS, each eligible candidate in the second phase of the delegate selection process is assigned a delegate score delegates based on both their vote and energy levels, $S_{delegate}^c$. Delegates are then ordered by their $S_{delegate}$, determining the sequence in which they validate blocks, ensuring that we prioritize both highly trusted and energy-sufficient nodes for block validation. V stands for votes of the delegate, E_c is candidate's energy and λ is a tunable weight factor that balances the importance of voting versus energy. The parameter λ controls the trade-off between stake-based and energy-aware selection strategies. A higher λ results in a delegate selection process more closely aligned with traditional DPoS, favoring vote-heavy nodes. Conversely, a lower λ shifts

the emphasis toward energy efficiency, promoting nodes with higher remaining energy to ensure operational sustainability.

$$S_{\text{delegate}}^c = \lambda \times \frac{V}{V_{\text{max}}} + (1 - \lambda) \times \frac{E_c}{E_{\text{max}}} \quad (4.7)$$

In addition to this ranking strategy, EDPoS introduces a self-regulating mechanism for determining the number of delegates per round. Many popular blockchain applications use fixed number of delegates for block production, for instance EOS [62] uses 21 delegates, and TRON [63] uses 27 validators for their system. Rather than fixing the number of delegates statically, the system dynamically adjusts the active delegate count based on overall network energy availability. This ensures that the validator set remains both scalable and energy-aware. The number of active delegates, $N_{\text{delegates}}$, is calculated as in Equation (4.8):

$$N_{\text{delegates}} = \frac{E_{\text{total}}}{E_{\text{node, avg}}} \quad (4.8)$$

Instead of fixing the maximum number of delegates, the number of selected delegates is in proportion to the sum of the total energy in the system compared to the average energy per node. E_{total} is total energy of all candidate nodes, while $E_{\text{node, avg}}$ is average energy per candidate node, and defined as follows:

$$E_{\text{node, avg}} = \frac{E_{\text{total}}}{N_{\text{nodes}}} \quad (4.9)$$

This adaptive formulation in Equation (4.9) ensures that:

- When the total energy in the system is high, more delegates are selected, promoting decentralization and higher throughput.
- When energy levels are low, the delegate count is automatically reduced, conserving resources and maintaining network stability.

Overall, this dynamic adjustment mechanism eliminates the need for manually defined delegate limits and provides a fair, scalable, and energy-resilient validator selection framework, well-suited for decentralized networks operating under fluctuating energy constraints.

4.5 Historical Energy Consumption Analysis

While real-time energy monitoring is crucial to deciding that selected delegates are currently capable of fulfilling their responsibilities, it does not account for long-term energy efficiency. A node may have high residual energy at the moment of selection, yet may historically exhibit poor energy usage behavior, leading to premature exhaustion and unreliable performance. To cover this deficiency, the EDPoS framework introduces a Historical Energy Consumption Analysis module in the delegate scoring.

This mechanism involves an extra scoring function that expands the initial delegate score to cover the node's average per-transaction energy consumption to incentivize nodes that have been consistent in being energy-efficient in the long term. The refined delegate score S_{delegate}^c in Equation (4.7) is defined as follows:

$$S_{\text{delegate}}^t = \lambda \times \frac{V}{V_{\max}} + (1 - \lambda) \times \left(\frac{E_c}{E_{\max}} + \left(1 - \frac{E_{\text{avg-use}}}{E_{\text{cap}}} \right) \right) \quad (4.10)$$

In Equation (4.10), $E_{\text{avg-use}}$ is average energy consumption per transaction over a predefined historical window, while $1 - \frac{E_{\text{avg-use}}}{E_{\text{cap}}}$ is newly introduced component which serves as a normalized historical efficiency metric. Nodes that consume less energy per transaction over time achieve higher values in this term, boosting their delegate score and increasing the likelihood of selection. This three-part scoring structure offers a multi-dimensional evaluation that balances:

- Voting trustworthiness (based on stakeholder support).

- Real-time operational capability (based on current energy).
- Sustainable behavior (based on past energy usage trends).

The use of historical efficiency enables the system to punish energy-wasteful nodes, despite their high current energies. Conversely, nodes that demonstrate consistent and frugal energy consumption are rewarded, promoting long-term network stability and block validation consistency.

By incorporating both instantaneous and trend-based metrics, the EDPoS model ensures a robust delegate selection strategy that accommodates with the goals of energy-efficient and sustainable blockchain networks, especially in environments where energy is scarce and must be conserved intelligently.

4.6 Simulation Setup

We evaluated the performance of proposed EDPoS model through simulations using the OMNeT++ simulator, [60]. In general, node placement and initial energy are randomly assigned while the transaction load and energy distribution among nodes depend on the scenario, in order to achieve high diversity in the system conditions. Table 4.1 indicates the simulation metrics.

Ten different battery capacity levels were defined for simulation: 1500, 2200, 3000, 4500, 5200, 6000, 7500, 10000, 12000, 15000 mAh. These values were randomly assigned to nodes in each run to ensure heterogeneous energy distribution across the network.

The energy of each node is expressed in Joules using the following conversion Equation (4.11):

$$E_{\text{Joule}} = V_{\text{battery}} \times E_{\text{mAh}} \times 3.6 \quad (4.11)$$

Where E_{Joule} represents the total energy in Joules, V_{battery} is the nominal battery voltage

Table 4.1: *Simulation Metrics for EDPoS Consensus Mechanism*

Number of Nodes	100
Number of Rounds	150
Delegate Count for DPoS	10
Episode	10 simulation runs
Battery Voltage	3.7 V
Energy Consumption per Packet	0.1% of node capacity
Energy Conversion Factor	3.6
Battery Capacity Levels	Varies from 1500 mAh to 15000 mAh
Simulation Tool	OMNeT++ 6.0+
Compiler	GCC 11.3.0
Operation System	Ubuntu 22.04 LTS (Linux Kernel 5.15+)
Framework	INET Framework 4.4

(3.7 V), and E_{mAh} denotes the battery capacity in milliamper-hours. The factor 3.6 is used to convert milliamper-hours into Coulombs ($1Ah = 3600C$). This equation ensures accurate energy conversion for evaluating battery depletion and system longevity.

To investigate the impact of energy-awareness in consensus mechanisms, we compared three different models (Traditional DPoS, EDPoS without Historical Data and EDPoS with Historical Data) with increasing levels of energy integration:

- **DPoS:** DPoS model follows the classic stake-based voting mechanism, where delegates are selected purely based on the number of votes they receive from token holders. In this model, there is no energy-awareness, meaning a delegate's energy

level does not influence its election or block validation performance. It uses a vote selection process, whereby holders vote for the validators, and then chooses the top N most voted nodes as block validators responsible for the production and validation of blocks.

- **EDPoS without historical data:** The EDPoS without historical data model improves upon DPoS by considering real-time energy awareness within the process of delegate selection process. Only those nodes that meet E_{th} are eligible to participate in voting, ensuring that low-energy nodes are filtered out before the election process begins. Once the energy check is completed, stake-based voting takes place as usual and the top N most-voted nodes are selected as final candidates. These candidates are then ranked based on their energy levels, determining the order in which they validate blocks. This approach prevents the election of energy-deficient delegates, significantly improving the sustainability and reliability of the blockchain network. However, because only real-time energy levels are considered, this model does not account for how efficiently a node has managed its energy over time.
- **EDPoS with historical data:** The EDPoS with historical data model enhances the energy awareness to a wider perspective: besides the real-time energy level, the historical trend of energy consumption, Equation (4.10), also plays a very important role in the delegate selection process. Similar to EDPoS without historical data, this model ensures that only nodes that meet or exceed the E_{th} are eligible for voting. However, instead of relying only on real-time energy levels, the top N most-voted nodes are further ranked based on their past energy consumption patterns. This means that a delegate's long-term energy stability plays a key role in determining its selection. If a node has a history of high energy usage, it may be deprioritized,

even if its current energy level is sufficient. This additional filtering mechanism improves delegate stability, ensuring that nodes that consistently manage their energy efficiently are prioritized for block validation.

A number of performance metrics were selected to be used in assessing the scalability, efficiency, and sustainability of each model. These performance metrics were tracked and recorded over a number of simulation runs to ensure statistical consistency and robustness of results. Each metric plays a crucial role in identifying the model's strengths and weaknesses of the model under real-world-like conditions, where node failure due to energy depletion or block verification delays would significantly undermine the network. The following key metrics are used to assess and compare the performance of each model:

- **System Longevity:** System longevity refers to the duration for which nodes remain active and functional in the network before depleting their energy, directly impacting the sustainability and reliability of the system.
- **Variance of Energy Consumption per Round:** Evaluating the variance helps in understanding the stability and fairness in energy consumption among nodes, highlighting the consistency of energy distribution.
- **Average Remaining Energy (%) Over Time:** This parameter illustrates the sustainability of the models by tracking the depletion of energy resources over consecutive rounds, offering a clear comparison of energy conservation capabilities.
- **Average Throughput per Round:** The throughput metric, represented by the number of verified transactions per round, reflects the processing efficiency and overall performance of the consensus mechanisms under varying conditions.
- **Number of Blocks Added:** This metric indicates the productivity of each model

by quantifying the total number of blocks successfully added to the blockchain throughout the simulation.

- **Number of Transactions per Block:** This analysis helps assess the scalability of the models by examining how the transaction load is managed within each block over time.
- **Block Validation Failure Rate:** This calculates the proportion of blocks that cannot be validated because of failure due to node inefficiencies or low energy levels or any other reasons.

4.7 Simulation Results and Evaluation

To enhance the understanding, simulation analysis split into several subsections according to our novel features as follows:

4.7.1 Adaptive Delegate Count

To evaluate adaptive delegate count feature, number of blocks added test simulated the system operated over 150 rounds, with different consensus models tested: DPoS, EDPoS without historical data, and EDPoS with historical data with 100 nodes. Figure 4.3

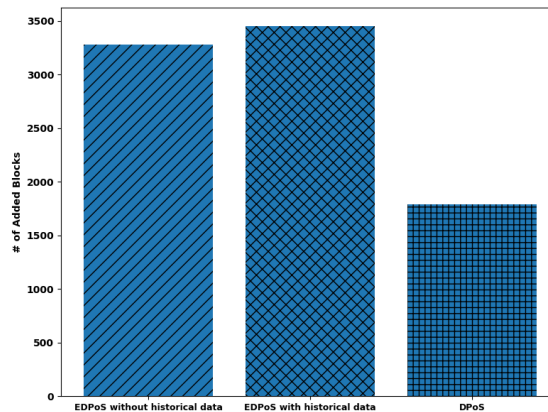


Figure 4.3: *Number of Blocks Added for EDPoS Tests*

illustrates the block addition capabilities of each model. EDPoS with historical data takes the lead with 3,450 blocks added, due to its adaptive delegate count innovation and historical energy data analysis. These mechanisms optimize energy consumption and transaction throughput while ensuring continuous block production. Additionally, its robust backup delegate activation strategy also extends node longevity, preventing system failures. EDPoS without historical data follows closely with approximately 3,300 blocks, a remarkable 83% increase versus DPoS. This leveraging an energy-aware selection mechanism, though it lacks the adaptive efficiency of EDPoS with historical data. In contrast, DPoS significantly lags behind with only 1,800 blocks added, as its static delegate selection method overloads certain nodes, causing rapid energy depletion and reduced block generation capacity. Figure 4.4 compares the system longevity for the different models, showing the ratio of remaining active nodes against simulation rounds. The traditional

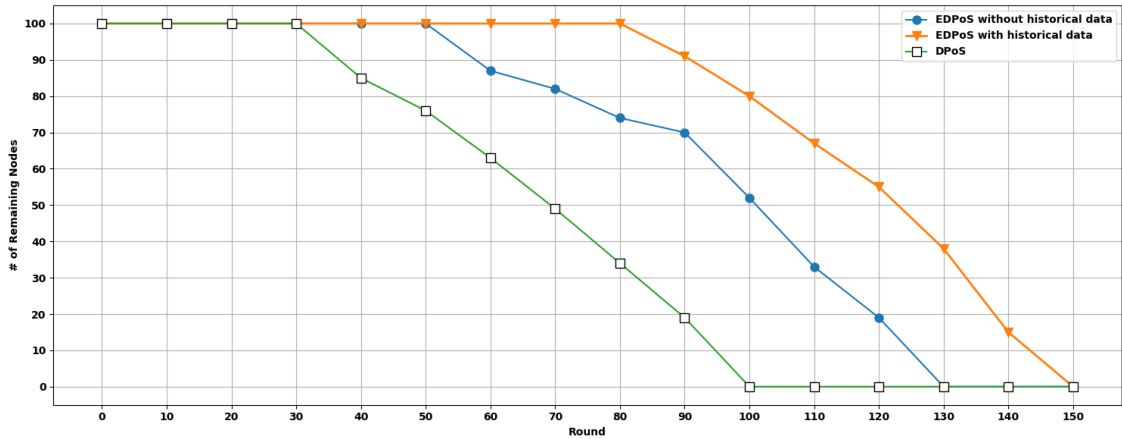


Figure 4.4: *System Longevity for EDPoS Tests*

DPoS scheme has the shortest system lifetime, with all nodes collapsing before the 100th round as its static delegate selection mechanism, which does not consider energy levels, leads to unbalanced load and rapid drainage of energy. EDPoS without historical data extends longevity to around the 130th round by including real-time energy-aware delegate election, which distributes tasks more evenly. Nonetheless, the absence of adaptive

block sizing and historical usage trends results in steady performance decline. EDPoS with historical data achieves the highest longevity, maintaining active nodes well beyond the 140th round. Its dynamic delegate voting, adaptive delegate count, and trend-based scoring enable more balanced energy usage and continuous replacement of low-energy nodes, which makes the system sustainable and the network as a whole more stable.

4.7.2 Adaptive Energy Threshold

Figure 4.5 demonstrates varying energy retention capabilities among the models. DPoS has the sharpest slope in average remaining energy, which drops to approximately 45% by the 150th round. Such fast depletion is due to the static mechanism of delegate selection, resulting in uneven distribution of energy and overloading of particular nodes, hence faster exhaustion. This EDPoS without historical data shows an even more smooth decrease, maintaining around 86% average remaining energy at the end of the simulation due to its energy-aware selection strategy, which distributes the workload more evenly among nodes based on their energy levels. However, its limited adaptability to dynamic

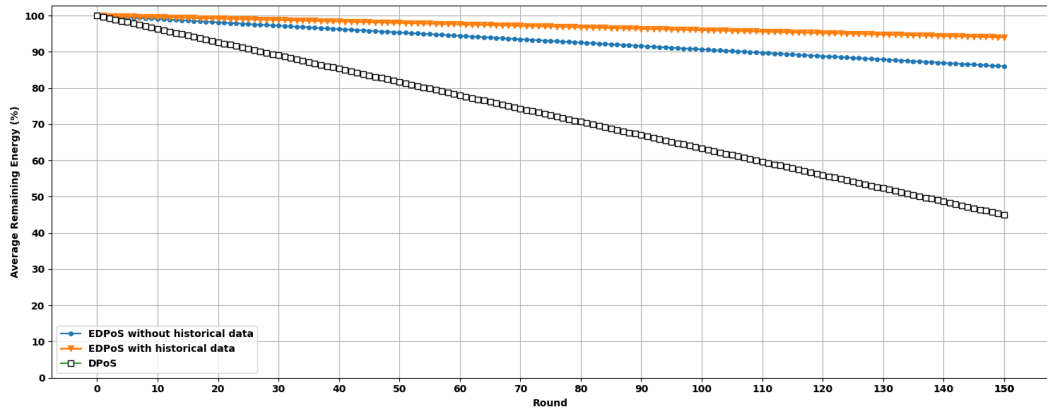


Figure 4.5: Average Remaining Energy (%) Over Rounds for EDPoS Tests

energy conditions results in a noticeable decline. In contrast, EDPoS with historical data outperforms both models, sustaining an impressive 94% average remaining energy by the

150th round through its dynamic adjustment mechanism that considers both real-time energy levels and historical consumption trends. This dual approach optimizes selection of delegates and workload distribution, effectively minimizing energy depletion, promoting longer node lifespans, and enhancing network sustainability. Furthermore, the adaptive energy threshold dynamically excludes low-energy nodes from delegate selection, ensuring sustained energy efficiency and preventing premature node failures.

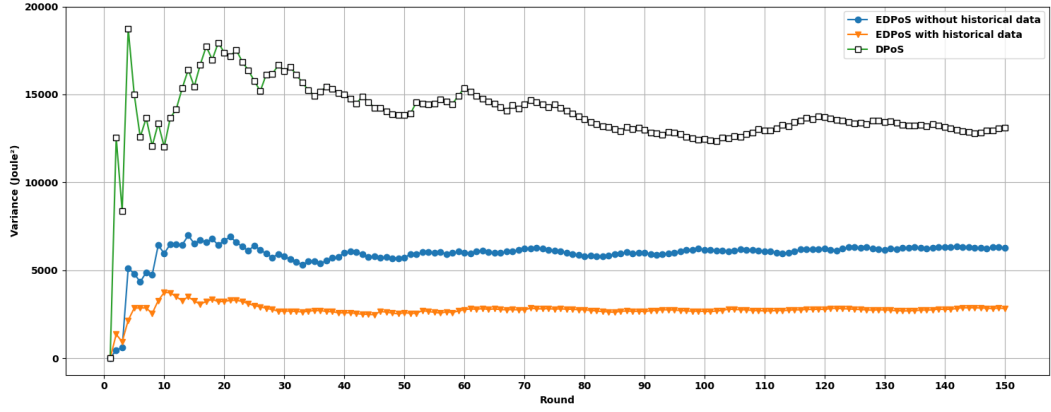


Figure 4.6: Variance of Energy Consumption per Round for EDPoS Tests

Figure 4.6 reveals distinct patterns in energy consumption variance among the models.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (E_i - \mu)^2 \quad (4.12)$$

Variance in energy consumption is mathematically defined as Equation (4.12). Where, E_i is the energy consumed by node i . μ is the mean energy consumption across all nodes and N is the total number of nodes. When energy consumption is uniformly distributed, the deviation $(E_i - \mu)$ is small across nodes, leading to a low variance. This indicates that the network effectively balances energy usage, ensuring consistent resource utilization. When energy consumption is inconsistent, some nodes consume significantly more energy than others, increasing $(E_i - \mu)^2$ for those nodes, leading to a higher variance. This reveals that some nodes are overburdened, causing imbalance in resource distribution, reducing system efficiency.

Stability in energy consumption means that energy usage remains predictable over time, avoiding large fluctuations. Fairness in workload distribution implies that all nodes share the burden of processing and validating transactions. Variance measures how widely energy consumption deviates from the average across different nodes, directly reflecting the balance and efficiency of workload distribution in the network. Out of the models analyzed, DPoS has the largest variance, fluctuating around 15,000 *Joules*², primarily due to its static delegate selection that leads to uneven task allocation. EDPoS without historical data significantly reduces this variance, stabilizing around 5,500 *Joules*² because it involves energy-efficient delegate filtering, albeit with variability. EDPoS with historical data possesses the lowest and most stable variance, consistently maintaining values near 3,600 *Joules*², thanks to its dynamic delegate adjustment and historical trend analysis, which together ensure more stable and consistent energy usage throughout the network.

4.7.3 Adaptive Block Size Adjustment

Figure 4.7 illustrates significant throughput differences among the models. ED-

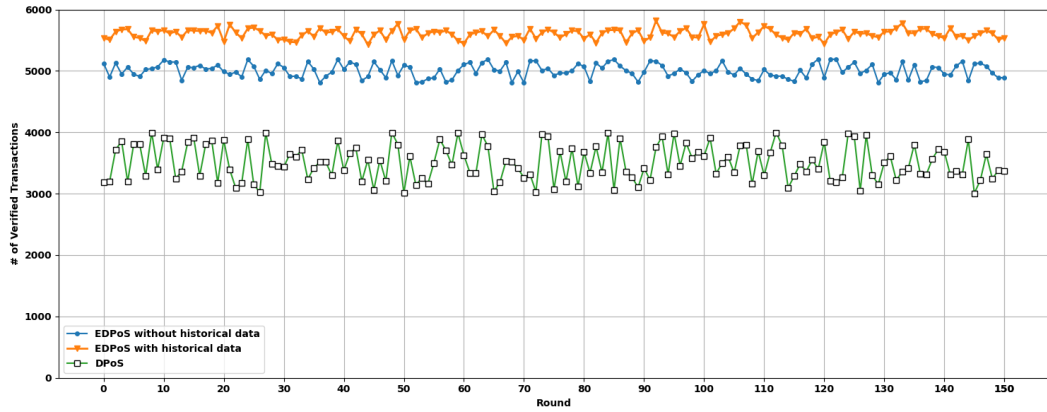


Figure 4.7: Average Throughput per Round for EDPoS Tests

PoS with historical data has consistently achieves the highest throughput, averaging approximately 5,600 verified transactions per round, due to its dynamic delegate selection optimizing resource allocation based on current energy levels and historical trends. It

is thus that EDPoS without historical data has a moderate performance of around 5,000 transactions per round, since it takes full advantage of the energy-aware delegate selection but is not adaptive to fully optimize under fluctuating conditions. On the other hand, DPoS exhibits the lowest throughput, at about 3,500 transactions per round, with noticeable variability due to its static delegate selection, which results in uneven workload distribution and lowered efficiency. Its adaptive mechanisms ensure consistent performance, making it the most suitable model for environments requiring high processing capacity and stability contexts. Its combined use of adaptive mechanisms, dynamic delegate selection and energy-aware block size adjustment, ensures consistent performance, making EDPoS without historical data and EDPoS with historical data the most appropriate model for high-throughput and energy-efficient blockchain environments.

The Figure 4.8 reveals significant differences in how each model handles transaction capacity over time. EDPoS with historical data achieves the highest transaction

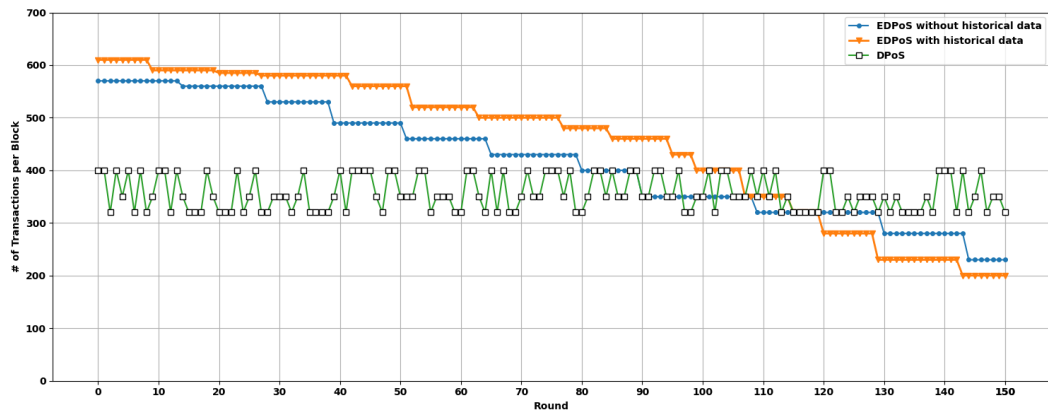


Figure 4.8: *Number of Transactions per Block for EDPoS Tests*

capacity per block, particularly during the early simulation rounds, at a maximum of 610 transactions. This is enabled by its dynamic block size adjustment algorithm, which varies block capacity adaptively based on both real-time energy fluctuation and past consumption patterns. As the simulation progresses, block sizes are gradually reduced in order to conserve energy while maintaining satisfactory throughput. EDPoS without historical

data follows a similar declining pattern but starts from a lower peak of approximately 580 transactions. Due to its reliance on static, real-time energy awareness without historical adaptation, its block size modulation is less responsive to system dynamics. On the other hand, DPoS exhibits a nearly fixed transaction rate per block, oscillating between 300 and 400 transactions, constrained by its static delegate set and fixed block size, which hinder its ability to respond to changing network conditions. Overall, EDPoS with historical data performs better than both alternatives in terms of transaction scalability throughout, especially at the initial period. These findings emphasize the value of incorporating energy-adaptive and trend-aware mechanisms for optimizing throughput in resource-constrained decentralized networks.

Block validation failure rate measures the percentage of blocks that fail to be validated due to various factors such as depleted energy levels or node inefficiencies. It can be mathematically expressed as shown in Equation (4.13).

$$\text{Failure Rate}(F) = \frac{B_f}{B_p} \times 100 \quad (4.13)$$

B_f refers to the number of blocks that were proposed but could not be successfully validated, while B_p is the total number of proposed blocks. Figure 4.9 reflects a critical distinction between node depletion and actual system failure. While the Figure 4.4, which implies that nodes remain in the system until round 100, the failure rate graph now accurately captures the fact that block validation begins to decline at round 80. This discrepancy arises because, while there are still nodes active, they do not have enough energy levels to operate the validation of new blocks.

Figure 4.9 illustrates the effective failure point of the blockchain system, defined as the round after which the network is no longer maintain operational block verification despite the presence of active nodes. In the DPoS system, failures start occurring as early as round 50, corresponding to the rapid depletion of delegate energy due to its static election and fixed block size. EDPoS without historical data exhibits improved resilience,

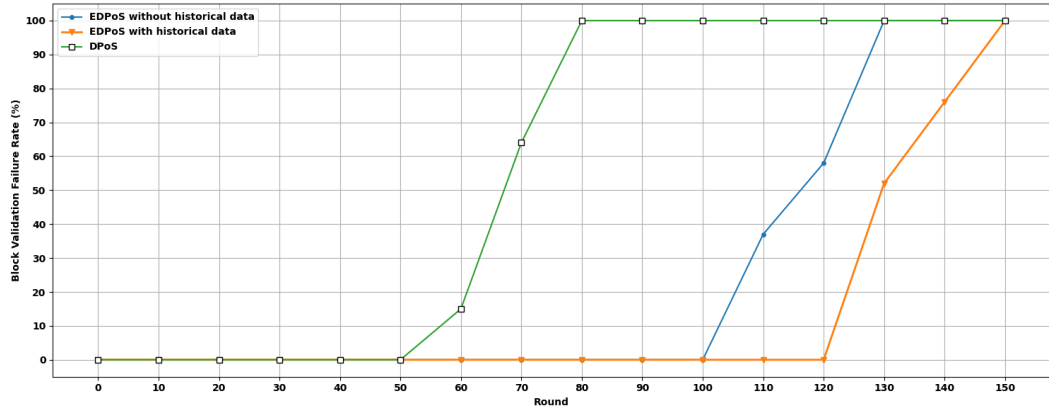


Figure 4.9: *Block Validation Failure Rate for EDPoS Tests*

with the failure events beginning around round 100, and some nodes remaining active. EDPoS with historical data further postpones this threshold to round 120, benefiting from its trend-based delegate voting and energy-adaptive behavior. Most importantly, both EDPoS versions avoid abrupt failure by adaptively reducing block sizes as energy continues to decline, allowing the system to continue processing reduced transaction loads under constrained conditions. In contrast, DPoS’s fixed block structure prevents such adaptation, resulting in premature system breakdown. Results show the implications of adaptive algorithms in extending the functionality of a system and deferring network collapse under energy constraints.

5. TEDPOS: TRUST AND ENERGY AWARE DELEGATED PROOF OF STAKE

Blockchain has become a key technology for facilitating decentralized, secure, and tamper-proof data exchange in distributed systems. Among blockchain consensus protocols, Delegated Proof-of-Stake (DPoS) [14, 15] offers scalability and fast block confirmation by selecting delegates, to validate transactions on behalf of the network. However, the majority of existing DPoS relies primarily on token-based voting or static reputation scores, overlooking the dynamic and heterogeneous nature of real-world nodes. This often leads to the selection of unreliable or energy-depleted delegates, which endangering both security and continuity of operations, especially in resource-constrained environment.

TEDPoS framework is introduced as an advanced consensus mechanism tailored to overcome critical limitations observed in traditional Delegated Proof-of-Stake (DPoS) systems. While DPoS achieves scalability and accelerated block finalization, it is susceptible to not having mechanisms for ensuring the behavioral dependability and operational longevities of consensus-participating nodes. TEDPoS addresses this gap by integrating trust awareness and energy efficiency directly into the delegate election and voting process. Unlike conventional methods that statically relying on token ownership or reputation, TEDPoS ranks nodes based on trust values, energy levels and votes. The model offers a strong, fair, and self-tuning scheme suitable for IoT, UAV, and other resource-constrained decentralized systems. The operational flow of TEDPoS is composed of six sequential phases: service provision, candidacy eligibility evaluation, candidate assignment, community voting, delegate finalization, and block validation. Each phase includes both local observation of behavior and system-wide consensus mechanisms, ensuring that only energy-capable and trustworthy nodes can participate in block creation. Through

temporal trust weighting integration, energy-based trust scaling, and behavior classification via clustering, TEDPoS achieves adaptability against evolving malicious behaviors such as camouflage or bad-mouthing. Additionally, its adaptive delegate scoring mechanism adapts the contribution of trust, energy, and votes based on the network's current energy status, ensuring performance sustainability and attack resistance. This energy-adaptive logic not only improves fault tolerance and long-term availability but also maintains fairness and decentralization, a balance that we quantify using our proposed Trust-Energy Composite Score (TECS). The key contributions of TEDPoS model can be summarized as follows:

- We present TEDPoS, a novel DPoS consensus framework that simultaneously integrates trust evaluation and real-time energy awareness. Unlike DPoS models that rely solely on token ownership or static reputation, our solution enables security, and sustainable delegate selection.
- We develop a trust computation model based on service quality metrics, specifically data integrity and timeliness. By incorporating temporal weighting and energy-based scaling, the model emphasizes recent and stable interactions. As a result, TEDPoS defends against trust manipulation strategies such as camouflage, bad-mouthing, and self-promotion attacks without relying on fixed rules or external monitoring.
- We introduce an adaptive multi-factor delegate scoring function that combines normalized trust scores, energy levels, and received votes. This scoring mechanism ensures that the relative importance of trust, energy, and voting dynamically shifts based on the network's operational conditions. We further introduce Trust-Energy Composite Score (TECS), a unified metric that enables fair comparison across models by jointly quantifying security and sustainability.

- We propose a clustering-based node behavior classification framework using one-dimensional k-means to group nodes into reliable, suspicious, or unreliable groups. The data-driven classification informs candidacy eligibility and system-level responses to misbehavior, including adaptive delegate set adjustment and malicious node removal. The approach eliminates reliance on fixed thresholds and enables real-time behavioral profiling in accordance with the network’s evolving trust distribution.

5.1 System Model: TEDPoS Architecture

In this section, we present the overall architecture and operational logic of our proposed TEDPoS model. The system integrates trust evaluation, energy awareness, and vote-based consensus in a fully decentralized environment. We begin by defining the network assumptions and key entities involved in the blockchain system. Then, we introduce the mechanisms for computing trust scores, detecting malicious behavior, and selecting delegates based on a dynamic combination of trust, energy, and voting. Furthermore, the system’s self-regulatory mechanisms are elaborated, including how it adapts to energy constraints and filters out unreliable nodes. The resilience of the model against adversarial strategies is also discussed to demonstrate its robustness and long-term stability. The proposed TEDPoS model is specifically designed to operate in energy-constrained environments while preserving trust and with scalable and secure consensus. This model is the enhanced version of Energy-Aware Delegated Proof of Stake (EDPoS).

5.1.1 Network Assumptions and Architecture

We consider a decentralized and permissionless blockchain network composed of a large number of heterogeneous nodes, such as IoT devices, edge servers, and low-power sensors. These nodes interact in a trustless environment, where each participant may be

one or more of the following roles: service provider, voter, candidate, or delegate.

In each round, nodes generate and exchange data, and each receiving node evaluates the quality of service based on two key attributes: integrity and punctuality. Based on this evaluation, service receivers assign a Trust Score (TS) upon service providers. All the nodes are assumed to be equipped with a battery or energy-restricted power source. Therefore, energy consumption plays a critical role in system sustainability. Unlike traditional DPoS, in TEDPoS, the energy levels of nodes are continuously monitored and integrated into decision-making mechanisms such as trust calculation and delegate election. No central control exists in the system. All trust assessment, score renewal, and delegate selection are all performed in a distributed and locally computable manner. The nodes maintain their own local trust tables and energy statistics and periodically broadcast them to their neighbors using lightweight status messages. The consensus process leverages a trust-energy-vote hybrid scoring scheme to ensure that only credible, available, and community-supported nodes participate in block validation. System's operational workflow can be detailed as follows, which also shown in Figure 5.1:

- ① **Service Provision:** In the initial phase, nodes exchange data and provide services to one another. After receiving a service, the recipient node evaluates the provider based on two key metrics: integrity and punctuality. This evaluation is translated into a numerical trust score, which is used in future assessments. Nodes that consistently fail to deliver valid or timely data (e.g., Node N6) may be flagged for removal and unable to be candidate for delegation.
- ② **Candidacy Eligibility Check:** Only nodes with sufficient trust and energy levels are eligible to become candidates. During this phase, each node evaluates its own historical trust score and current energy capacity. Nodes below the minimum trust or energy thresholds (e.g., Node N2) are disqualified from entering the delegation pool.

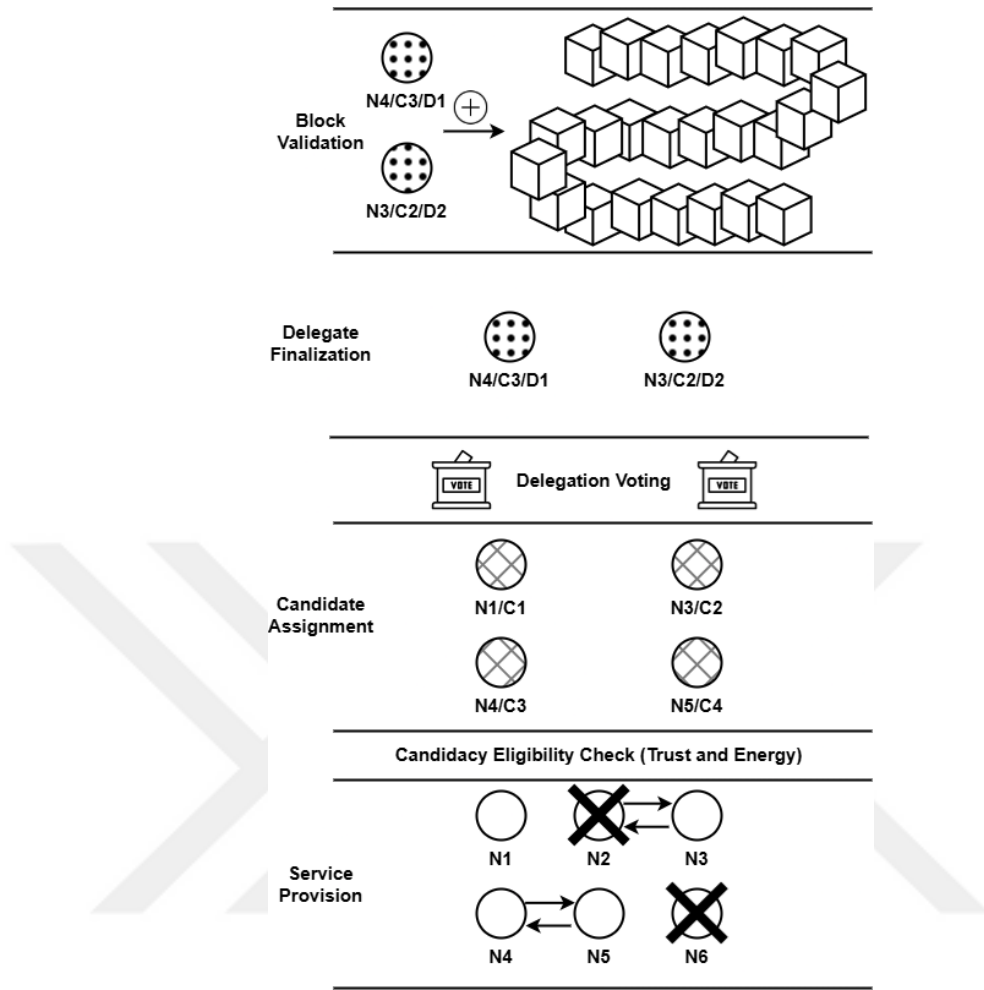


Figure 5.1: *TEDPoS Workflow.*

The eligibility check ensures that only reliable and sustainable nodes are allowed to progress to the voting phase.

- ③ **Candidate Assignment:** Eligible nodes that pass the Phase 2 criteria are formally recognized as candidates and assigned candidate IDs (e.g., $N4$ becomes $C3$). These nodes are now part of the candidate pool and will participate in the upcoming vote-based selection process. At this point, the candidate status does not imply authority but reflects potential for selection.
- ④ **Delegation Voting:** In this phase, other nodes in the network (voters) cast their votes

for the candidate nodes. The voting process is distributed and based on each voter's perception of a candidate's historical trust and availability. The number of votes received by each candidate will later be integrated with trust and energy scores to compute the final delegate selection score.

- ⑤ **Delegate Finalization:** Nodes with the highest composite scores, combining trust, energy, and received votes, are selected as delegates. The delegate set (e.g., N3/C2/D2 and N4/C3/D1) is finalized at this phase. These nodes are now authorized to participate in block validation and consensus for the current round.
- ⑥ **Block Validation:** Finally, selected delegates engage in block validation and append validated blocks to the blockchain. This phase represents the consensus process in action. Once the round ends, the system resets and enters the next cycle starting again from Phase 1, continuously updating trust and energy metrics for adaptivity and resilience.

This workflow captures the step-by-step operation of TEDPoS model, presenting how trust, energy, and community voting jointly govern delegate eligibility and consensus.

5.1.2 Trust and Energy Computation Framework

In decentralized environments where nodes interact without any central authority, trust becomes a fundamental factor for ensuring secure and reliable collaboration. In the proposed TEDPoS framework, each node in the network can act as both a service provider and a service receiver, participating in data exchange processes that require mutual assessment. Since the network is open and potentially includes malicious participants, nodes must dynamically evaluate the behavior of their peers to identify trustworthy and untrustworthy actors. Delegation voting denotes the amount of support a node receives from other stakeholders as proposed in [15], usually through token-based voting

mechanisms. It reflects the perceived reputation or importance of a node, directly influencing its chances of being selected as a delegate in DPoS-based systems. In this model, we optimize the conventional DPoS mechanism by integrating three critical factors into the delegation process: trust and energy.

- **Trust:** In blockchain and DPoS systems, trust is the confidence the network has in a node's honesty, consistency, and reliable behavior. It is typically derived from historical interactions. Trust is defined in [64], is a mental process of expectation, interpretation, and suspension, where individuals transition from partial knowledge to favorable expectation through a "leap of faith" beyond rational reasoning.
- **Energy:** It typically represents the computational or battery resources a node has available, especially in resource-constrained environments like IoT-integrated systems. Nodes with higher energy are more likely to maintain consistent availability and fulfill their roles in the consensus process without interruption.

5.1.2.1 Quantitative Trust Feedback Derivation

Let $\mathcal{N} = \{n_1, n_2, \dots, n_k\}$ denote the set of all participating nodes in the TEDPoS network. Nodes interact by providing and consuming services such as data transmission, sensing results, or computational tasks. For every service transaction between a provider node n_i and a receiver node n_j , a trust feedback score is generated by n_j and assigned to n_i based on two quantifiable metrics: data integrity and delivery punctuality.

- **Integrity Score Function:** The integrity score $I(n_i, t) \in [0, 1]$ represents the correctness of the data provided by node n_i at interaction time t . It is formally defined in Equation (5.1) as:

$$I(n_i, t) = \frac{\delta_{\text{valid}}(n_i, t)}{\delta_{\text{total}}(n_i, t)} \quad (5.1)$$

Where:

- $\delta_{\text{valid}}(n_i, t)$ = Number of correctly delivered and verified data units by node n_i at time t ,
- $\delta_{\text{total}}(n_i, t)$ = Total number of data units received from node n_i at time t .

If $\delta_{\text{total}} = 0$, then $I(n_i, t)$ is undefined and excluded from computation.

- **Punctuality Score Function:** The punctuality score $P(n_i, t) \in [0, 1]$ measures the delivery timeliness of the data. In Equation (5.2) it is defined as:

$$P(n_i, t) = \frac{\tau_{\text{on-time}}(n_i, t)}{\tau_{\text{expected}}(n_i, t)} \quad (5.2)$$

Where:

- $\tau_{\text{on-time}}(n_i, t)$ = Number of data units delivered within the expected time window,
- $\tau_{\text{expected}}(n_i, t)$ = Total number of expected time-sensitive data units.

If no time-sensitive packets are expected, $P(n_i, t)$ is excluded from scoring.

- **Composite Trust Feedback Score:** After both metrics are evaluated, the receiver node n_j calculates a composite trust feedback score $P_{\text{new}}(n_i, t)$ for the provider node n_i in Equation (5.3) as:

$$P_{\text{new}}(n_i, t) = w_I \cdot I(n_i, t) + w_P \cdot P(n_i, t) \quad (5.3)$$

Where:

- $w_I, w_P \in [0, 1]$ are the weight factors assigned to integrity and punctuality, respectively,
- $w_I + w_P = 1$

In our implementation as shown in Equation (5.4), unless otherwise stated, we assume equal contribution from both metrics:

$$w_I = w_P = 0.5 \quad (5.4)$$

5.1.2.2 Trust Score Calculation Process

The trust score of each node is continuously updated over time as new feedback comes. However, in dynamic and potentially malicious environment, assigning equal weight to all past and present feedback may lead to delayed or inaccurate responses to behavioral changes. Trust feedback serves as the instantaneous assessment of node behavior, whereas the trust score is the result of a continuous integration of such assessments over time. To mitigate this, TEDPoS integrates a temporal weighting mechanism that prioritizes recent trust observations while gradually diminishing the influence of older scores. Let:

- $T_\theta^{(t)} \in [0, 1]$: The current trust score of node n_i at time step t ,
- $P_{\text{new}}(n_i, t) \in [0, 1]$: The latest trust feedback computed,
- $E_{\text{factor}}(n_i, t) \in \mathbb{R}^+$: The energy-aware scaling coefficient,
- $\phi \in \mathbb{R}^+$: A temporal weighting function.

The trust score is updated according to the following rule as shown in Equation (5.5):

$$T_\theta^{(t+1)} = \frac{T_\theta^{(t)} + \phi \cdot P_{\text{new}}(n_i, t) \cdot E_{\text{factor}}(n_i, t)}{1 + \phi} \quad (5.5)$$

This formulation enables the new feedback to influence the current trust value more significantly than older feedback as t increases. We define ϕ as temporal weighting function in Equation (5.6):

$$\phi = \frac{1}{\sqrt{t+1}} \quad (5.6)$$

Temporal weighting function is based on inverse-root structure. Which is a well-established method in adaptive control systems, online filtering, and learning environments to dynamically prioritize recent observations while avoiding abrupt shifts in system state [65–67]. Unlike fixed-weight or exponentially weighted schemes, the temporal weighting function ensures that:

- Early interactions (e.g., during a node’s initial participation) still retain noticeable influence, promoting gradual trust accumulation rather than sudden spikes.
- Later updates gain more weight incrementally, allowing the trust model to adapt quickly to changes in behavior while suppressing overreactions to outliers or one-time anomalies.

Compared to exponential decay, which forgets past too aggressively, and linear decay, which adapts too slowly, the temporal weighting function enables a smoother transition, giving more weight to recent feedback while preserving early observations just enough to detect behavioral shifts over time.

5.1.2.3 Energy Scaling Factor

While behavioral feedback is critical for evaluating trust, it is equally important to consider a node’s sustainability in resource-constrained environments. In TEDPoS, each node is assumed to operate on a limited energy source (e.g., battery-powered IoT devices). Nodes with insufficient energy may become unstable, unreliable, or may drop out during critical phases like block validation. Therefore, trust feedback is adjusted dynamically based on the current energy level of the evaluated node.

To integrate this dimension, we define an energy scaling coefficient $E_{\text{factor}}(n_i, t)$, which adjusts the influence of feedback on the trust score update. Nodes with relatively high energy levels are slightly rewarded, while nodes with below-average energy have diminished impact on their trust growth. Let:

- $E(n_i, t) \in [0, E_{\max}]$: Current energy level of node n_i at time t ,
- $E_{\text{avg}}(t)$: Average energy level across all active nodes at time t ,
- $\alpha \in \mathbb{R}^+$: Sensitivity coefficient.

Then the energy scaling coefficient is defined as Equation (5.7):

$$E_{\text{factor}}(n_i, t) = 1 + \alpha \cdot \left(\frac{E(n_i, t)}{E_{\max}} - \frac{E_{\text{avg}}(t)}{E_{\max}} \right) \quad (5.7)$$

Interpretation:

- When $E(n_i, t) > E_{\text{avg}}$,
 $E_{\text{factor}} > 1 \rightarrow$ trust gain is amplified.
- When $E(n_i, t) < E_{\text{avg}}$,
 $E_{\text{factor}} < 1 \rightarrow$ trust gain is reduced.
- When $E(n_i, t) = E_{\text{avg}}$,
 $E_{\text{factor}} = 1 \rightarrow$ no change applied.

This mechanism introduces a lightweight form of energy-aware fairness into the trust update process without compromising the behavioral integrity of evaluations. In practice, setting $\alpha \in [0.1, 0.5]$ provides an optimal balance between behavioral feedback and energy contribution.

5.1.3 Node Evaluation and Behavior Profiling

Nodes are continuously evaluated based on their trust scores, and accordingly classified into three behavioral categories: reliable, suspicious, and unreliable. This classification directly influences node roles in the system, including candidacy permission, behavioral monitoring, and removal.

Rather than using fixed thresholds, we adopt a clustering-based trust thresholding method to ensure the classification reflects the real-time distribution of trust values in the network. Let:

- $\mathcal{T}(t) = \{T_1^{(t)}, T_2^{(t)}, \dots, T_N^{(t)}\}$ denote the trust scores of all N nodes in the network at evaluation round t ,
- $k = 3$ be the number of trust behavior categories we aim to discover: low, medium, and high trust.

To extract these behavioral groups, we apply one-dimensional k-means clustering [68] to the dataset $\mathcal{T}(t)$. K-means is a well-established unsupervised learning algorithm used to minimize intra-cluster variance by partitioning data into k clusters based on similarity. In our case, the one-dimensional version operates over trust scores and divides the dataset into three disjoint clusters:

$$\mathcal{T}_{\text{low}}, \mathcal{T}_{\text{mid}}, \mathcal{T}_{\text{high}} \subset \mathcal{T}(t), \quad \text{with} \quad \mathcal{T}_{\text{low}} \cup \mathcal{T}_{\text{mid}} \cup \mathcal{T}_{\text{high}} = \mathcal{T}(t)$$

The clustering minimizes in Equation (5.8) with the cluster variance:

$$\min_{C_1, C_2, C_3} \sum_{i=1}^3 \sum_{T_j \in \mathcal{T}_i} (T_j - C_i)^2 \quad (5.8)$$

where:

- C_1, C_2, C_3 are the centroids of the clusters $\mathcal{T}_{\text{low}}, \mathcal{T}_{\text{mid}}, \mathcal{T}_{\text{high}}$, respectively.

For notation consistency, let:

- $C_{\text{low}} = \mu(\mathcal{T}_{\text{low}})$
- $C_{\text{mid}} = \mu(\mathcal{T}_{\text{mid}})$
- $C_{\text{high}} = \mu(\mathcal{T}_{\text{high}})$

Once the clusters are formed and centroids calculated, the trust classification thresholds are computed in (5.9) as the midpoints between adjacent cluster centers:

$$\theta_{\text{low}} = \frac{C_{\text{low}} + C_{\text{mid}}}{2}, \quad \theta_{\text{high}} = \frac{C_{\text{mid}} + C_{\text{high}}}{2} \quad (5.9)$$

This classification approach ensures the following:

- Nodes belonging to the low-trust cluster ($\mathcal{T}_{\text{low}}$) are labeled as unreliable, satisfying the condition $T_{\theta}^{(t)} < \theta_{\text{low}}$ which will be discarded.
- Nodes assigned to the high-trust cluster ($\mathcal{T}_{\text{high}}$) are deemed reliable, such that $T_{\theta}^{(t)} \geq \theta_{\text{high}}$.
- Nodes with trust scores in the intermediate range ($\theta_{\text{low}} \leq T_{\theta}^{(t)} < \theta_{\text{high}}$) are classified as suspicious.

This dynamic approach adapts to trust score distribution in real time, allowing classification thresholds to naturally reflect the statistical and behavioral characteristics of the network.

5.1.4 Delegate Selection Process in TEDPoS

To ensure fair and efficient consensus, TEDPoS employs a structured delegate selection mechanism that integrates behavioral trust, energy sustainability, and community approval. This subsection outlines the mathematical and procedural foundation of candidate eligibility, voting dynamics, and delegate assignment.

5.1.4.1 Candidate Eligibility Criteria

Only nodes classified as reliable or suspicious may be considered for candidacy. Let $T_{\theta}^{(t)}$ be the trust score of node n_i at evaluation time t , and $\theta_{\text{low}}, \theta_{\text{high}}$ be the dynamic classification thresholds. A node n_i is marked eligible in Equation (5.10) for candidacy if

the following condition holds:

$$T_{\theta}^{(t)} \geq \theta_{\text{low}} \quad \text{and} \quad E(n_i, t) \geq E_{\text{min}} \quad (5.10)$$

Where:

- $E(n_i, t)$ is the node's current energy level,
- E_{min} is the minimum energy threshold required for candidacy.

Reliable nodes ($T_{\theta} \geq \theta_{\text{high}}$) are prioritized. Suspicious nodes ($\theta_{\text{low}} \leq T_{\theta} < \theta_{\text{high}}$) are admitted as candidates with limited voter confidence and flagged for increased monitoring during delegation.

5.1.4.2 Voting Procedure and Delegate Scoring

All non-candidate nodes in the network act as voters. Each voter selects a subset of eligible candidates to support. To prevent manipulation and ensure fairness, TEDPoS integrates three dimensions into delegate selection:

- **Trust Score T** : Long-term behavioral reliability.
- **Energy Level E** : Operational sustainability.
- **Vote Count V** : Collective endorsement by the network.

In delegate scoring function let:

- $V(n_i, t)$: number of valid votes received by candidate n_i ,
- $V_{\text{max}}(t)$: highest vote count among all candidates,
- $T(n_i, t), E(n_i, t)$: current trust score and energy level,
- $T_{\text{max}}, E_{\text{max}}$: normalization constants (e.g., maximum observed values).

We define the delegate selection score ($S_D(n_i, t)$) in Equation (5.11) for candidate n_i as:

$$S_D(n_i, t) = \left(\frac{T(n_i, t)}{T_{\max}} \cdot \frac{E(n_i, t)}{E_{\max}} \right)^{\beta(t)} \cdot \left(\frac{V(n_i, t)}{V_{\max}(t)} \right)^{1-\beta(t)} \quad (5.11)$$

Where $\beta(t) \in [0, 1]$ is an adaptation coefficient reflecting the current energy state of the system. It is dynamically computed in Equation (5.12) as:

$$\beta(t) = \frac{E_{\text{avg}}(t)}{E_{\max}} \quad (5.12)$$

with $E_{\text{avg}}(t)$ being the average energy level across all candidates. Interpretation:

- When system energy is low, $\beta(t)$ increases and trust and energy dominate delegate scoring.
- When system energy is high, $\beta(t)$ decreases as vote counts carry more weight.
- This structure ensures energy-adaptive consensus: reliable nodes are promoted in resource-constraint conditions, while community preference is respected under normal operation.

5.1.4.3 Delegate Selection and Block Validation

Candidates are ranked in descending order of their $S_D(n_i, t)$ scores. The top $N_{\text{delegates}}$ nodes are selected to form the delegate set for the next consensus round. Delegate selection function defined in (5.13):

$$\mathcal{D}(t) = \text{Top-}N_{\text{delegates}}(\{S_D(n_i, t)\}_{n_i \in \text{Candidates}}) \quad (5.13)$$

Selected delegates participate in block validation and creation. The block production process is organized in a round-robin scheme among selected delegates. In case of tie-breaking, trust score is used as a secondary criterion to determine the order of block proposers. Each delegate is responsible for:

Algorithm 4 TEDPoS Delegate Selection Algorithm

Input: Node set $\mathcal{N}$, trust scores $T_s^{(t)}$, energy levels $E(n_i, t)$, vote data $V(n_i, t)$

Output: Delegate set $\mathcal{D}(t)$

```
1: CandidateSet  $\leftarrow \emptyset$ 
2: for each node  $n_i \in \mathcal{N}$  do
3:   if  $T_s^{(t)}(n_i) \geq \theta_{\text{low}}$  and  $E(n_i, t) \geq E_{\text{min}}$  then
4:     CandidateSet  $\leftarrow$  CandidateSet  $\cup \{n_i\}$ 
5:   end if
6: end for
7: VoterSet  $\leftarrow \mathcal{N} \setminus \text{CandidateSet}$ 
8: for each voter  $v \in \text{VoterSet}$  do
9:   allocate votes to eligible candidates in CandidateSet
10: end for
11: Compute  $T_{\text{max}}, E_{\text{max}}, V_{\text{max}}(t)$  from CandidateSet
12:  $E_{\text{avg}}(t) \leftarrow \frac{1}{|\text{CandidateSet}|} \sum_{n_i \in \text{CandidateSet}} E(n_i, t)$ 
13: Compute  $\beta(t) = \frac{E_{\text{avg}}(t)}{E_{\text{max}}}$ 
14: for each  $n_i \in \text{CandidateSet}$  do
15:   Compute delegate score:
```

$$S_D(n_i, t) = \left(\frac{T_s(n_i, t)}{T_{\text{max}}} \cdot \frac{E(n_i, t)}{E_{\text{max}}} \right)^{\beta(t)} \cdot \left(\frac{V(n_i, t)}{V_{\text{max}}(t)} \right)^{1-\beta(t)}$$

```
16: end for
17: Sort CandidateSet by  $S_D(n_i, t)$  in descending order
18: DelegateSet  $\mathcal{D}(t) \leftarrow$  top  $N_{\text{delegates}}$  nodes from CandidateSet
19: return  $\mathcal{D}(t)$ 
```

- Validating incoming transactions,
- Packaging them into blocks,
- Broadcasting the blocks to the network.

The trust and energy metrics of delegates are continuously monitored; nodes falling below defined operational thresholds are demoted and replaced dynamically.

Algorithm 4 outlines the delegate selection process in the TEDPoS model by integrating trust, energy, and vote factors into a unified scoring mechanism. Nodes that meet the candidacy criteria based on dynamic trust and energy thresholds are evaluated through a voting phase. The algorithm computes a delegate score $S_D(n_i, t)$ for each candidate,

adaptively balancing behavioral reliability, energy sustainability, and network approval via a dynamic weighting coefficient $\beta(t)$. The top-ranked nodes are then selected to form the delegate set for the upcoming consensus round.

5.2 Simulation Setup

We evaluated the performance and robustness of the proposed TEDPoS model through extensive simulations conducted in OMNeT++, a modular C++, based discrete event simulator widely used for network protocol evaluation [60]. The simulation environment (See Table 5.1) simulates a heterogeneous decentralized network of nodes operating at varying levels of energy, trust conditions, and transaction loads. Nodes are assigned random spatial distribution and initialized with energy levels drawn from a uniform range to simulate real-world heterogeneity. The simulation is organized in discrete rounds, each representing a full cycle of data exchange, trust evaluation, candidate selection, voting, and block validation. To promote diversity in system behavior and multiple tests the protocol under adversarial conditions, malicious nodes are injected with predefined deceptive strategies (e.g., self-promotion, camouflage, bad-mouthing), and transaction frequencies are varied across different scenarios. Energy consumption rates are linked to the node roles and activities during consensus operations. 10 different battery options were used during simulations as follows: 1500, 2200, 3000, 4500, 5200, 6000, 7500, 10000, 12000, 15000 mAh. During the simulation runs these battery options are randomly distributed to nodes by OMNeT++ simulator itself.

5.3 Simulation Results and Evaluation

In order to evaluate the resilience of the TEDPoS framework against adversarial behavior, we experiment with three of the most widely recognized malicious approaches

Table 5.1: *Simulation Metrics*

Number of Nodes	100
Number of Rounds	150
Delegate Count for DPoS	10
Episode	10 simulation runs
Battery Voltage	3.7 V
Energy Consumption per Packet	0.1% of node capacity
Energy Conversion Factor	3.6
Battery Capacity Levels	Varies from 1500mAh to 15000mAh
Simulation Tool	OMNeT++ 6.0+
Compiler	GCC 11.3.0
Operation System	Ubuntu 22.04 (Linux Kernel 5.15+)
Framework	INET Framework 4.4

in decentralized trust systems. Each attack type is designed to exploit a different vulnerability in either the voting or trust mechanism:

- **Self-Promotion Attack [69]:** Attacker nodes always give themselves high scores irrespective of the quality of services they offer. This aims to artificially inflate their trust score and increase chances of being selected as a delegate.
- **Camouflage Attack [70]:** Malicious node strategically submits fair ratings to camouflage itself, while sometimes giving unfair ratings to targeted nodes. This attack makes it harder for trust management systems to distinguish malicious nodes from honest ones, as their rating patterns appear similar.

- **Bad-Mouthing Attack [71]:** Malicious nodes deliberately give low trust ratings to good nodes with the intention of decreasing their probability of being selected as candidates. This attack aims to manipulate the network by discouraging honest nodes and unfairly biasing the election process.

These attack models are injected into the simulation environment with adjustable ratios to evaluate the adaptive trust strategies, and robustness of TEDPoS against targeted manipulation. To validate the effectiveness of TEDPoS, we compare it with against the following baseline models:

- **DPoS [14]:** Delegated Proof-of-Stake scheme where nodes are elected solely based on received votes or token ownership, without taking into account trust or energy. The framework does not incorporate adaptive behavior profiling and is susceptible to strategic manipulation.
- **EDPoS (Energy-Aware DPoS with Historical Data) 4:** We propose EDPoS in 4, an enhanced DPoS protocol that incorporates energy-level considerations into node selection to prevent resource depletion 4. Among its two variants, we adopt the one that leverages historical data tracking, as it offers better accuracy in monitoring energy consumption and serves as a robust baseline for our evaluation. However, this variant does not account for node trustworthiness or behavioral consistency, making it vulnerable to trust-based attacks.
- **Semi-Centralized Trust Management Model [2]:** The proposed model integrates both direct and indirect trust evaluations, incorporating trust decay functions and recommendation credibility metrics to effectively detect and mitigate the impact of malicious nodes in the network efficiently. Even though this approach offers strong capabilities in behavioral trust assessment and enhances overall system resilience to malicious feedback, it does not account for the energy consumption or resource

constraints of IoT devices, which limits its feasibility in energy-constraint environments commonly encountered in large-scale or battery-powered IoT systems.

Although TEDPoS introduces slightly higher energy overhead compared to EDPoS, it significantly enhances system reliability through adaptive trust mechanisms. This subtle trade-off between energy efficiency and trustworthiness will be further explored in the simulation analysis section.

5.3.0.1 Trust and Security

The Figure A.2 is helps to to determine the system's ability to distinguish between honest nodes and malicious ones through the evolution of trust scores. In trust-based consensus algorithms, the trust score decides the suitability of a node for delegation or block verification. Therefore, observing how such scores diverge over time provides valuable insights about the resilience and learning ability of the consensus model. In Figures 5.2–5.3, to simulate a realistic adversarial environment, a malicious node ratio (MNR) of 50% was applied at the beginning of the simulation. Moreover, the attackers were programmed to alter their attack strategies every 50 rounds as follows:

- Starting with camouflage attacks in the first 50 rounds.
- Switching to self-promoting attacks in rounds 51–100.
- Finally performing bad-mouthing attacks in the last 50 rounds.

This variation in adversarial behavior further validates the system's ability to adaptively and accurately respond to different threat types over time. In Figure 5.2, Semi-Centralized [2] achieves the sharpest separation, consistently assigning trust scores around 0.9 to honest nodes while reducing malicious node scores below 0.05 within the first 30 rounds. This is expected, as it operates solely on trust-based evaluation with strict behavioral

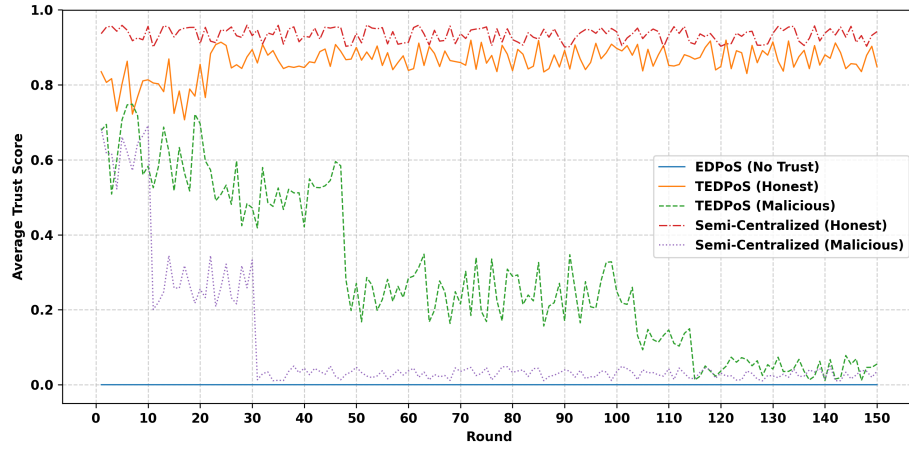


Figure 5.2: Average trust score evolution of honest and malicious nodes over time in 50% MNR environment.

thresholds. TEDPoS, while not as aggressive as the Semi-Centralized model, still maintains a reliable separation. Initially, both groups start having the similar values due to the absence of prior knowledge. However, with progressing rounds, honest nodes accrue a higher trust score, converging around 0.9, while malicious nodes experience a progressive decline, reaching below 0.2 at the end. This sharp separation strengthened by the use of inverse-time decay and reinforcement of behaviors such that TEDPoS model can distinguish between untrustworthy nodes early in the process. This reflects its hybrid and adaptiveness, where trust evolves through time-decay, behavior scoring, and energy sensitivity rather than abrupt filtering. Finally, EDPoS 4 assigns a static trust value of zero, as it lacks any trust capability. Overall, TEDPoS offers a middle ground, which is not as reactive as Semi-Centralized, but significantly more behavior-aware than EDPoS 4, achieving a balance between fairness, adaptability, and attack resilience.

The Figure 5.3 evaluates the resilience of the consensus models in resisting the election of malicious nodes as delegates over time, a major security aim in DPoS networks. Since malicious delegates can severely undermine the integrity and consistency of the blockchain, a lower Malicious Delegate Ratio (MDR) indicates a stronger trust evaluation and voting system. As shown in the figure, TEDPoS achieves excellent performance,

sustaining the lowest MDR during the entire 150 rounds. In the initial rounds, TEDPoS starts with an MDR of approximately 20%, reflecting a natural uncertainty phase due to the lack of prior trust data. However, as the simulation runs on, the system can identify

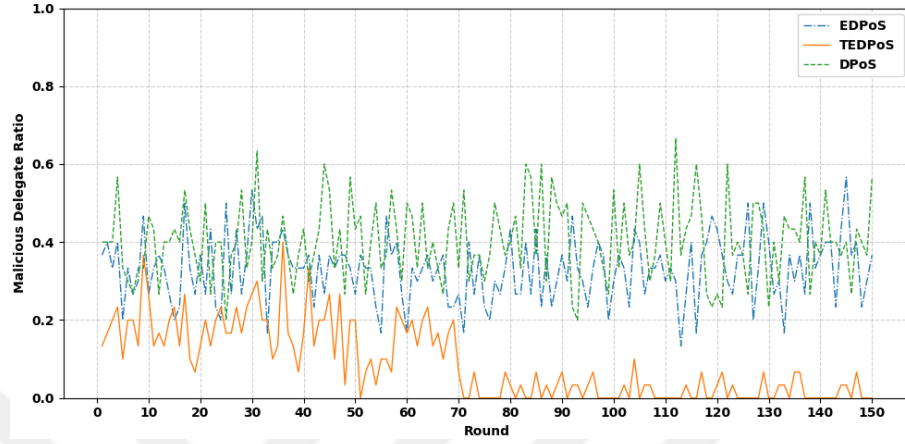


Figure 5.3: *Malicious delegate ratio across 150 rounds in 50% MNR environment.*

and reject malicious nodes, and the MDR rapidly declines, eventually converging close to 0. This is due to the fact that TEDPoS uses a dynamic trust evaluation system that combines inverse-time trust decay, energy-weighted behavior scoring, and adaptive candidate filtering to detect inconsistencies in node behavior. On the other hand, EDPoS 4 improved moderately over traditional DPoS but could not suppress malicious delegates consistently. Lacking any real-time trust inference, EDPoS's 4 MDR fluctuated between 20% and 40%, revealing its vulnerability to evolving attack strategies that manipulate historical voting records. Traditional DPoS suffered the most, with its MDR consistently ranging between 30% and 50%, due to the absence of any trust-aware mechanisms. This makes it seriously susceptible to strategic vote manipulation and collusion, more so when the adversarial presence in the network is high. Overall, TEDPoS model outperforms EDPoS 4 in terms of trust convergence accuracy and malicious delegate ratio. Its mechanism of dynamic assessment of trust is not only improving the identification of malicious players, but it is also enforcing truthfulness in conduct in the long run, causing a more resilient and secure delegating process. Semi-Centralized is excluded as it does not employ delegate-based

consensus.

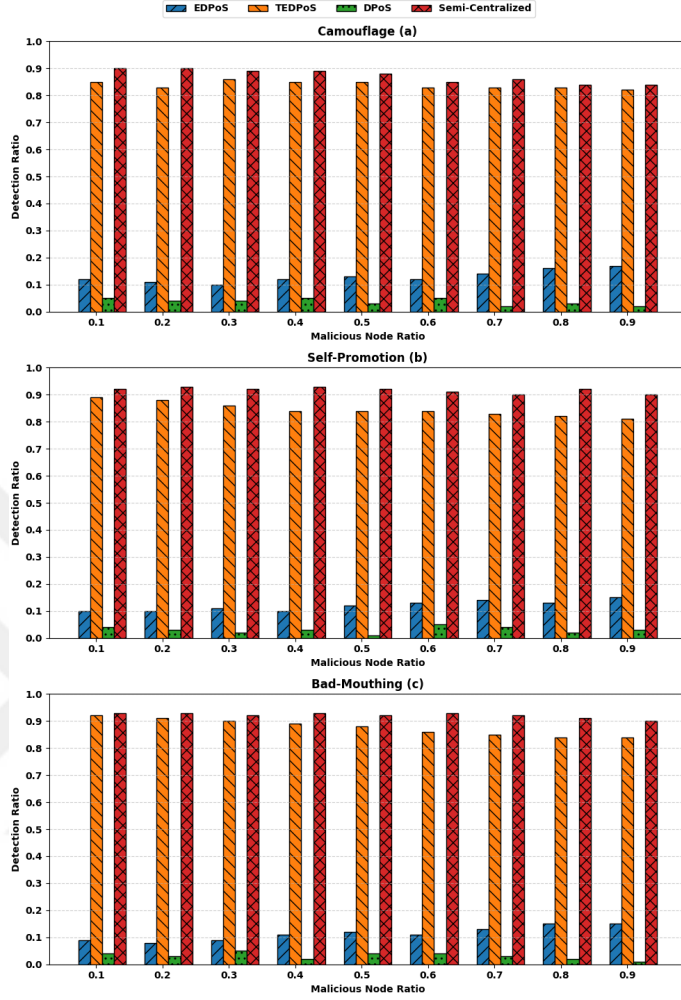


Figure 5.4: Detection rates of camouflage (50% camouflage rate), self-promotion, and bad-mouthing attacks across varying MNR, comparing TEDPoS, EDPoS 4 and DPoS.

In Figure 5.4 illustrates the ability of the consensus models to detect adversarial behavior under three various trust-based attack strategies: camouflage, self-promotion, and bad-mouthing. The simulation evaluates model performance across increasing levels of malicious node ratio (MNR), ranging from 0.1 to 0.9. Each attack scenario is assessed independently, enabling focused analysis on specific adversarial behaviors. Detection rate defined as follows:

$$\text{Detection Rate} = \frac{|\mathcal{M}_{\text{detected}}|}{|\mathcal{M}_{\text{total}}|} \quad (5.14)$$

Where, $\mathcal{M}_{\text{detected}}$ denotes the set of malicious nodes correctly identified by the system, and $\mathcal{M}_{\text{total}}$ is the total set of malicious nodes injected into the network. In all three subplots, Semi-Centralized achieves the highest detection performance, consistently maintaining detection ratios around or above 90%, regardless of the MNR level or attack type. This is expected, as it applies strict trust-based decision-making with sharp thresholds, enabling rapid and aggressive malicious node elimination. TEDPoS also performs strongly and outperforms both EDPoS 4 and DPoS by a significant margin. Even under the highest MNR, TEDPoS achieves a detection ratio above 83% for all attack types. This success stems from its hybrid trust evaluation framework that includes inverse-time decay, behavioral clustering, and trust score fluctuations to effectively identify deceptive behaviors. EDPoS 4 lacks a dedicated trust evaluation mechanism. Therefore, its limited detection ratios stem from incidental effects, such as rare voting inconsistencies or historical vote misalignment, rather than explicit detection of malicious activity, and its detection capability does not significantly respond to changes in attacks. In contrast, DPoS model exhibits the poorest performance, with detection ratios generally below 5% across all MNR values. This is expected, as DPoS is purely vote-based and lacks any notion of trust or behavior profiling. Malicious nodes can easily manipulate voting outcomes or remain undetected, since there is no feedback mechanism for service quality or behavioral reliability. DPoS's rare successes attributable only to statistical chance rather than structured resilience.

In Camouflage Attack (a), where nodes alternate between good and bad behavior with a 50% split, Semi-Centralized and TEDPoS both respond well: Semi-Centralized maintains $\approx 86\%$, and TEDPoS achieves 82%, due to their strong behavioral tracking and temporal sensitivity. EDPoS 4 and DPoS remain ineffective. In Self-Promotion attack (b), Semi-Centralized slightly outperforms TEDPoS ($\approx 93\%$), while TEDPoS still provides

robust detection ($\approx 86\%$). In contrast, EDPoS 4 and DPoS fail to detect such manipulative behavior, with detection ratios near or below 13%. Last, in Bad-Mouthing attack (c), Semi-Centralized again leads detection with ratios around 92-93%, while TEDPoS closely follows with an average of 87%, benefiting from its trust reinforcement mechanism that safeguards honest nodes. EDPoS 4 and DPoS, being unaware of trust dynamics, cannot counter the effects of malicious downvoting.

Trust-Energy Composite Score (TECS) To provide a flexible and balanced evaluation of both malicious behavior detection effectiveness and energy sustainability, we introduce a novel metric called the Trust-Energy Composite Score (TECS). This metric linearly combines the average detection ratio ($DR \in [0, 1]$) and the normalized average remaining energy (E) using a weighting factor $\psi \in [0, 1]$, which allows prioritizing either trust-based detection or energy efficiency depending on the system designer's objective. TECS is formally defined as:

$$\text{TECS} = \psi \cdot \overline{DR}(t) + (1 - \psi) \cdot \frac{\overline{E}(t)}{E_{\max}} \quad (5.15)$$

Here in Equation (5.15), $\overline{DR}(t)$ denotes the average detection ratio of malicious nodes at time t , and $\overline{E}(t)$ is the average remaining energy normalized by the maximum energy capacity $E_{\max}$. The weighting factor ψ enables different analysis preferences:

- If detection ratio is the primary concern, a higher α (e.g., $\psi = 0.8$) can be chosen.
- If energy sustainability is more important, a lower α (e.g., $\psi = 0.3$) should be used.
- For a balanced evaluation of both security and sustainability, we select $\psi = 0.5$.

The TECS metric is bounded in $[0, 1]$ and facilitates direct comparison between trust-aware, energy-aware, and hybrid models by highlighting trade-offs between behavioral resilience and operational longevity. Figure 5.5 illustrates the TECS comparison under

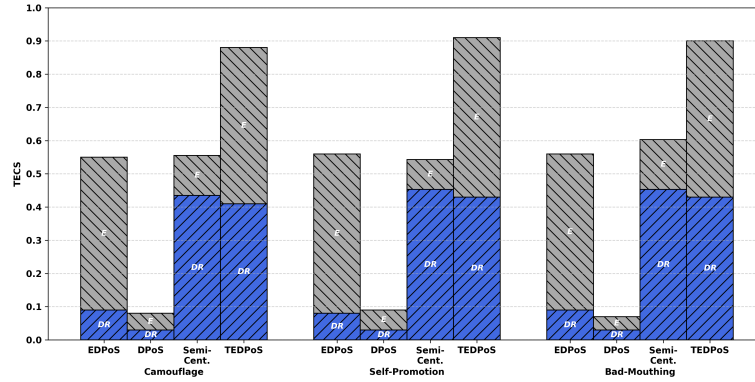


Figure 5.5: *TECS comparison of TEDPoS, Semi-Centralized [2], EDPoS 4 and DPoS under different attack types.*

three attack types, with a fixed MNR of 50% across all tests. For the camouflage attack, the camouflage ratio was also set to 50%. TEDPoS the highest scores around 0.89-0.91 by jointly sustaining a high detection ratio ($DR \approx 0.44$) and energy level ($E \approx 0.45$). In contrast, the Semi-Centralized model [2] remained between 0.55-0.61, as it lacks energy-aware mechanisms and relies solely on behavioral detection. EDPoS 4 performs moderately ($TECS \approx 0.52$) with strong energy balance but low detection, while DPoS remains the weakest overall ($TECS \approx 0.08$). Thus, TEDPoS's superior balance of security and sustainability under adversarial conditions.

5.3.0.2 Energy and Sustainability

This part of the results illustrates multiple energy analysis of nodes over 150 rounds to compare the energy efficiency and sustainability of the proposed TEDPoS model with other existing approaches. In resource-constrained environment, maintaining as much as energy is critical for long-term operation and fairness in contribution to consensus. As can be seen in Figure 5.6, EDPoS 4 conserves energy the most, with the end energy at approximately 92% of the initial. This is to be expected since the model prioritizes low-energy nodes during delegate selection, minimizing energy expenditure. On the other hand, DPoS demonstrates the poorest performance, rapidly depleting energy

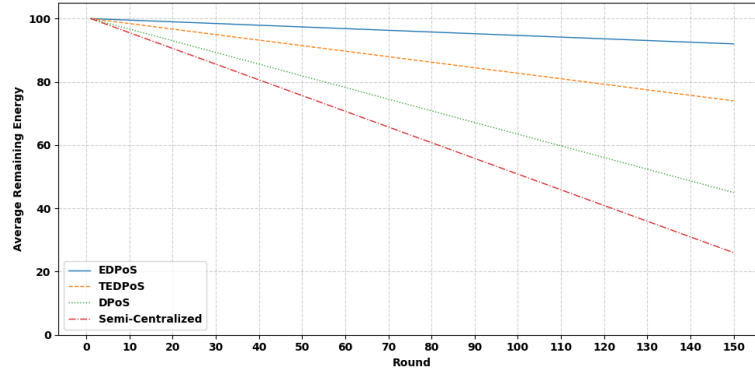


Figure 5.6: Average remaining energy levels over 150 rounds, showing the trade-off between energy efficiency and security across consensus models.

and dropping to below 45% by the end of the simulation. This is because DPoS does not consider energy awareness, and thus repeated selection of identical nodes occurs regularly, resulting in large energy disparity and premature draining of nodes. TEDPoS, while more energy-consuming than EDPoS 4, due to the energy consumption of trust calculations, however it outperforms DPoS significantly. Its final energy level is approximately 74%, reflecting its balanced design which considers not only energy levels but also trust and voting behaviors in delegation. While consuming more energy than EDPoS 4, this trade-off leads to better security and resilience, as reflected in earlier trust evaluations. Finally, Semi-Centralized depletes energy the fastest, ending with just 26% of initial energy. It performs even worse than DPoS due to the added cost of continuous trust calculations on top of unbalanced delegate selection. While effective in detection, it severely compromises sustainability in energy-constrained settings.

Figure 5.7, evaluates how evenly energy consumption is distributed across nodes by tracking the standard deviation (SD) of their energies over 150 rounds. Lower SD indicates more balanced energy use, while higher SD reveals load imbalance and potential for node exhaustion. Semi-Centralized exhibits the worst performance, with SD rising sharply above 27, indicating model can not handle energy levels. DPoS also performs

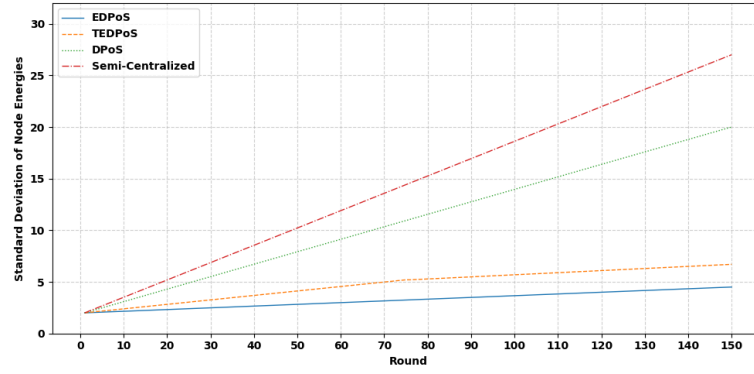


Figure 5.7: *Standard deviation of node energies over time, showing load balance and fairness in energy consumption across consensus models.*

poorly, reaching a SD of 20, and this confirms DPoS’s inherent fault of favoring high-vote nodes repeatedly, regardless of their energy level, which will eventually result in severe energy imbalance in the network. In contrast, EDPoS 4 maintains the most balanced energy usage, with SD values rising very slowly and remaining below 5 throughout the simulation. This is due to its prioritization of low-energy nodes, which balances the workload evenly over time. TEDPoS finds a middle ground, keeping SD under 7, showing that its hybrid model mitigates energy skew effectively, even though perfect balance is traded for better trust-awareness.

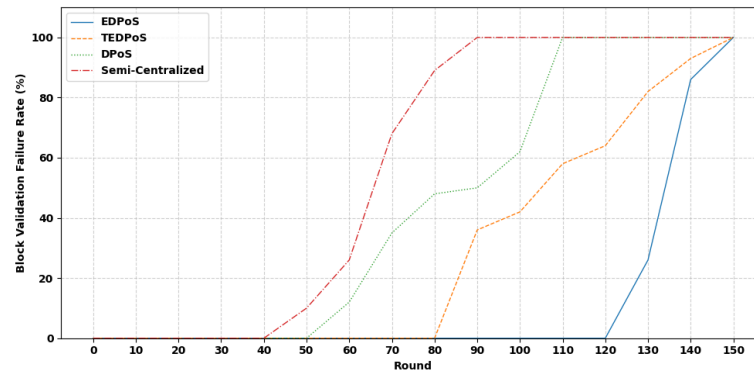


Figure 5.8: *Block validation failure rate over time, highlighting sustainability and load-balancing efficiency across consensus models.*

The Figure 5.8 presents the block validation failure rate of four consensus models over 150 rounds, highlighting their long-term feasibility under energy constraints. As

rounds progress, nodes expend energy through repeated participation, and once eligible delegates cannot be found, validation fails. Semi-Centralized fails first, starting from round 40 and reaching 100% failure by round 90. Due to the absence of energy awareness, it repeatedly selects the same high-trust nodes, rapidly depleting them. DPoS follows, starting to fail after round 50 and eventually total failure at round 110. TEDPoS, as more robust due to its hybrid use of trust and energy, starts failures round 80 and reaches full failure by round 150. This is because its trust mechanism introduces additional energy costs and occasional selection of trusted but energy-depleted nodes. EDPoS 4 achieves the longest operational lifespan, with system failure delayed until round 150. It distributes the workload more effectively. However, as shown in Figure 5.7, despite maintaining high average energy levels, some nodes remain underutilized due to low vote counts. Block validation fails not due to energy exhaustion, but because the system cannot select eligible nodes that simultaneously satisfy both energy and voting thresholds.

5.3.0.3 Performance

To evaluate the overall efficiency and responsiveness of each consensus model, we focus on performance-oriented metrics such as transaction throughput and finality delay. These metrics help us understand the trade-offs between security mechanisms and operational speed in resource-constrained environment.

Table 5.2: *Consensus Models: Performance Summary*

Model	Avg. TPS	Lifetime (Fig. 3.5)	Total Tx.
EDPoS	≈20	150	≈3000
TEDPoS	≈18	140	≈2520
DPoS	≈22	110	≈2420
Semi-Centralized	≈25	90	≈2250

Table 5.2 indicates the performance metrics of the four consensus models regarding average throughput (TPS), lifetime (rounds), and total transaction capacity. While exhibiting the highest average TPS (≈ 25), Semi-Centralized suffers from the shortest operational lifespan due to its lack of energy-aware mechanisms and costly trust evaluations, resulting in the lowest total transaction volume (≈ 2250). DPoS has relatively lower throughput (≈ 22 TPS) but achieves a slightly higher cumulative throughput (≈ 2420) due to its moderate lifetime. However, the absence of energy considerations makes it vulnerable to early node depletion. TEDPoS provides a more balanced performance by integrating energy evaluations. Although it comes with higher computation overhead due to trust computation, its adaptive selection and moderated energy consumption extend the system lifespan to 140 rounds, yielding a competitive total transaction count of ≈ 2520 . Finally, EDPoS 4 demonstrates the most sustainable performance. With energy-aware delegate selection and lightweight computation, it reaches the longest system up-time and ultimately processes the highest number of transactions (≈ 3000), despite having a lower average TPS. These findings highlight the trade-off between throughput and sustainability in consensus protocol design. Figure 5.9 presents the average finality delay in

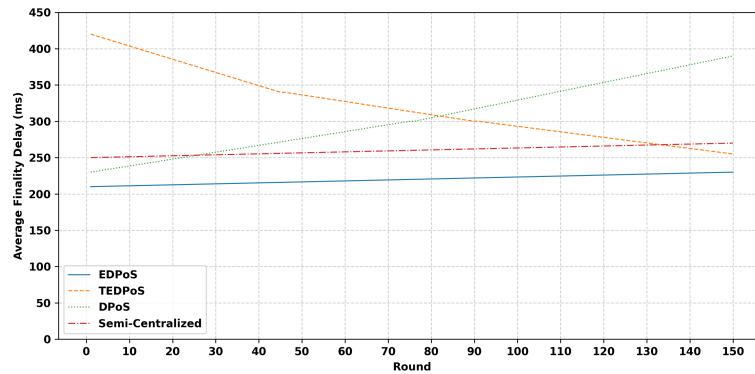


Figure 5.9: Block finality delay trends over time, showing latency convergence and performance optimization across consensus models.

milliseconds against time for the four consensus frameworks. Finality delay is a crucial performance metric, as it is a metric of how fast a transaction is considered irreversible

and completely confirmed on the blockchain. Lower values indicate faster consensus and more interactive user experience in real-world applications. EDPoS 4 consistently maintains the lowest finality delay of approximately 220 ms throughout simulation. That is primarily because its lightweight decision-making mechanism, which prioritizes high-energy and underutilized nodes. In comparison, DPoS exhibits a sharply increasing delay, beginning at 240 ms early on and eventually nearing 400 ms. This rapid increase because of its fast, vote-based selection. However, it exhibits a steady increase due to growing instability and imbalance in node energy levels, resulting from repeated selection of the same high-vote nodes, which reduces the system's ability to promptly reach consensus. TEDPoS, although starting with the highest delay by 422 ms, shows a notable decreasing trend, falling to less than 260 ms by round 150. Initially, trust evaluations and behavior checks initially incur overhead but energy awareness helps to reduce delay. Finally, the Semi-Centralized model shows a moderate and nearly linear increase in delay. Its semi centralized trust logic ensures quick decisions early on, but the absence of energy-awareness causes gradual degradation in node availability and responsiveness, thereby inflating the average delay.

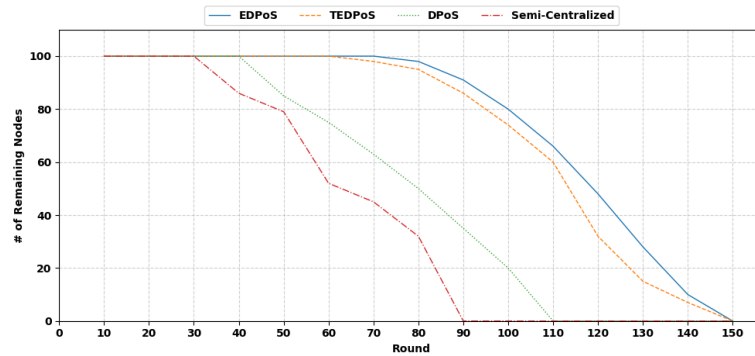


Figure 5.10: *System longevity.*

Figure 5.10 demonstrates the system longevity of all consensus models, in the number of active nodes over time. EDPoS 4 is the most operationally resilient with

energy-aware and rotation-based delegate selection, which optimizes energy consumption while preventing early depletion of nodes. TEDPoS performs reasonably well, but its trust-based selection mechanism occasionally favors high-trust nodes of low energy consumption, leading to faster depletion than EDPoS 4. In contrast, DPoS experiences early node losses as it repeatedly selects the same high-vote nodes regardless of their energy levels, leading to rapid imbalance and system degradation. Semi-Centralized exhausts fastest in that it lacks both energy-awareness and adaptive rotation. Although total energy may still exist in the system, it becomes fragmented across low-energy nodes so that none of them becomes eligible for further validation, effectively halting the consensus process.

6. CONCLUSION AND FUTURE WORK

This thesis presented a comprehensive study on the design and analysis of trust-aware and energy-efficient mechanisms tailored for decentralized IoT networks that function in dynamic, adversarial, and resource-constrained environments. In contexts where traditional centralized control is unavailable or impractical, such as mission-critical or infrastructure-less IoT deployments, it becomes essential to ensure secure, reliable, and autonomous coordination among heterogeneous nodes. This coordination essentially relies on efficient trust management, adaptive consensus integration, and energy-efficient behavior. To this end, the research proposed three different mechanisms and one use-case, each addressing particular shortcomings of existing approaches:

- TrustVista had proposed a trust assessment framework that measured node credibility at individual, group, and community levels. Utilizing an exponentially weighted moving average and a dynamic, multi-aspect trust decay function, TrustVista could rapidly distinguishing between trusted and malicious nodes, and resist long-term strategic manipulation attempts such as camouflage or discrediting.
- EDPoS: Energy-Aware Delegated Proof of Stake Resource-Constrained Devices is introduced dynamic energy aware mechanisms into the consensus layer, such as adaptive block sizing, historical energy trend tracking, and resource-sensitive participation control. This allowed the system to prevent early node energy depletion and sustain long-term operability in low-power environments.
- TEDPoS: Trust and Energy Aware Delegated Proof of Stake integrates trust evaluation, energy awareness, and voting into a single dynamic delegate selection protocol. TEDPoS introduced new concepts such as inverse-time trust weighting, energy-scaled trust ratings, behavior-based clustering, and a dynamic trust-energy-vote hybrid rating system that shifts its priorities based on the real-time energy

structure of the network. We further introduce Trust-Energy Composite Score (TECS), a unified metric that enables fair comparison across models by jointly quantifying security and sustainability.

In addition, a use-case-specific model, TEMBUV, was introduced in a dedicated appendix to demonstrate the applicability of the proposed trust concepts in specialized structures. It features a dual-layer blockchain design for structured environments, separating communication domains and employing a credit-based delegate selection strategy. TEMBUV shows how trust evaluation mechanisms can be adapted to operate robustly and securely even in partitioned networks.

The three proposed models: TrustVista, EDPoS, and TEDPoS, were implemented and evaluated using a specially designed OMNeT++ simulation framework. In contrast, TEMBUV, as discussed in the appendix, was modeled through a lightweight mathematical simulation to illustrate its conceptual applicability in domain-specific scenarios. Simulations employed various attack types, at different malicious node ratios. Evaluation metrics included trust accuracy, malicious node detection rate, energy fairness, block finality, system lifespan, and transaction rate. In all scenarios, the proposed models performed better than typical DPoS and existing energy-aware baselines, demonstrating not only improved resistance to adversarial activities but also improved sustainability and performance. This work demonstrates that integrating multi-perspective trust modeling, real-time energy feedback, and adaptive voting dynamics is not only feasible but also essential in building secure and scalable consensus protocols for upcoming decentralized networks. The established methodologies contribute to blockchain-based trust management theoretically and practically, serving as a point of departure for further research on trust-aware, self-regulatory, and energy-aware distributed systems.

6.1 Summary of Contributions

- A multi perspective trust framework, TrustVista, was proposed to evaluate trust at the node, group, and community levels. Trust scores were updated through an EWMA-based method, and a multi-factor dynamic decay function was applied to penalize stale or contradictory action. Improved trust convergence was achieved and made the system robust against long-term manipulation. A statistically driven approach towards malicious node detection and removal was employed using the implementation of Z-score analysis and dynamic trust thresholds to create fair, comprehensible, and decentralized decision-making for node isolation and elimination.
- A novel energy-aware consensus algorithm, EDPoS, was introduced to extend system lifespan and ensure sustainable validation procedures in IoT and energy constrained environments. Features such as adaptive block size, dynamic round-robin scheduling, and historical energy trend analysis were integrated in order to adaptively control participation dynamically according to real-time system conditions.
- A hybrid consensus protocol, TEDPoS, was put forward by combining trust, energy, and voting information to dynamically select reliable and sustainable delegates. TEDPoS employs temporal trust weighting, energy-aware scaling, and adaptive scoring to resist strategic attacks such as self-promotion, camouflage, and bad-mouthing, and guarantee balanced energy usage and fair participation in consensus rounds.
- The TEMBUV model was designed for UAV- and VANET-aggregated networks in post-disaster environments. With demultiplexing of the ground and air blockchains based on a credit-based DPoS protocol, high trust accuracy and robust witness selection were achieved even at advanced attack models.

6.2 Suggestions for Future Research

Although the suggested trust models produced encouraging outcomes, numerous fields for potential future work remain open for investigation:

- **Trust-aware Federated Learning Integration:** Federated learning systems can be improved by integrating trust scoring to increase data privacy and the trustworthiness of model updates in distributed machine learning environments.
- **Real-World Deployment and Hardware Evaluation:** The evaluation of the proposed systems on actual UAVs, vehicles, or IoT nodes would enable gauging the real-world performance regarding latency, effect of mobility, and energy consumption.
- **Scalability Enhancements:** A few of the optimization techniques, such as trust caching, partial node calculation, or hierarchical delegation, are worth exploring to mitigate communication overhead for large networks.

REFERENCES

- [1] L. Xiong and L. Liu, "Peertrust: supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 843–857, 2004.
- [2] Y. Liu, C. Zhang, Y. Yan, X. Zhou, Z. Tian, and J. Zhang, "A semi-centralized trust management model based on blockchain for data exchange in iot system," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 858–871, 2023.
- [3] K. Ashton *et al.*, "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.
- [4] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.
- [5] S. Vaidya, P. Ambad, and S. Bhosle, "Industry 4.0—a glimpse," *Procedia manufacturing*, vol. 20, pp. 233–238, 2018.
- [6] S. Al-Sarawi, M. Anbar, R. Abdullah, and A. B. Al Hawari, "Internet of things market analysis forecasts, 2020–2030," in *2020 Fourth World Conference on smart trends in systems, security and sustainability (WorldS4)*, pp. 449–453, IEEE, 2020.
- [7] S. Das and E. Mao, "The global energy footprint of information and communication technology electronics in connected internet-of-things devices," *Sustainable Energy, Grids and Networks*, vol. 24, p. 100408, 2020.
- [8] L. Tawalbeh, F. Muheidat, M. Tawalbeh, and M. Quwaider, "Iot privacy and security: Challenges and solutions," *Applied Sciences*, vol. 10, no. 12, p. 4102, 2020.
- [9] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of things (iot): Taxonomy of security attacks," in *2016 3rd international conference on electronic design (ICED)*, pp. 321–326, IEEE, 2016.
- [10] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
- [11] O. Novo, "Blockchain meets iot: An architecture for scalable access management in iot," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.
- [12] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in *International workshop on open problems in network security*, pp. 112–125, Springer, 2015.
- [13] F. Saleh, "Blockchain without waste: Proof-of-stake," *The Review of financial studies*, vol. 34, no. 3, pp. 1156–1190, 2021.

- [14] D. Larimer, “Transactions as proof-of-stake,” *Nov-2013*, vol. 909, 2013.
- [15] D. Larimer, “Delegated proof-of-stake (dpos),” *Bitshare whitepaper*, vol. 81, p. 85, 2014.
- [16] M. Blaze, J. Feigenbaum, and J. Lacy, “Decentralized trust management,” in *Proceedings 1996 IEEE symposium on security and privacy*, pp. 164–173, IEEE, 1996.
- [17] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. Leung, “Blockchain-based decentralized trust management in vehicular networks,” *IEEE internet of things journal*, vol. 6, no. 2, pp. 1495–1505, 2018.
- [18] C. V. Mendoza and J. H. Kleinschmidt, “Mitigating on-off attacks in the internet of things using a distributed trust management scheme,” *International Journal of Distributed Sensor Networks*, vol. 11, no. 11, p. 859731, 2015.
- [19] J. Guo, R. Chen, and J. J. Tsai, “A survey of trust computation models for service management in internet of things systems,” *Computer Communications*, vol. 97, pp. 1–14, 2017.
- [20] Z. Yan, P. Zhang, and A. V. Vasilakos, “A survey on trust management for internet of things,” *Journal of network and computer applications*, vol. 42, pp. 120–134, 2014.
- [21] A. Altaf, H. Abbas, F. Iqbal, and A. Derhab, “Trust models of internet of smart things: A survey, open issues, and future directions,” *Journal of Network and Computer Applications*, vol. 137, pp. 93–111, 2019.
- [22] A. Sharma, E. S. Pilli, A. P. Mazumdar, and P. Gera, “Towards trustworthy internet of things: A survey on trust management applications and schemes,” *Computer Communications*, vol. 160, pp. 475–493, 2020.
- [23] G. Zacharia and P. Maes, “Trust management through reputation mechanisms,” *Applied Artificial Intelligence*, vol. 14, no. 9, pp. 881–907, 2000.
- [24] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, “Trm-iot: A trust management model based on fuzzy reputation for internet of things,” *Computer Science and Information Systems*, vol. 20, pp. 1207–1228, 2011.
- [25] N. B. Truong, H.-W. Lee, R. Askwith, and G. M. Lee, “Toward a trust evaluation mechanism in the social internet of things,” *Sensors (Basel, Switzerland)*, vol. 17, 2017.
- [26] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis, “The role of trust management in distributed systems security,” *Secure Internet programming: security issues for mobile and distributed objects*, pp. 185–210, 1999.
- [27] A. Abdul-Rahman and S. Hailes, “A distributed trust model,” in *Proceedings of the 1997 workshop on New security paradigms*, pp. 48–60, 1998.

- [28] R. Zhou and K. Hwang, "Powertrust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Transactions on parallel and distributed systems*, vol. 18, no. 4, pp. 460–473, 2007.
- [29] A. Josang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th bleled electronic commerce conference*, vol. 5, pp. 2502–2511, Citeseer, 2002.
- [30] R. Zhou, K. Hwang, and M. Cai, "Gossiptrust for fast reputation aggregation in peer-to-peer networks," *IEEE transactions on knowledge and data engineering*, vol. 20, no. 9, pp. 1282–1295, 2008.
- [31] U. E. Tahta, S. Sen, and A. B. Can, "Gentrust: A genetic trust management model for peer-to-peer systems," *Applied Soft Computing*, vol. 34, pp. 693–704, 2015.
- [32] D. He, C. Chen, S. Chan, J. Bu, and A. V. Vasilakos, "Retrust: Attack-resistant and lightweight trust management for medical sensor networks," *IEEE transactions on information technology in biomedicine*, vol. 16, no. 4, pp. 623–632, 2012.
- [33] W. Li and H. Song, "Art: An attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 960–969, 2015.
- [34] J. Zhang, K. Zheng, D. Zhang, and B. Yan, "Aatms: An anti-attack trust management scheme in vanet," *IEEE Access*, vol. 8, pp. 21077–21090, 2020.
- [35] X. Fan, L. Liu, M. Li, and Z. Su, "Eigentrustp++: Attack resilient trust management," in *8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pp. 416–425, IEEE, 2012.
- [36] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th international conference on World Wide Web*, pp. 640–651, 2003.
- [37] S. Schmidt, R. Steele, T. S. Dillon, and E. Chang, "Fuzzy trust evaluation and credibility development in multi-agent systems," *Applied Soft Computing*, vol. 7, no. 2, pp. 492–505, 2007.
- [38] A. Lahbib, K. Toumi, A. Laouiti, A. Laube, and S. Martin, "Blockchain based trust management mechanism for iot," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–8, IEEE, 2019.
- [39] P. K. Singh, R. Singh, S. K. Nandi, K. Z. Ghafoor, D. B. Rawat, and S. Nandi, "Blockchain-based adaptive trust management in internet of vehicles using smart contract," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3616–3630, 2020.

- [40] M. Zhaofeng, W. Lingyun, W. Xiaochang, W. Zhen, and Z. Weizhe, “Blockchain-enabled decentralized trust management and secure usage control of iot big data,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4000–4015, 2019.
- [41] S. Malik, V. Dedeoglu, S. S. Kanhere, and R. Jurdak, “Trustchain: Trust management in blockchain and iot supported supply chains,” in *2019 IEEE international conference on blockchain (Blockchain)*, pp. 184–193, IEEE, 2019.
- [42] C. Marche and M. Nitti, “Trust-related attacks and their detection: A trust management model for the social iot,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3297–3308, 2020.
- [43] Q. Xiong, N. Sohrabi, H. Dong, C. Xu, and Z. Tari, “Aicons: An ai-enabled consensus algorithm driven by energy preservation and fairness,” *ArXiv*, vol. abs/2304.08128, 2023.
- [44] H. W. Kuhn, ed., *Classics in Game Theory*. Princeton: Princeton University Press, 1997.
- [45] H. Zaheer, M. Shoaib, F. Iqbal, S. Arshad, A. Altaf, E. G. Villena, I. de la Torre Diez, and I. Ashraf, “An energy-efficient technique to secure internet of things devices using blockchain,” *J. Netw. Syst. Manage.*, vol. 32, Oct. 2024.
- [46] X. Xu, G. Sun, and H. Yu, “An efficient blockchain pbft consensus protocol in energy constrained iot applications,” in *2021 International Conference on UK-China Emerging Technologies (UCET)*, pp. 152–157, 2021.
- [47] C. Singhal, S. Barick, and R. Sonkar, “Application and energy-aware data aggregation using vector synchronization in distributed battery-less iot networks,” in *Proceedings of the 25th International Conference on Distributed Computing and Networking, ICDCN ’24*, (New York, NY, USA), p. 222–231, Association for Computing Machinery, 2024.
- [48] S. Wadhwa, S. Rani, Kavita, S. Verma, J. Shafi, and M. Wozniak, “Energy efficient consensus approach of blockchain for iot networks with edge computing,” *Sensors*, vol. 22, no. 10, 2022.
- [49] P. Wang, “Fedchain: An efficient and secure consensus protocol based on proof of useful federated learning for blockchain,” *ArXiv*, vol. abs/2308.15095, 2023.
- [50] S. Li, Q. Hu, and Z. Wang, “Pofel: Energy-efficient consensus for blockchain-based hierarchical federated learning,” 2023.
- [51] N. Yonezawa, “Proof of team sprint: A collaborative consensus algorithm for reducing energy consumption in blockchain systems,” 2024.

- [52] A. Alioua, N. Bouchemal, R. Mati, and M.-L. Messai, "Blockchain-inspired incentive mechanism for trust-aware offloading in mobile edge computing," in *2024 IEEE 49th Conference on Local Computer Networks (LCN)*, pp. 1–8, IEEE, 2024.
- [53] A. Padma and M. Ramaiah, "Glsbiot: Gwo-based enhancement for lightweight scalable blockchain for iot with trust based consensus," *Future Generation Computer Systems*, vol. 159, pp. 64–76, 2024.
- [54] B. Yu and M. P. Singh, "Detecting deception in reputation management," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, (New York, NY, USA), p. 73–80, Association for Computing Machinery, 2003.
- [55] F. Bao, R. Chen, M. Chang, and J.-H. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *IEEE transactions on network and service management*, vol. 9, no. 2, pp. 169–183, 2012.
- [56] G. M. da Silva Barros, G. S. de Aquino, P. H. de Queiroz Lopes, and F. T. Gonçalves de Oliveira, "Edraft: An energy-aware consensus algorithm for low-power iot devices," in *2023 Symposium on Internet of Things (SIoT)*, pp. 1–5, 2023.
- [57] D. Hanggoro, J. H. Windiatmaja, A. Muis, R. F. Sari, and E. Pournaras, "Energy-aware proof-of-authority: Blockchain consensus for clustered wireless sensor network," *Blockchain: Research and Applications*, vol. 5, no. 3, p. 100211, 2024.
- [58] Y. Liu, J. Gao, Y. Lu, R. Cao, L. Yao, Y. Xia, and D. Han, "Lightweight blockchain-enabled secure data sharing in dynamic and resource-limited uav networks," *IEEE Network*, 2024.
- [59] Z. Su, Y. Wang, Q. Xu, and N. Zhang, "Lvbs: Lightweight vehicular blockchain for secure data sharing in disaster rescue," *IEEE Transactions on dependable and secure computing*, vol. 19, no. 1, pp. 19–32, 2020.
- [60] O. Ltd., "Omnet++, discrete event simulator." <https://omnetpp.org/>.
- [61] V. Held, "On the meaning of trust," *Ethics*, vol. 78, no. 2, pp. 156–159, 1968.
- [62] B. Xu, D. Luthra, Z. Cole, and N. Blakely, "Eos: An architectural, performance, and economic analysis," *Retrieved June*, vol. 11, no. 2019, p. 41, 2018.
- [63] T. DAO, "Advanced decentralized blockchain platform," tech. rep., Tech. Rep: 2018. Available online: <https://static.hamtapay.net/uploads...>, 2018.
- [64] G. Möllering, "The nature of trust: From georg simmel to a theory of expectation, interpretation and suspension," *Sociology*, vol. 35, no. 2, pp. 403–420, 2001.

- [65] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. Adaptive computation and machine learning, MIT Press, 2009.
- [66] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. The MIT Press, second ed., 2018.
- [67] S. Haykin, *Adaptive Filter Theory*. Pearson, 2014.
- [68] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, vol. 5, pp. 281–298, University of California press, 1967.
- [69] K. Hoffman, D. Zage, and C. Nita-Rotaru, “A survey of attack and defense techniques for reputation systems,” *ACM Computing Surveys (CSUR)*, vol. 42, no. 1, pp. 1–31, 2009.
- [70] Y. Cai and D. Zhu, “Fraud detections for online businesses: a perspective from blockchain technology,” *Financial Innovation*, vol. 2, pp. 1–10, 2016.
- [71] C. Dellarocas, “Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior,” in *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pp. 150–157, 2000.
- [72] İpek Motorcu, E. Şahin Mektepli, A. Hacıfevzioğlu, and K. Kalkan, “Tembuv: Trust ensuring mechanism based on blockchain using uavs and vanets,” *2024 6th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, pp. 1–4, 2024.



APPENDICES

APPENDIX A. TEMBUV: TRUST ENSURING MECHANISM BASED ON BLOCKCHAIN USING UAVS AND VANETS

This appendix presents an additional model developed during the research process, named TEMBUV (Trust Ensuring Mechanism Based on Blockchain using UAVs and VANETs). While not part of the core thesis contributions, TEMBUV offers a trust management and consensus framework tailored for UAV-VANET integration. It was designed to explore the integration of air and ground communication layers through separate blockchains, with a focus on credit-based DPoS and dual-layer trust recording. Due to its application-specific nature, TEMBUV is excluded from the primary evaluation sections but is included here for completeness. Today, the world has witnessed a sharp rise in the frequency and severity of natural disasters such as earthquakes, floods, and wildfires. One of the most crucial challenges that follow such catastrophes is the breakdown of the communication system, which adversely impacts the coordination and effectiveness of emergency operations. The lack of reliable communication channels delays information flow, hinders decision-making, and increases the risk to both victims and rescue teams. To address this challenge, UAVs and VANETs offer effective solutions by establishing secure, and decentralized communication networks that are infrastructure-independent. Also, the model proposes an solid integration of governmental and private smart devices to increase privacy and scalability. Nonetheless, the decentralized and heterogeneous nature of these networks introduces new vulnerabilities, particularly with regard to trust, security, and node coordination.

This chapter presents TEMBUV [72], a trust ensuring mechanism based on blockchain, utilizing UAVs and VANETs, designed specifically for post-disaster situations. Through the transparency and immutability of blockchain, TEMBUV securely records node behaviors and trust values while enabling a credit-based DPoS voting mechanism. The suggested model ensures dynamic trust assessment, secure data exchange, and secure consensus in highly dynamic environments.

A.1 Overview of TEMBUV Architecture

In the aftermath of disasters, traditional communication infrastructure such as Road Side Unit (RSU)s and base stations are often damaged and become inoperative, which creates significant challenges for coordination, information dissemination, and situational awareness. The TEMBUV architecture addresses this problem by combining aerial and ground-based technologies into a hybrid, decentralized communication and trust management framework. As depicted in Figure A.1, the system is composed of four key components: UAVs, RSUs, VANETs, and a dual-layered blockchain infrastructure, namely the Air Blockchain and the Ground Blockchain. Each component act as follows:

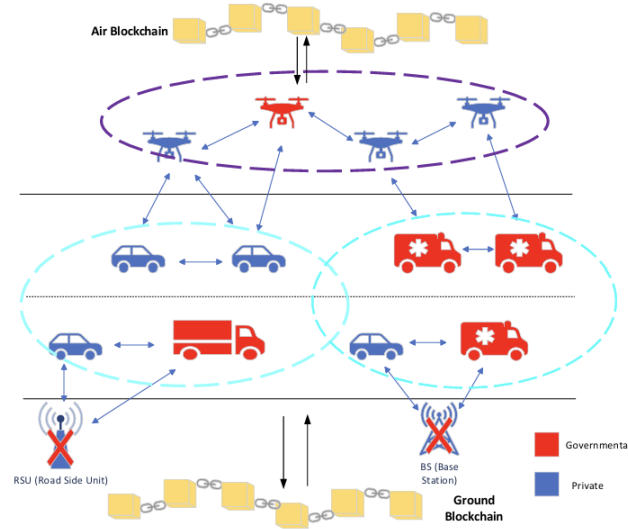


Figure A.1: *System Model of TEMBUV*

- **UAV:** This component operates as mobile relay stations equipped with position sensors, processors, and communication modules. In the event of RSU failure, they dynamically take over communication responsibilities, ensuring continuous service coverage. UAVs were addressed in two categories:
 - **Governmental UAVs:** They begin with high initial trust levels, but also their behaviors affect their score.
 - **Private UAVs:** They are integrated from the public domain and start with lower trust values that are incrementally updated based on service quality.

This classification allows robust trust evaluation and ensures more secure aerial communication during critical operations.

- **VANET:** On the ground layer, VANETs include both governmental vehicles, such as ambulances, fire trucks, and privately owned smart vehicles. Governmental VANETs are privileged by high start trust score, while private ones does not.
- **RSUs:** These are stationary devices, and they have critical importance for VANETs. They are mainly positioned at critical places along the road.
- **Blockchain:** This technology integrated to enhance security and privacy to store vehicles meta data and trust scores, also it is used to ensure incoming and outgoing packet integrity. This component divided into two layers:
 - **Air Blockchain:** This layer stores UAVs' data.
 - **Ground Blockchain:** This layer stores VANETs' data.

Both layers store and manages the same type of data, as mentioned before meta data and trust scores and both has governmental and private nodes as well. At the initial stage of the network run, only governmental devices are allowed to be delegates.

To manage all of these components, a modified DPoS consensus mechanism is proposed, which is enhanced by gathering both trust scores and validator selection voting. The novelty of this proposed consensus architecture is that the system also considers negative votes unlike conventional DPoS. The division between blockchains make sure the isolation for potential malicious activities, even one of those blockchains captured, the malicious actor could not be able to diffused to other. As it shown in Figure A.1, RSUs and base stations are down, so UAVs and VANETs are in use. Thus, their data is securely recorded and stored in both blockchains.

A.2 Trust Model

As mentioned in the previous section, within the start of the network run only governmental devices are be able to delegates. Nevertheless, as time passes if any private device achieve enough trustworthiness, then they can become a delegate if they also can pass the specified threshold. Each node in the system, whether a UAV or a ground vehicle, maintains a quantified TS, which reflects its past service performance and reputation within the network. These trust scores evolve dynamically based on feedback received from other nodes that benefit from its services. This TS will be assigned after each incoming or outgoing data transaction, regardless of its goodness or badness. If a transaction is well-transactioned by the provider then it can be good TS or vice versa. The threshold for eligibility of delegation is calculated by evaluating governmental-private vehicle ratio. The trust score of a node is determined using a weighted update mechanism, as shown in Equation A.1:

$$TS = TS_i + c_k \sum_k TS_k - c_n \sum_n TS_n \quad (\text{A.1})$$

The initial trust score TS_i is assigned based on the classification of the node: governmental nodes are given higher trust values at initialization, while private nodes start with a baseline trust score. As nodes begin to offer services, their trust scores are updated according to positive and negative feedback from service recipients. In Equation (A.1), TS_k denotes the trust scores of the nodes that approve the service, while TS_n represents those that disapprove. The coefficients c_k and c_n are weighting factors that determine the impact of positive and negative feedback, respectively. This structure ensures that trustworthy nodes can quickly improve their scores when providing successful services, whereas misbehaving nodes are penalized through disapproval. The summation terms incorporate collective feedback:

- $\sum_k TS_k$ boosts the score of the service provider when many trusted nodes confirm the service quality.

- While $\sum_n TS_n$ imposes a penalty based on the weight of disapproving nodes.

This trust model enables the network to maintain self-regulation by rewarding cooperative behavior and penalizing malicious activity. It is particularly critical in the absence of centralized control since trust is the foundation layer for consensus participation, delegation of service, and overall system integrity.

A.3 Credit-based DPoS and Voting Model

In the proposed TEMBUV system, consensus is achieved through a credit-based DPoS mechanism. This DPoS variation consensus algorithm is used to securely and immutably protect the data. Unlike traditional DPoS, which treats all nodes as equally eligible for block production based solely on votes, the credit-based variant integrates each node's TS into the selection process, thereby prioritizing more reliable participants. In this model, nodes cast both positive and negative votes, which are used to update the Consensus Score (CS) of candidate nodes. Initially, the consensus score is set equal to the trust score of each node. However, this score is dynamically adjusted during the voting process to reflect real-time trust feedback from the network. The nodes with trust scores below a certain threshold are excluded from participating, as they are deemed unreliable. The equation for CS is given by:

$$CS = TS + d \sum_k TS_k - d \sum_n TS_n \quad (\text{A.2})$$

Here, TS is the initial trust score of the node, $\sum_k TS_k$ represents the cumulative trust scores of nodes giving positive votes, $\sum_n TS_n$ denotes those giving negative votes, and d is a coefficient that adjusts the influence of each vote. The voting process is designed to avoid centralization by enforcing a balance between approval and disapproval. Each node can vote just once positive and negative vote during an election cycle. This limitation encourages fair competition and reduces the risk of single handed domination from building up within the network. Once all votes are collected, the nodes with the highest consensus scores are selected as validators. This approach not only enhances fairness but also renders that block validators are selected based on both reputation and network feedback, so more secure and trustworthy blockchain infrastructure can be achieved, especially suitable for mission-critical systems like disaster response environments involving UAVs and VANETs.

A.4 Simulation Setup

To evaluate the performance and robustness of the proposed TEMBUV trust mechanism in realistic post-disaster environments, a comprehensive simulation setup was designed. Key system parameters used in the simulation are summarized in Table A.1:

To measure system accuracy and resistance to adversarial behaviors, two key evaluation metrics are used:

Table A.1: *Simulation Parameters for TEMBUV Evaluation*

Parameter	Value
Number of VANET Nodes	1000
Number of UAVs	100
Simulation Rounds	60
Trustworthiness Threshold (TS_{th})	0.5
Approval Coefficient (c_k)	0.2
Disapproval Coefficient (c_n)	0.2
Consensus Score Coefficient (d)	0.5
Ground Witness Count	101
Air Witness Count	21

- **ACC:** This metric measures the classification accuracy of the system in identifying trustworthy and malicious nodes. It is calculated as:

$$|ACC| = \frac{TP + TN}{TP + TN + FP + FN} \quad (A.3)$$

Where:

- TP : True positives.
- FP : False positives.
- TN : True negatives.
- FN : False negatives.

- **Secure Witness Ratio (SWR):** This metric captures the proportion of benevolent nodes among the selected consensus committee. It is defined as:

$$SWR = \frac{B}{B + M} \quad (A.4)$$

Where:

- B : The number of benevolent witnesses.
- M : The number of malicious witnesses in the voting group.

In order to rigorously assess the resilience and adaptability of the proposed TEM-BUV system, we simulated a variety of adversarial behaviors that represent real-world threats encountered in disaster environments. These attack models target the trust and consensus mechanisms by manipulating trust relationships, voting behavior, and the perception of node reliability. All attack models challenge the robustness of the system regarding how efficiently it maintains accuracy and consistency in trust evaluations even against coordinated or dishonest attackers.

- **Malicious Nodes with Camouflage Attack:** It represents attackers who behave well for a certain period to gain trust, and then become malicious. The camouflage level is 0 to 100, with 0 indicating constant malicious behavior, and 100 indicating perfect camouflage, making malicious nodes totally indistinguishable from honest nodes. This attack is particularly challenging as it relies on the system's reliance on behavior history, testing how quickly and effectively TEMBUV can detect and respond to abrupt changes in node behavior.
- **Non-discrediting Malicious Supportive Attack:** Malicious nodes in this case do not launch individual attacks against honest nodes but, instead, artificially increase the other malicious nodes' trust value by consistently providing positive ratings. The hidden support manipulates the system's consensus process by increasing the probability that malicious nodes get consensus roles, without explicitly triggering trust penalties.
- **Malicious Supportive and Discrediting Attack:** This dual attack combines the strategy of mutual support among malicious nodes with proactive attempts to lower the trust scores of legitimate nodes. By sending negative ratings to honest nodes and positive ratings to collaborators, attackers will attempt to dominate the consensus committee and corrupt the entire decision-making process. This attack focuses on defense of the system against trust tampering and disinformation spreading.
- **Malicious Collaborative Attack:** In this scenario, attackers create densely connected groups of nodes, all of which consistently support each other's trust scores through mutual support and simultaneously downvote non-members of their group. These malicious clusters simulate coordinated attacks in decentralized settings and test the network's ability to prevent the formation of trust monopolies or voting cartels.

All these attack models introduce new manipulation methods based on exploiting vulnerabilities of decentralized systems. By evaluating performance against these threat models, the TEMBUV framework demonstrates its capability to preserve trust integrity, detect collusion, and mitigate the risk of compromised consensus in extremely dynamic

and critical scenarios such as post-disaster scenarios. The combination of node-level feedback, hierarchical trust calculation, and credit-based DPoS ensures a balanced and adaptive defense against both stealthy and aggressive malicious attempts.

A.5 Simulation Results and Evaluation

Simulations done versus mentioned attacking scenarios and our model is compared with [59]. LVBS offers a model that prioritizes recent behaviors are given higher priority than past ones and assesses it with an attack model whereby malicious nodes consistently exhibit malicious behavior. While they achieve good results, their model lacks robustness toward heterogeneous attack strategies, such as coordinated attacks with camouflage and different strategies used by malicious collectives. The mode tested different attack models and provided the following findings:

A.5.1 Malicious Nodes with Camouflage Attack

In this scenario, malicious nodes aim to evade detection by periodically behaving like good ones, thereby attempting to blend in with benevolent peers. The camouflage ratio represents how frequently the nodes mimic honest behavior. As shown in Figure A.2, the simulation compares the performance of the proposed TEMBUV system against the baseline LVBS model under increasing camouflage intensity, ACC as the evaluation metric. The results show that TEMBUV has a very stable accuracy rate within the range 0.95

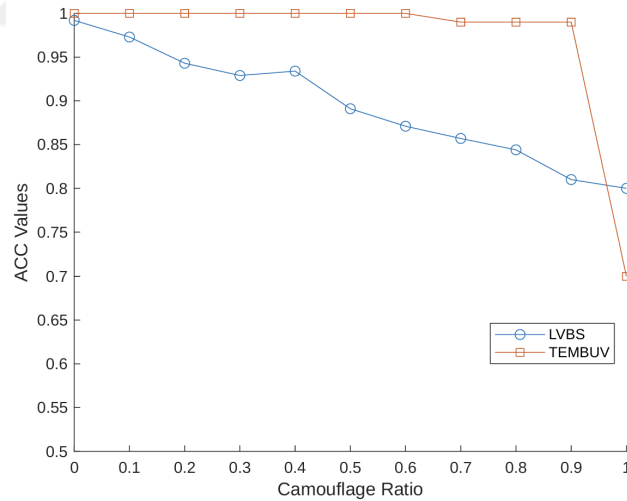


Figure A.2: System comparison among different camouflage ratios

to 1.0 for the majority of camouflage levels, with the exception of a sharp decline at the extreme situation of complete camouflage (camouflage ratio = 1.0). By contrast, LVBS begins to show gradual accuracy loss even at lower camouflage ratios, with a clear declining trend as camouflage increases. This performance gap can be attributed to several key differences in the internal trust dynamics of both models:

- LVBS places more weight on recent transactions, which makes it more susceptible to deceptive behaviors when malicious nodes behave correctly in the short term. This can lead to temporary increases in TS for attackers, allowing them to bypass detection mechanisms.
- TEMBUV, on the other hand, utilizes a cumulative and feedback-enriched trust architecture, where both recent and historical trust scores contribute to the overall evaluation. Additionally, the integration of both approval and disapproval feedback mechanisms makes the system more sensitive to deviations from long-term behavioral consistency.
- The presence of governmental and private trust baselines also allows TEMBUV to differentiate node reliability more effectively, making it less likely for camouflage to succeed across trust layers.

Overall, the experimental results validate that TEMBUV exhibits strong resilience against camouflage attacks up to a very high deception threshold, offering a more strong defense compared to traditional reactive models such as LVBS.

A.5.2 Malicious Supportive and Discrediting Attack

This attack scenario involves malicious nodes acting in a polarized and aggressive manner by giving positive votes exclusively to other malicious nodes and discrediting honest nodes with negative votes. Unlike camouflage-based attacks, here the malicious behavior is overt and consistent, making it more traceable for a well-structured trust mechanism. As illustrated in Figure A.3, our proposed TEMBUV model demonstrates robust

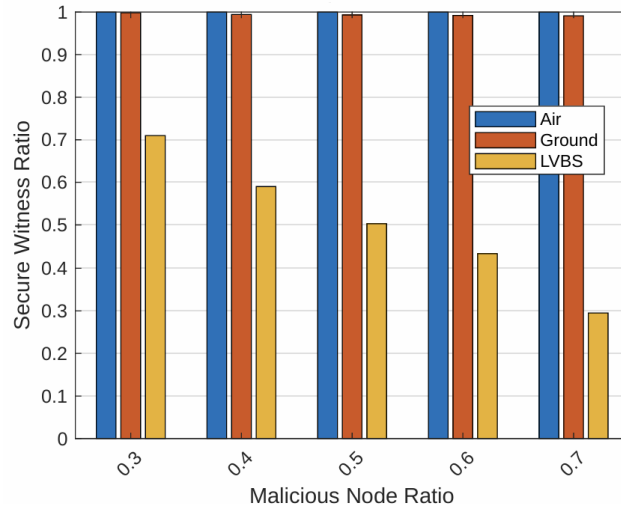


Figure A.3: System comparison among different malicious supportive and discrediting attack ratios

performance against this type of threat. Even at high malicious node ratios such as 0.7,

both the Air and Ground layers of our system maintain a SWR close to 1.0, indicating nearly perfect witness selection integrity. In contrast, the baseline LVBS model's performance deteriorates significantly with rising malicious node ratios, dropping to below 0.4 SWR at the highest tested threat level. Several factors contribute to TEMBUV's superior performance:

- The strict penalty mechanism in the trust update process effectively downgrades nodes that continuously issue biased or malicious votes.
- Unlike models that weigh only recent activity, TEMBUV leverages both cumulative trust history and network-wide feedback, making it resistant to temporary behavioral manipulation.
- The separation into air and ground blockchain layers provides a redundancy buffer, where even if malicious influence starts spreading in one layer, the other layer remains uncompromised, thereby preserving decision integrity.

The consistent trend in the bar graph shows that VANETs as well as UAVs are favorably influenced by a tightly coupled feedback-based trust evaluation model. In spite of the increasing malicious presence, TEMBUV ensures reliable witness selection due to its adaptive and distributed trust system.

A.5.3 Non-discrediting Malicious Supportive Attack

Figure A.4 illustrates the performance of the proposed system under the Non-discrediting Malicious Supportive Attack, where malicious nodes consistently support each other by assigning positive votes, yet refrain from discrediting benevolent nodes. Despite this passive-aggressive behavior, the model sustains a remarkably high SWR across different malicious node ratios ranging from 0.3 to 0.7. Although there is a slight degradation in performance compared to the Figure A.3, the system still preserves strong detection capabilities. This minor drop can be attributed to the fact that malicious behavior in this scenario is less aggressive and harder to distinguish, since malicious nodes do not actively penalize honest nodes. However, thanks to the dual-layered air and ground blockchain design, coupled with the proposed credit-based DPoS voting strategy, the model continues to penalize suspicious support clusters and preserves SWR above 0.9 in both layers. In contrast, LVBS shows a significantly sharper performance decline, particularly as the ratio of malicious nodes increases. This is due to its higher reliance on recent behavior data, which makes it more susceptible to slow-collusion attacks among malicious entities. Our model's hybrid trust framework allows broader historical context and community-level consensus verification, thus greater resilience against such stealthy coordination strategies.

A.5.4 Malicious Collaborative Attack

Figure A.5 illustrates the performance of the system when it is subjected to the Malicious Collaborative Attack, where 30% of all the nodes are marked as malicious

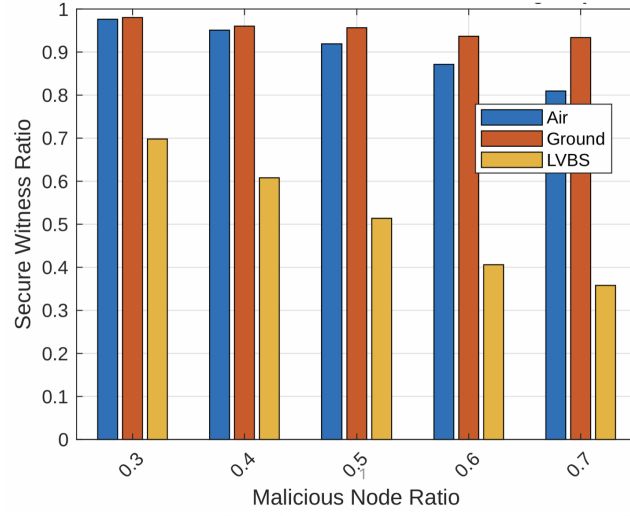


Figure A.4: System comparison among different non-discrediting malicious supportive attack ratios

and partitioned into different numbers of collectives, i.e., 1, 2, 10, and 100 groups. This attack is for testing the resilience of the system against distributed coordination, as malicious agents attempt to exploit their influence through group-based support. The results

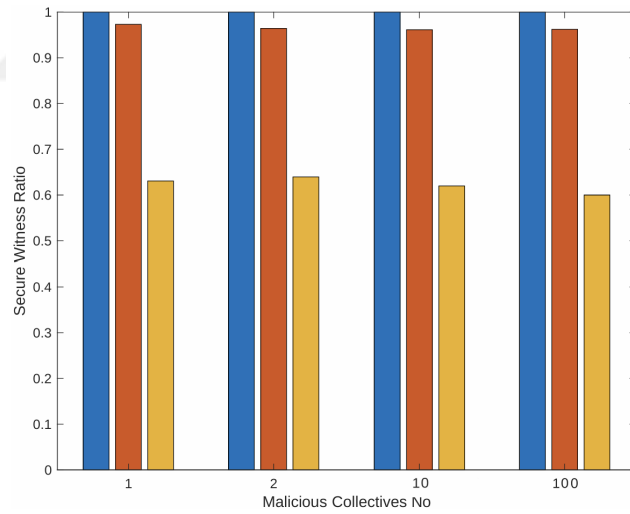


Figure A.5: System comparison among different malicious collaborative attack collectives number

clearly show that our proposed model maintains its high SWR in every setup regardless of the way in which the attacking nodes are broken into parts. Both the air and ground blockchain layers still have high accuracy even during the most fragmented attack (100 collectives). This proves that the system's detection methods are not affected

by the structural strategies of the attackers, as long as their general ratio remains unchanged. This robustness stems from the trust model's reliance not only on absolute trust scores but also on contextual group and community-level evaluations. Additionally, since trust measurement is being carried out in a decentralized and cross-validated manner in both layers, the system effectively precludes even the possibility of small-scale malicious clusters from passing undetected. In contrast, the LVBS model does significantly worse, particularly in situations with compromised attacker structures. That is likely because its relatively superficial evaluation logic does not detect elusive collusion patterns in small attacker subgroups. Ultimately, Fig. A.5 demonstrates that the proposed model is not only resilient to centralized, large-scale attacks but even against distributed and misleading collaborative manners, which represents its scalability and resilience for realistic deployment environments.

