RESEARCH PAPER
Future Research in AI & IoT, 2026, 2(2)
Homepage: https://frai.journals.umz.ac.ir/
DOI: 10.22080/frai.2026.31194.1051

KPIforBlockchain: A Model-Driven and AI-Assisted Framework for Blockchain Performance Evaluation

Kimiya Karimi Dehkordi[1], Leila Samimi-Dehkordi [2]*, Abbas Horri[3]

[1] Department of Computer Engineering, Faculty of Technology and Engineering, Shahrekord University, Shahrekord, Iran.

[2] Department of Computer Engineering, Faculty of Technology and Engineering, Shahrekord University, Shahrekord, Iran.

[3] Department of Computer Engineering, Faculty of Technology and Engineering, Shahrekord University, Shahrekord, Iran.

Article Info
Received 4 February 2026
Accepted 20 February 2026
Available online -----

Keywords:
Blockchain performance evaluation;
Key Performance Indicators (KPI);
Metamodel-based analysis; Model-Driven Engineering (MDE);
Large Language Models (LLMs).

Abstract:

Blockchain technology has become a fundamental infrastructure for decentralized and trustworthy data management across a wide range of application domains. As blockchain systems increasingly operate in large-scale, intelligent, and interconnected environments, systematic performance evaluation has emerged as a critical challenge. Existing studies predominantly focus on isolated performance indicators or platform-specific benchmarking, lacking a unified and structured representation of performance dimensions and their interdependencies. This study proposes KPIforBlockchain, a model-driven framework for systematically representing, organizing, and analyzing key performance indicators (KPIs) in blockchain systems. The framework introduces an extensible metamodel that explicitly captures relationships between blockchain features and multi-dimensional performance indicators, including performance, cost, scalability, security, decentralization, and auxiliary factors. A set of formally defined computational indicators is integrated into the framework to support structured analytical reasoning. The applicability and analytical capability of the proposed framework are demonstrated through representative case-study blockchain models, illustrating how diverse configurations can be evaluated in a unified manner without relying on empirical benchmarking data. In addition, large language model–assisted generation of blockchain models is investigated, and the quality of generated models is assessed using precision, recall, and F1-score metrics under different prompting strategies. A metamodel-level structural comparison with existing KPI-oriented approaches further evaluates maintainability, understandability, and extensibility. The results demonstrate that KPIforBlockchain provides a consistent and expressive foundation for structured blockchain performance evaluation, while AI-assisted modeling significantly enhances model generation quality. The proposed framework supports comparative analysis and informed decision-making, and lays the groundwork for future intelligent, automated blockchain evaluation mechanisms.

***Corresponding Author:** samimi@sku.ac.ir

Supplementary information: Supplementary information for this article is available at https://frai.journals.umz.ac.ir/

Please cite this paper as:

1. Introduction

Blockchain technology has emerged as a foundational paradigm for decentralized data management, enabling secure, transparent, and trustless transactions across distributed environments. By eliminating centralized intermediaries and relying on cryptographic mechanisms and distributed consensus, blockchain has redefined how digital trust is established in modern information systems. These characteristics have led to the widespread adoption of blockchain in diverse domains, including financial services, supply chains, healthcare, and large-scale cyber-physical systems [1]. As blockchain-based infrastructures increasingly intersect with intelligent and interconnected environments, such as Internet of Things (IoT) ecosystems, understanding and evaluating their operational performance has become a critical research concern.

At the core of many blockchain-enabled applications are smart contracts—self-executing programs that automate agreements and enforce rules without human intervention. Their performance and reliability directly impact overall system efficiency, particularly in environments with high transaction volumes and real-time constraints. As adoption scales in distributed and interconnected settings, the absence of systematic performance evaluation mechanisms increasingly undermines predictability, scalability, and long-term sustainability of blockchain-based applications [1].

ISSN 3092-7552

Unlike traditional centralized database systems, blockchain operates in a decentralized and peer-to-peer manner, introducing unique challenges for performance analysis. Network latency, node behavior, consensus delays, and dynamic participation of distributed entities can significantly influence system efficiency. These challenges are amplified in large-scale, heterogeneous environments, such as IoT systems, where resource constraints, network variability, and real-time responsiveness are critical [1]. Consequently, blockchain performance evaluation requires multi-dimensional analysis that goes beyond isolated metrics. With the rapid expansion of blockchain applications across domains such as banking, insurance, logistics, healthcare, and smart infrastructures, the identification and analysis of key performance indicators (KPIs) have become essential [2], [1]. Indicators such as throughput, latency, scalability, security, and cost provide measurable insight into system behavior; however, their interactions often introduce non-trivial trade-offs. For instance, improvements in throughput may adversely affect latency or energy consumption, a concern that becomes particularly critical in intelligent and connected environments where performance decisions directly influence adaptability and reliability.

Although existing studies have investigated specific performance aspects of blockchain systems, most approaches remain fragmented, focusing on isolated indicators or application-specific scenarios. There is still a lack of a comprehensive and structured framework capable of systematically capturing the complex relationships among diverse performance indicators in blockchain systems. This limitation limits stakeholders' ability to perform holistic performance analysis and make informed decisions, particularly in dynamic, evolving environments [3]. Metamodeling offers an effective mechanism for addressing this challenge by enabling explicit representation of concepts, relationships, and dependencies within complex systems. By formalizing performance indicators and their interconnections, a metamodeling approach supports systematic analysis across heterogeneous blockchain configurations and provides a foundation for advanced analytical and automated evaluation processes [3]. In blockchain-based systems, performance evaluation extends beyond purely technical parameters and encompasses network characteristics, user behavior, application logic, and governance policies. This complexity is further intensified in interconnected environments, where heterogeneous components interact dynamically, and performance conditions evolve over time [4]. As a result, blockchain performance analysis requires an integrated perspective that can accommodate both structural and operational dimensions. Accordingly, this work addresses the problem of identifying and structurally organizing key performance indicators in blockchain systems, with particular emphasis on their interrelationships. The objective is to design a comprehensive metamodel that captures these relationships and supports systematic performance evaluation and informed decision-making in the design, deployment, and optimization of blockchain systems.

The contribution of this study lies in providing a structured, extensible metamodel for blockchain performance evaluation that addresses the limitations of fragmented, indicator-specific approaches. By offering an integrated view of key performance indicators and their relationships, the proposed framework supports comparative analysis and lays the groundwork for future intelligent and automated evaluation mechanisms. This contribution is particularly relevant as blockchain systems increasingly operate in data-intensive, interconnected environments.

The remainder of this paper is organized as follows. Section 2 provides the background on blockchain systems, KPIs, and model-driven engineering concepts relevant to this study. Section 3 reviews related work on blockchain performance evaluation and analytical frameworks, highlighting existing limitations. Section 4 presents the proposed KPIforBlockchain methodology, including the metamodel design and the definition of computational indicators. Section 5 demonstrates the behavior and applicability of the proposed framework through representative case studies and analyzes the quality of large-language-model–assisted model generation. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Background

2.1. Blockchain Architecture and Performance Challenges

Blockchain systems are inherently decentralized infrastructures in which data processing and transaction validation are distributed across multiple network nodes. Unlike centralized architectures, blockchain relies on peer-to-peer communication, consensus mechanisms, and cryptographic verification to ensure data integrity and trustworthiness [1]. These architectural characteristics introduce distinct performance behaviors that are influenced by factors such as network latency, node synchronization, transaction validation processes, and smart contract execution. In practical deployments, blockchain performance is shaped by both technical design choices and dynamic operational conditions, including network scale, transaction load, and participant behavior [1]. As blockchain-based applications expand across domains, a holistic evaluation perspective becomes necessary, since isolating individual components fails to capture emergent system-level performance characteristics.

2.2. Blockchain Performance Indicators

Blockchain system performance is commonly assessed using key performance indicators (KPIs), which provide measurable insights into various aspects of system behavior [5]. Indicators such as throughput, latency, scalability, security, and operational cost are widely used to support comparison across platforms, configurations, and deployment scenarios [2], [1].

However, KPIs in blockchain systems are rarely independent. Improvements in one indicator may degrade another; for instance, enhancing security mechanisms may increase latency or computational cost. Such trade-offs complicate performance assessment, particularly in large-scale and interconnected environments. Consequently, effective performance evaluation requires not only identifying relevant KPIs but also understanding their interactions and combined effects on overall system behavior [2].

2.3. Structured KPI Representation

Although many studies identify and analyze individual blockchain performance indicators, merely enumerating KPIs is insufficient for comprehensive evaluation. Without a structured representation, dependencies among indicators remain implicit, limiting consistent comparison and systematic reasoning about performance trade-offs across alternative system configurations [3].

A structured representation of performance indicators enables systematic organization of KPIs and explicit identification of their relationships. By formalizing how indicators influence one another, such representations support more consistent analysis and facilitate comparative evaluation across different blockchain scenarios [3], [4]. This approach is particularly valuable in dynamic environments, where system conditions and performance requirements evolve over time.

Model-driven engineering (MDE) addresses complexity in software-intensive systems by elevating models to first-class artifacts throughout the system lifecycle. An explicit definition of abstract syntax, relationships, and transformation rules supports consistency, reusability, and automation—properties that are particularly valuable when analyzing the performance characteristics of complex, evolving blockchain-based systems [6].

Furthermore, domain-specific modeling and metamodeling techniques allow performance-related concepts to be formally captured and extended in a controlled manner [7]. By leveraging these principles, performance indicators can be organized, related, and evolved systematically, supporting comparative analysis across heterogeneous system configurations. This study adopts a model-driven approach to structure blockchain performance indicators and their interdependencies within a unified, extensible metamodel.

3. Related Work

3.1. Performance Evaluation and Code-Level Metrics in Smart Contracts

Several studies have evaluated the performance and quality of smart contracts using software metrics and empirical analysis. Tonelli et al. [8] investigated the lack of standardized software metrics for smart contracts by analyzing over 85,000 Ethereum contracts and comparing them with traditional software systems. Their results showed that smart contracts exhibit narrower ranges in metrics such as lines of code and structural complexity, due to constraints on blockchain resources. Similarly, Ajienka et al. [12] examined correlations between source code metrics and resource consumption, demonstrating that code complexity and the number of functions significantly impact gas usage and execution time. Pinna et al. [13] further analyzed large-scale Ethereum smart contract development practices, identifying common code patterns and vulnerabilities through extensive metric-based analysis. These studies collectively highlight the importance of code-level metrics in understanding smart contract performance and quality. However, they primarily focus on isolated metrics and do not provide a structured mechanism for linking them to broader system-level performance indicators.

3.2. Blockchain Performance, Consensus, and Benchmarking Frameworks

A number of works have addressed blockchain performance from a system-level perspective, with particular emphasis on consensus mechanisms, scalability, and benchmarking. Merrad et al. [11] proposed a comprehensive evaluation framework to analyze the performance trade-offs of various consensus algorithms, offering quantitative insights into their scalability and security implications. Thakkar et al. [22] focused on performance optimization and benchmarking of Hyperledger Fabric, identifying improvements in throughput and latency through systematic evaluation. Similarly, Hao et al. [23] and Rouhani et al. [26] analyzed the performance characteristics of consensus algorithms and Ethereum transactions, respectively, highlighting variability across configurations and platforms. Other studies evaluated blockchain platforms under different workloads and deployment scenarios. Pongnumkul et al. [27] compared the performance of private blockchain platforms such as Hyperledger Fabric and Ethereum, while Fan et al. [30] conducted a systematic review of blockchain performance evaluation methodologies, identifying inconsistencies in metrics, benchmarks, and evaluation practices. Although these approaches provide valuable benchmarking insights, they generally treat performance indicators independently and lack explicit modeling of relationships among key indicators.

3.3. Blockchain Integration with IoT and Industrial Systems

Blockchain integration with IoT and industrial environments has been extensively studied, particularly with respect to performance, security, and resource constraints. Alaslani et al. [14] evaluated end-to-end latency in blockchain-based IoT systems through simulation, showing that blockchain configuration parameters significantly influence system latency. Alrubei et al. [15] and Cui et al. [16] explored the use of Ethereum and distributed ledger technologies to enhance security and privacy in IoT systems, noting performance overhead and latency as key challenges. In industrial contexts, Huang et al. [17] proposed a reputation-based consensus mechanism for Industrial IoT to improve data integrity and honest participation, while Cao et al. [18] and Liu et al. [19] introduced lightweight consensus and blockchain architectures to address the resource limitations of IoT and IIoT devices. Novo [20] focused on access management in IoT systems using blockchain, emphasizing scalability and security. Jiang et al. [24] extended these ideas to the Internet of Vehicles (IoV), proposing a layered blockchain architecture to improve data security and fault tolerance, albeit at the cost of increased latency. These studies demonstrate blockchain's potential in IoT-driven environments but largely focus on specific performance aspects or architectural optimizations rather than comprehensive performance evaluation frameworks.

3.4. Application-Specific and Decision-Oriented Performance Studies

Several studies investigated blockchain performance in specific application domains and decision-making contexts. Bovenzi et al. [9] analyzed the performance limitations of blockchain in Industry 4.0 applications, identifying consensus mechanisms, network architecture, and data storage as key performance factors. Kuhi et al. [21] and Madhwal et al. [10] proposed blockchain-based solutions for supply chain management and logistics, emphasizing transparency, traceability, and delivery performance. Meng et al. [28] extended this line of work by integrating blockchain with machine learning to predict delivery performance, introducing novel reliability metrics while highlighting challenges related to data quality and synchronization.

Other decision-oriented approaches include Hayat et al. [29], who applied the Analytic Hierarchy Process (AHP) to select suitable blockchain platforms for product lifecycle management, and Grida et al. [2], who identified critical success factors for blockchain adoption through expert surveys. While these approaches support informed decision-making, they rely on fixed criteria or subjective assessments and do not systematically model relationships among performance indicators.

3.5. Identified Research Gaps

Based on the reviewed literature [8]–[30], existing studies predominantly focus on specific performance indicators, application scenarios, or architectural components of blockchain systems. Although valuable insights have been reported on smart contract metrics, consensus mechanisms, IoT integration, and application-level performance, there is still no comprehensive, systematic approach for modeling and analyzing the relationships among key performance indicators. The lack of such structured representation limits holistic performance evaluation and hinders consistent comparison across blockchain systems. Table 1 summarizes the primary focus and evaluation scope of existing blockchain performance studies, highlighting the fragmented nature of current approaches. These gaps motivate the need for an integrated framework that captures both performance indicators and their interdependencies.

Table 1. Focus of Existing Blockchain Performance Studies.

References	Primary Focus	Evaluation Scope
[8], [12], [13]	Smart contract metrics	Code-level KPIs
[11], [22], [23], [26], [27], [30]	System performance & benchmarking	Platform-specific
[14]–[20], [24]	Blockchain–IoT integration	Latency, security
[9], [10], [21], [28], [29], [2]	Application & decision support	Domain-dependent
This work	Integrated KPI evaluation	Cross-KPI analysis

Table 2 further compares prior studies with respect to KPI coverage and the explicit modeling of relationships among performance indicators.

Table 2. KPI Coverage and Relationship Modeling in Prior Studies.

References	KPI Coverage	Explicit KPI Relationships
[8], [12], [13]	Limited	No
[11], [22], [23], [26], [27]	Multiple	No
[14]–[20], [24]	Application-specific	No
[28], [29], [2]	Decision-oriented	Partial
This work	Comprehensive	Yes

4. Methodology

This methodology follows a model-driven, conceptual analysis approach to structure blockchain performance evaluation using key performance indicators (KPIs). Rather than relying on empirical benchmarking or platform-specific measurements, the approach constructs an abstract and extensible evaluation framework that supports comparative analysis, analytical reasoning, and decision-making across heterogeneous blockchain systems. The methodology integrates research design principles, metamodel conceptualization, structural organization of model elements, and formal definition of computational KPI functions.

4.1. Research Design and Modeling Approach

A model-driven conceptual design is employed to address the complexity and multidimensionality of blockchain performance evaluation. Due to the decentralized, heterogeneous, and evolving nature of blockchain systems, evaluation approaches based on isolated metrics or platform-specific benchmarks are insufficient for capturing interdependencies among performance factors. Metamodeling is therefore used as a formal mechanism to organize, relate, and analyze blockchain performance indicators within a unified analytical framework.

Metamodeling enables the abstraction of domain concepts, relationships, and constraints at a higher level, independent of specific blockchain implementations. This approach is particularly suitable for performance evaluation in complex systems, as it facilitates consistency, extensibility, and reusability of evaluation criteria across different contexts. In this research, the metamodel serves as a conceptual backbone that links measurable blockchain features to analytical performance categories, enabling systematic reasoning about performance trade-offs and system behavior. The methodological process consists of the following steps:

1. Identifying relevant blockchain performance dimensions and KPIs based on prior studies;
2. Defining core blockchain feature sets that influence these indicators;
3. Organizing KPIs into abstract and hierarchical categories; and
4. Establishing conceptual associations between blockchain features and performance indicators.

This structured approach ensures that the resulting metamodel captures both the structural and operational aspects of blockchain systems while remaining adaptable to emerging technologies and application domains.

4.2. Conceptual Design of the Blockchain KPI Metamodel

The proposed metamodel provides a holistic representation of blockchain performance evaluation criteria by explicitly modeling relationships between blockchain features and key performance indicators. Instead of treating KPIs as isolated metrics, the model captures their interdependencies and contextual relevance within the broader blockchain ecosystem, enabling structured and comparative performance analysis. At the conceptual level, the metamodel distinguishes between two fundamental layers:

- blockchain feature representations, which describe measurable characteristics of the system, and
- KPI categories, which interpret these characteristics from performance, security, scalability, cost, and decentralization perspectives.

This separation allows performance indicators to be evaluated based on concrete system attributes while maintaining analytical abstraction. To ensure consistency and standardization, the metamodel includes a set of enumeration types that define commonly used blockchain characteristics, including consensus mechanisms, transaction types, cryptographic primitives, and qualitative scales for security strength and temporal properties. These enumerations do not introduce computational logic themselves; instead, they provide controlled vocabularies that support uniform interpretation of blockchain features across different model instances. Extensibility and generality are central design principles of the proposed metamodel, allowing new blockchain features, KPIs, and evaluation dimensions to be incorporated without modifying the core structure. This design choice supports adaptation to emerging blockchain architectures and evolving application requirements, particularly in intelligent and interconnected environments.

4.3. Structure of the Proposed Metamodel

The blockchain KPI metamodel follows a layered and modular structure that organizes measurable blockchain characteristics and relates them to abstract performance evaluation criteria. This feature-centric architecture enables a clear separation between observable system properties and their analytical interpretation, supporting systematic, consistent evaluation across heterogeneous blockchain configurations. At the core of the metamodel lies the FeatureModel class, which functions as the central aggregation point for all blockchain-related features. This class provides a unified container that integrates different feature domains, ensuring consistency and coherence across the evaluation process. By acting as the root of the model, the FeatureModel allows heterogeneous blockchain configurations to be represented within a common analytical structure.

4.3.1.Blockchain Feature Layer

The first structural layer of the metamodel captures blockchain characteristics through a set of specialized feature classes. These classes represent different aspects of a blockchain system that directly or indirectly influence performance evaluation. Specifically, the metamodel distinguishes between block-level features, transaction-related features, network characteristics, application behavior, consensus mechanisms, and external environmental factors.

- **BlockFeature** models properties associated with block creation and structure, enabling the analysis of block generation behavior, resource consumption, and block-level efficiency.
- **TransactionSetFeature** and **TransactionFeature** capture both aggregated and individual transaction characteristics, supporting the evaluation of transaction throughput, fees, confirmation behavior, and execution dynamics.
- **NetworkFeature** represents network-level attributes such as node distribution, connectivity, latency, and cryptographic mechanisms, which are critical for understanding communication efficiency and fault tolerance.
- **ApplicationFeature** focuses on blockchain applications, particularly smart contracts, enabling analysis of execution cost, invocation frequency, and computational overhead.
- **ConsensusFeature** models consensus-related properties, including validator behavior, consensus

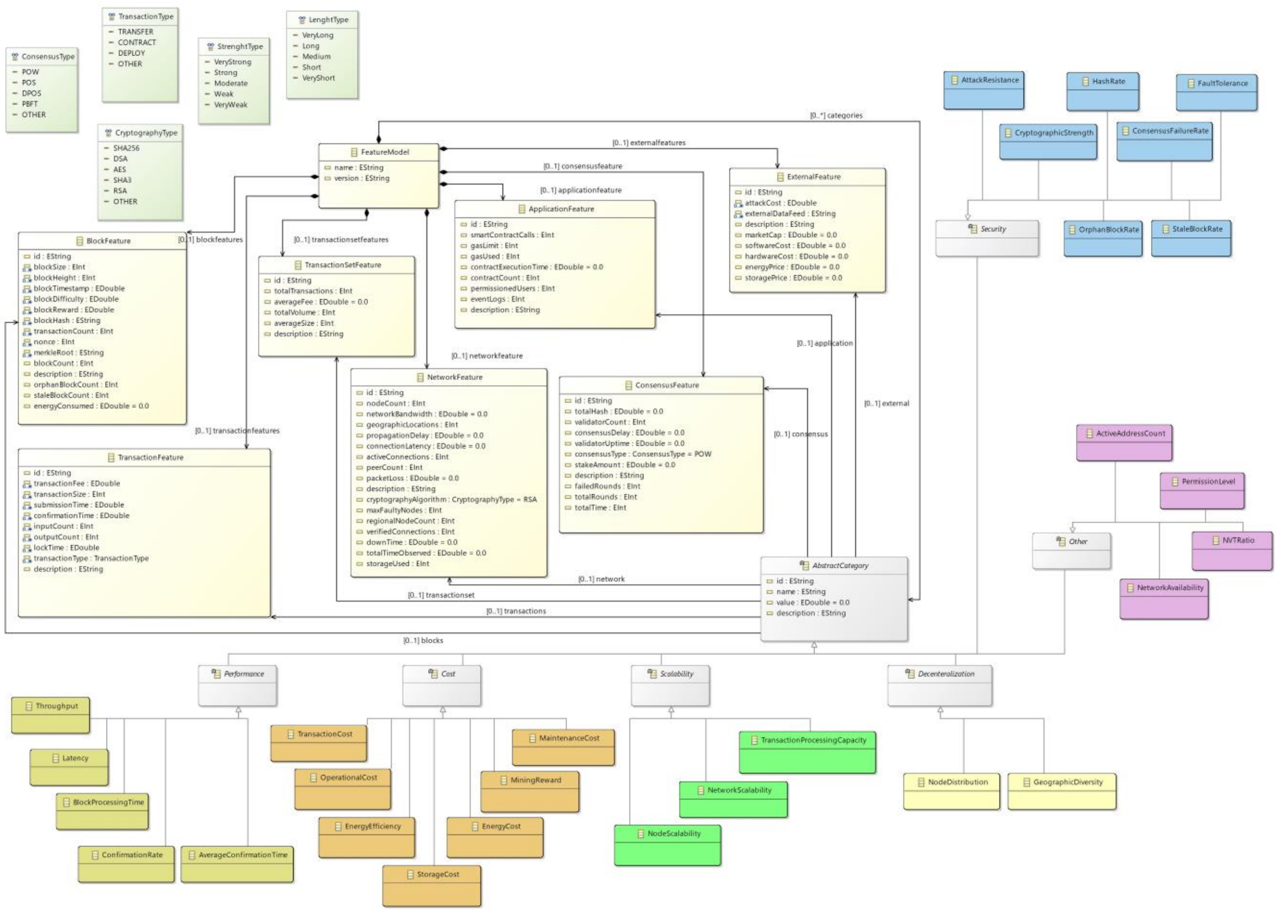

Figure 1. The proposed Metamodel called KPIforBlockchain for Evaluation of blockchains.

delays, and participation dynamics, which are central to both performance and security evaluation.

- **ExternalFeature** accounts for environmental and economic factors—such as energy prices, hardware costs, and attack costs—that influence blockchain operations but originate outside the protocol itself.

These feature classes are connected to the FeatureModel through composition relationships, emphasizing that they collectively define a single blockchain system instance. Enumeration data types, such as consensus mechanisms, transaction types, cryptographic algorithms, and qualitative scales, are employed to standardize feature values and improve interpretability across different blockchain platforms.

4.3.2. KPI Categorization Layer

The second structural layer of the metamodel organizes key performance indicators through the AbstractCategory class. This class provides a generic structure for defining KPIs, including identifiers, descriptive attributes, and evaluative values. By abstracting KPI representation, the metamodel avoids hard-coding metrics into specific feature classes and enables flexible, reusable performance definitions.

Six primary KPI categories inherit from the AbstractCategory class: Performance, Security, Scalability, Cost, Decentralization, and Other. Each category groups conceptually related indicators while maintaining associations with relevant blockchain feature classes. This design allows KPIs to be interpreted in relation to multiple blockchain characteristics rather than a single isolated attribute. Concrete KPI classes—such as throughput, latency, confirmation time, attack resistance, energy efficiency, transaction cost, network scalability, and geographic diversity—are defined as subclasses of their respective abstract categories. These KPI classes do not introduce new structural elements; instead, they inherit the generic evaluation structure and rely on associated blockchain features for their computation and interpretation. This approach supports consistency in KPI definition while allowing diverse indicators to coexist within a unified evaluation framework.

4.3.3. Structural Rationale

The proposed metamodel is intentionally modular, extensible, and implementation-independent. By decoupling blockchain features from performance interpretation and organizing KPIs through abstract categories, the model enables systematic comparison of different blockchain systems and supports the introduction of new indicators without restructuring the entire framework. This design choice is particularly important for emerging and evolving blockchain environments, where performance requirements and evaluation priorities may change over time.

4.4. Assessing the Quality of Model-Based Results

To facilitate structured interpretation of blockchain performance indicators, this study introduces a *model-driven quality assessment schema* derived directly from the proposed meta-model. The objective of this assessment is not to benchmark or empirically rank blockchain platforms, but rather to demonstrate how the meta-model can systematically support the definition, organization, and interpretation of key performance indicators (KPIs). The assessment schema operationalizes the meta-model by mapping abstract KPI categories to concrete indicators and associated blockchain feature layers. This mapping provides traceability between observable blockchain characteristics and higher-level evaluation dimensions such as performance, cost, scalability, decentralization, and security. As a result, the framework enables transparent and explainable reasoning about blockchain behavior across heterogeneous systems.

4.4.1. Overview of KPI Categories and Indicators

Table 3 presents an overview of the main KPI categories and their associated indicators as defined in the proposed metamodel. Each indicator is explicitly linked to the blockchain feature layers from which it derives its evaluative meaning. This structural alignment ensures that KPIs are not treated as isolated metrics but as analytical constructs grounded in measurable system properties.

To operationalize these indicators, the proposed framework defines their computational logic at the model level using the Epsilon Object Language (EOL). Rather than relying on runtime measurements or platform-specific instrumentation, each KPI is computed directly from metamodel-conformant model elements and their associated attributes. This approach enables automated, repeatable, and implementation-independent evaluation of blockchain characteristics based on structured models.

```
operation Throughput computeThroughput() : Real {
 var min_time = KPIforBlockchain!TransactionFeature.
allInstances.first.submissionTime.min();
 var max_time = KPIforBlockchain!TransactionFeature.
allInstances.first.submissionTime.max();
 var total_time = max_time-min_time;
 var result = 0.0;
  if (total_time > 0) {
      result = self.transactionset.totalTransactions / total_time;
}}
```

This example illustrates how performance indicators are formally defined and evaluated within the proposed meta-model, enabling systematic analysis without assuming execution-level data or empirical benchmarking.

Table 3. Blockchain KPI Categories and Indicators Derived from the Proposed Metamodel.

KPI Category	Indicator	Related Feature Layer(s)
Performance	Throughput	TransactionSetFeature, NetworkFeature
	Latency	TransactionFeature, NetworkFeature
	Block Processing Time	BlockFeature
	Confirmation Rate	TransactionFeature
Cost	Transaction Cost	TransactionFeature
	Operational Cost	ExternalFeature, NetworkFeature
	Energy Efficiency	BlockFeature, ExternalFeature
	Storage Cost	NetworkFeature
Scalability	Network Scalability	NetworkFeature
	| Node Scalability	NetworkFeature
	Transaction Processing Capacity	BlockFeature, TransactionFeature
Decentralization	Node Distribution	NetworkFeature
	Geographic Diversity	NetworkFeature
Security	Attack Resistance	ConsensusFeature, ExternalFeature
	Orphan Block Rate	BlockFeature
	Stale Block Rate	BlockFeature
	Consensus Failure Rate	ConsensusFeature
	Cryptographic Strength	NetworkFeature
Other	Network Availability	NetworkFeature
	Active Address Count	TransactionSetFeature
	NVT Ratio	ExternalFeature

4.4.2. Quality Interpretation and Illustrative Computational Criteria

Table 4 summarizes the complete set of key performance indicators supported by the proposed metamodel. The indicators are grouped conceptually into performance, cost, scalability, decentralization, security, and auxiliary metrics; however, the category column is intentionally omitted to improve readability.

The qualitative levels (Good, Medium, Weak) represent illustrative evaluation ranges and are not intended as universal benchmarks. Instead, they demonstrate how domain-specific thresholds can be instantiated within the meta-model to support structured and explainable analysis.

Table 4. Illustrative Quality Criteria and Computational Interpretation of Blockchain KPIs.

Indicator	Explanation	Quality Criteria (G / M / W)	Formula

Throughput	Transactions processed per unit time	G: >1000 M: 100–1000 W: <100	$\frac{Total\ Transactions}{Elapsed\ Time}$
Latency	Avg. time from submission to confirmation	G: <1 M: 1–10 W: >10	$\frac{\sum(t_{confirmation} - t_{submission})}{count\ of\ time\ interval}$
Block Processing Time	Avg. time to generate a block	G: <10 M: 10–60 W: >60	$\frac{\sum BTstamp_i - BTstamp_{i-1}}{blockCount}$
Confirmation Rate	Ratio of confirmed transactions	G: >95% M: 80–95% W: <80%	$\frac{\sum Confirmed\ Transactions}{Total\ Transactions}$
Average Confirmation Time	Avg. confirmation duration	G: <10 M: 10–30 W: >30	$\frac{\sum Confirmation\ time}{Confirmation\ count}$
Maintenance Cost	Software + hardware cost	G: <100k M: 100k–500k W: >500k	$Software\ Cost + Hardware\ Cost$
Energy Efficiency	Transactions per energy unit	G: >1000 M: 100–1000 W: <100	$\frac{Total\ Transactions}{Energy\ Consumed}$
Energy Cost	Total energy expenditure	G: <10k M: 10k–50k W: >50k	$energyConsumed \times energyPrice$
Transaction Cost	Avg. cost per transaction	G: <100 M: 100–1000 W: >1000	$\frac{Gas\ Used}{Total\ Transactions}$
Operational Cost	Transaction + maintenance cost	G: <1k M: 1k–10k W: >10k	$Transaction\ Cost + Maintenance\ Cost$
Mining Reward	Avg. mining incentive	G: >100 M: 1–100 W: <1	$\frac{\sum_{Blocks} Difficulty \times Reward}{Mined\ Block\ Count}$
Storage Cost	Data storage expenditure	G: <10k M: 10k–50k W: >50k	$Used\ Storage \times Storage\ Price$
Network Scalability	Tx to bandwidth ratio	G: >0.1 M: 0.01–0.1 W: <0.01	$\frac{Troughput}{Network\ Bandwidth}$
Node Scalability	Tx per node	G: >10 M: 1–10 W: <1	$\frac{Troughput}{Node\ Counts}$
Processing Capacity	Max tx per second	G: >1000 M: 100–1000 W: <100	$\frac{Maximum\ Block\ Size}{Avg.Transaction\ Size \times BT}$
Node Distribution	Regional node ratio	G: >0.1 M: 0.01–0.1 W: <0.01	$\frac{Regional\ Node\ Counts}{Node\ Counts}$
Geographic Diversity	Geo locations per node	G: >0.1 M: 0.01–0.1 W: <0.01	$\frac{Geographical\ Locations}{Node\ Counts}$
Attack Resistance	Cost × hash power	G: >10^9 M: 10^7–10^9 W: <10^7	$Hardware\ Cost \times Total\ Hash$
Orphan Block Rate	Orphan block ratio	G: <0.01 M: 0.01–0.05 W: >0.05	$\frac{Orphan\ Block\ Count}{Block\ Count}$
Stale Block Rate	Stale block ratio	G: <0.01 M: 0.01–0.05 W: >0.05	$\frac{Stale\ Block\ Count}{Block\ Count}$
Consensus Failure Rate	Failed rounds ratio	G: <0.01 M: 0.01–0.1 W: >0.1	$\frac{Failed\ Rounds}{Total\ Rounds}$
Cryptographic Strength	Algorithm security score	G: 4–5 M: 2–3 W: 0–1	Algorithm Score, SHA256/SHA3: 5, AES: 4, DES: 3, RSA: 2, Unknown: 0
Hash Rate	Hashes per second	G: >1000 M: 100–1000 W: <100	$\frac{Total\ hash}{Total\ time}$
Fault Tolerance	Faulty node ratio	G: <0.1 M: 0.1–0.33 W: >0.33	$\frac{Maximum\ Tolerable\ Faulty\ Nodes}{Node\ Count}$
NVT Ratio	Market value / volume	G: <50 M: 50–100 W: >100	$\frac{Market\ Capitalization}{Total\ Volume}$
Active Address Count	Active address ratio	G: >50 M: 10–50 W: <10	$\frac{Active\ Connections\ Count}{Verified\ Connections\ Count}$
Permission Level	Consensus openness	G: 1 M: 2 W: 0	Consensus Score: POS/POW: 1 DPOS/PBFT: 2, Unknown: 0.
Network Availability	Active time ratio	G: >0.98 M: 0.95–0.98 W: <0.95	$((Total\ Network\ Monitoring\ Time - Network\ Monitoring\ Downtime))/(Total\ Network\ Monitoring\ Time)$

Performance-related indicators (e.g., throughput, latency, block processing time) reflect the system's operational efficiency and responsiveness. Cost-related indicators capture economic sustainability from operational, energy, and storage perspectives. Scalability indicators assess the system's ability to accommodate increasing load and network growth. Decentralization and security indicators focus on resilience, fault tolerance, and resistance to attacks. Finally, auxiliary indicators such as NVT ratio and network availability provide complementary insights into economic soundness and system reliability.

4.4.3. Methodological Significance

By integrating KPI categorization, feature traceability, and quality interpretation within a unified meta-model, the proposed approach provides a systematic foundation for blockchain performance evaluation. This framework supports analytical comparison, design-space exploration, and informed decision-making without relying on platform-specific assumptions or empirical benchmarking data. Moreover, the modular structure of the assessment schema enables future extensions, such as incorporating intelligent analytics, adaptive weighting mechanisms, or AI-driven performance reasoning—making it particularly suitable for evolving blockchain-enabled IoT and cyber-physical systems.

5. Evaluation

5.1. Evaluation Objectives and Methodology

This evaluation focuses on assessing the applicability, consistency, and analytical capability of the proposed KPIforBlockchain framework, rather than benchmarking real-world blockchain deployments. Accordingly, the evaluation adopts a demonstration-oriented approach that examines how the proposed metamodel, computational indicators, and LLM-assisted model generation process interact within representative blockchain scenarios.

Specifically, the evaluation aims to address three key aspects:

1. The ability of the proposed metamodel to structurally represent diverse blockchain features and key performance indicators in a unified and extensible manner;
2. The correctness and completeness of blockchain models generated with the assistance of a large language model under different learning settings; and
3. The behavior of the proposed key performance indicators when applied to multiple blockchain configurations, enabling comparative and multi-dimensional analysis.

A set of case-study-based blockchain models is defined and instantiated using the proposed metamodel to support this evaluation. These models are not intended to reflect measured runtime data from deployed systems; instead, they serve to demonstrate the expressive capacity of the metamodel and the analytical capabilities of the framework. Key performance indicators are computed using formally defined formulas, and their outcomes are analyzed comparatively across different blockchain configurations. In addition, the quality of the LLM-assisted model generation process is evaluated using standard information retrieval metrics, including precision, recall, and F1-score, by comparing generated models with expert-validated reference models. Finally, a structural comparison between the proposed metamodel and existing KPI-oriented metamodels is conducted using established software design metrics to assess maintainability, understandability, and extensibility.

5.2. Case Study–Based Framework Demonstration

To demonstrate the analytical behavior and applicability of the proposed KPIforBlockchain framework, a set of representative case-study blockchain models is defined and instantiated using the proposed metamodel. These case studies are intentionally designed to cover diverse blockchain configurations by varying key structural and operational characteristics, including block properties, transaction sets, network attributes, consensus mechanisms, application-level features, and external influencing factors. The purpose of this demonstration is not to report empirical measurements from deployed blockchain systems, but to illustrate how the proposed framework operates across different representative configurations. Each case study is modeled as a structured XMI instance conforming to the proposed metamodel. The resulting models include instances of core feature classes such as *BlockFeature*, *TransactionFeature*, *TransactionSetFeature*, *NetworkFeature*, *ApplicationFeature*, *ConsensusFeature*, and *ExternalFeature*, as well as instances of KPI classes that inherit from the abstract KPI categorization hierarchy. Figure 2 illustrates the tree-view representation of one representative case-study model, highlighting the hierarchical organization of features and indicators within the proposed framework. Based on these models, the computational formulas defined for each key performance indicator are executed to illustrate the analytical behavior of the framework. The resulting indicator values for the selected blockchain configurations are reported in Table 5 and should be interpreted as illustrative analytical outcomes that reflect variations in modeled blockchain characteristics rather than benchmark measurements from real-world deployments.

To improve readability, the results are further analyzed through normalized visual comparisons across six KPI categories. The performance-related indicators, including throughput, latency, block processing time, confirmation rate, and average confirmation time, illustrate how the framework captures differences in transaction-handling capacity and consensus responsiveness across configurations. Similarly, cost-related indicators—such as maintenance cost, transaction cost, operational cost, energy efficiency, energy cost, mining reward, and storage cost—demonstrate the framework's ability to model and analyze economic aspects of blockchain systems in a structured manner.

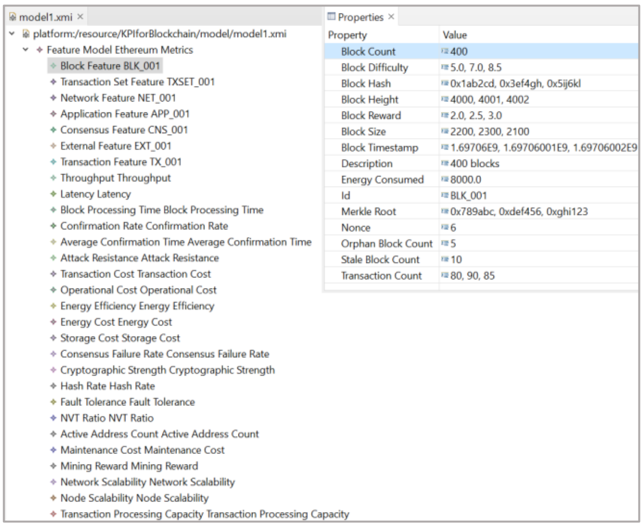

Figure 2. Tree view of an example model.

Table 5. Illustrative model-derived KPI values (not empirical benchmarks).

KPI	Model 1	Model 2	Model 3	Model 4	Model 5
Throughput	55000.0	42500.0	45000.0	47500.0	31000.0
Latency	1.6887E9	1.6927E9	1.690E9	1.677E9	1.697E9
Block Processing Time	0.0022	0.005	0.006	0.0042	0.006
Confirmation Rate	3.75E-5	3.334E-5	3.34E-5	3.1578948 E-5	3.0E-5
Average Confirmation Time	1.6970601 06E9	1.6970606 067E9	1.6970611 06E9	1.6970616 067E9	1.6970701 06E9
Maintenance Cost	14000.0	13300.0	16000.0	14400.0	16000.0
Energy Efficiency	10.0	16.3636	18.0	19.0	8.3333
Energy Cost	400000.0	264000.0	275000.0	260000.0	720000.0
Transaction Cost	31	24	33	27	11
Operational Cost	14031.0	13324.0	16033.0	14427.0	16000.0
Mining Reward	17.6666	14.0166	28.0	12.8735	69.7917
Storage Cost	7500.0	6160.0	10500.0	8680.0	10500.0
Network Scalability	8.8888	10.0	9.0	8.6363	10.0
Node Scalability	114.2857	128.5714	112.5	105.5555	125.0
Transaction Processing Capacity	2.9462E-9	2.9462E-9	3.5355E-9	2.9462 E-9	2.9462E-9
Node Distribution	0.0428	0.0285	0.045	0.0355	0.05
Geographic Diversity	0.0285	0.0257	0.0225	0.0244	0.03
Resistance to 51% Attack	3.15E7	2.72E7	4.0E7	3.312E7	5.0E7
Orphan Block Rate	0.0125	0.0066	0.008	0.0042	0.016
Stale Block Rate	0.025	0.0116	0.014	0.0085	0.03
Consensus Failure Rate	0.04	0.0392	0.0428	0.0281	0.0375
Cryptographic Algorithm	SHA3	SHA256	SHA256	SHA256	SHA256
Cryptographic Score	5	5	5	5	5
Hash Rate	53.75	42.6666	48.4444	45.3529	40.0
Fault Tolerance	0.0171	0.0114	0.0175	0.0111	0.02
NVT Ratio	1.3333E-5	1.3846E-5	1.3888E-5	1.6296E-5	1.5E-5
Active Address Count	25%	25.63%	24.5%	18.5%	27.58%
Permission Level	Open participati on	Open participati on	Open participati on	Open participati on	Open participati on
Permission Level Score	1	1	1	1	1

KPI	Model 1	Model 2	Model 3	Model 4	Model 5
Network Uptime	0.95	0.9493	0.9777	0.9548	0.95

Scalability indicators, including network scalability, node scalability, and transaction processing capacity, further show how the framework supports comparative analysis of scaling behavior under different modeling assumptions. Decentralization-related indicators, such as node distribution and geographic diversity, capture structural dispersion characteristics that are essential for analyzing resilience and fault tolerance. Security-related indicators—including resistance to majority attacks, orphan and stale block rates, consensus failure rate, cryptographic strength, hash rate, and fault tolerance—provide a comprehensive view of modeled security properties within the unified framework. In addition, auxiliary indicators such as NVT ratio, active address count, permission level, and network availability extend the analysis toward economic activity, user participation, access control, and system reliability. The comparative visualization of selected indicator groups is presented in Figures 3-8, which further illustrate how the proposed framework enables multi-dimensional analysis across representative blockchain configurations.

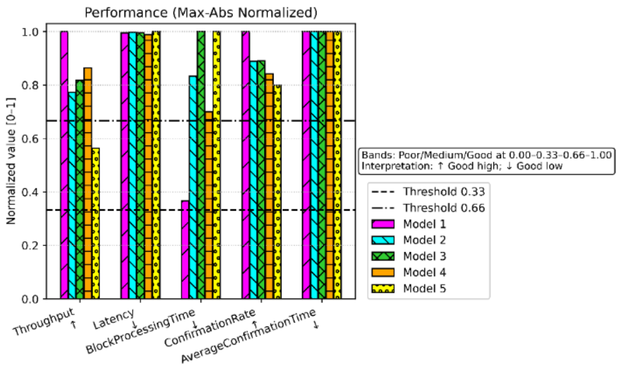

Figure 3. Normalized performance indicators across representative blockchain configurations.

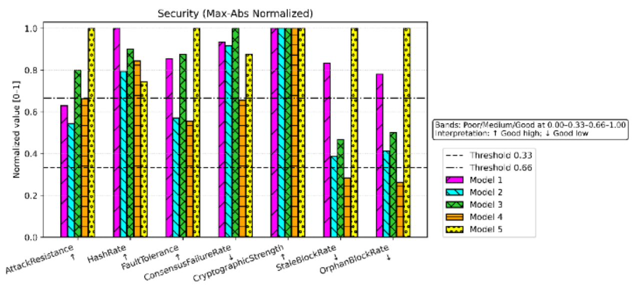

Figure 4. Security-related indicator comparison for representative blockchain models.

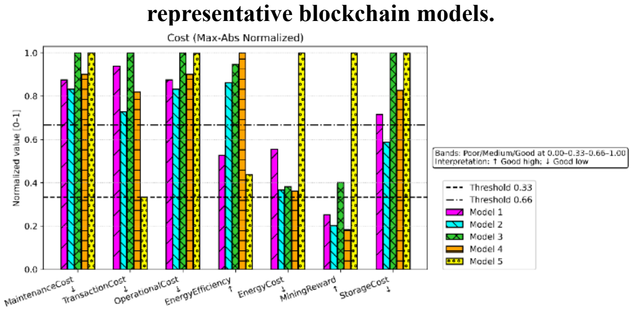

Figure 5. Cost-related indicator analysis across representative blockchain configurations.

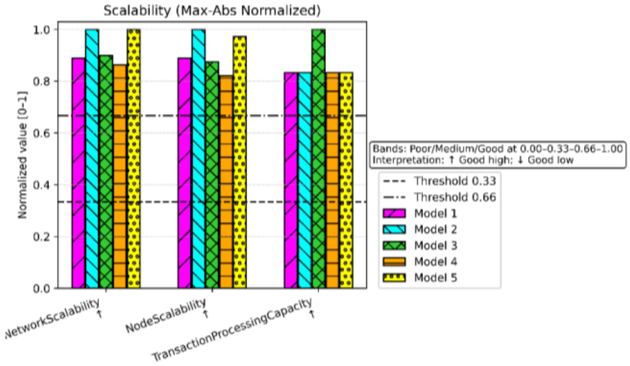

Figure 6. Scalability indicator comparison for representative blockchain models.

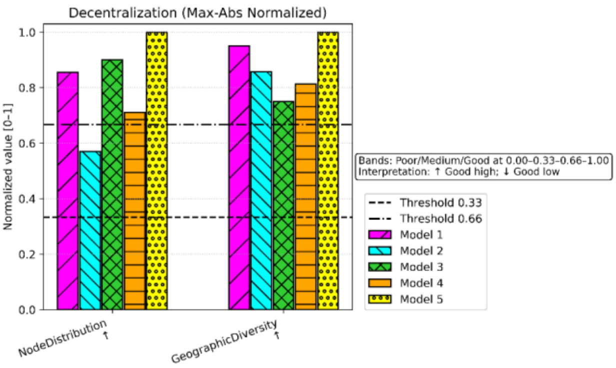

Figure 7. Decentralization-related indicators for representative blockchain configurations.

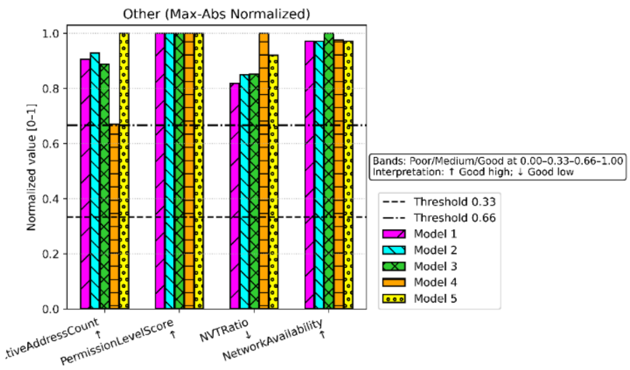

Figure 8. Additional indicators (NVT ratio, activity, permission level, and availability) across blockchain models.

Overall, this case-study–based demonstration shows that the proposed KPIforBlockchain framework can consistently model, compute, and organize a wide range of blockchain indicators. The presented results highlight the framework's ability to support structured analytical reasoning and comparative evaluation at the model level, rather than empirical performance benchmarking.

5.3. LLM-Assisted Model Generation Analysis

This subsection analyzes the effectiveness of large language model–assisted generation of blockchain models within the proposed KPIforBlockchain framework. The focus of this evaluation is on assessing the quality of the model generation process itself—specifically, the structural correctness and completeness of generated models—rather than on evaluating blockchain runtime behavior or system-level performance. In this study, the Grok 3 large language model developed by xAI is employed to generate initial blockchain model instances in structured XMI format. The language model serves as an intelligent assistant that generates candidate model structures from textual descriptions of blockchain characteristics. These generated models are subsequently reviewed and refined by domain experts to ensure compliance with the proposed metamodel and semantic consistency. The expert-refined models serve as reference models for evaluation.

To investigate the impact of different levels of guidance on generation quality, three learning configurations are considered: zero-shot, one-shot, and few-shot learning. In the zero-shot configuration, the language model generates blockchain models without being provided with any example instances. In the one-shot configuration, a single example model is supplied to guide the generation process. In the few-shot configuration, multiple example models are provided, enabling the model to better capture recurring structural patterns and relationships defined by the metamodel. The quality of the generated models is evaluated by comparing them against the corresponding expert-completed reference models. This comparison focuses on the presence and correctness of metamodel elements, including classes, attributes, and relationships. Standard evaluation metrics—precision, recall, and F1-score—are used to quantify generation accuracy and completeness. Precision measures the proportion of correctly generated elements among all generated elements, while recall measures the proportion of reference elements successfully reproduced by the language model. The F1-score provides a balanced measure of both aspects. The quantitative results of this analysis are summarized in Table 6. The results indicate a consistent improvement in generation quality as the level of guidance increases. While the zero-shot configuration achieves moderate precision and recall, the one-shot configuration shows notable improvement, and the few-shot configuration yields the highest precision, recall, and F1 score. This trend indicates that example-based prompting significantly enhances the reliability of LLM-assisted model generation in structured modeling tasks.

Table 6. Comparison of large language model performance in different modes based on evaluation metrics.

Mode	Precision	Recall	F1 Score
No-Shot	81.5%	83.2%	82.3%
One-Shot	89.2%	90.1%	89.6%
Few-Shot	95.7%	96.6%	96.1%

Overall, this analysis demonstrates that large language models can effectively support model-driven blockchain analysis when combined with a well-defined metamodel and expert validation. The findings emphasize the role of AI-assisted modeling as a complementary technique within model-driven engineering workflows, aligning with current research directions in artificial intelligence–enabled system modeling and analysis.

5.4. Metamodel-Level Structural Comparison

This subsection presents a structural comparison between the proposed KPIforBlockchain metamodel and existing metamodels that support performance and KPI analysis in related domains. The objective of this comparison is not to assess functional superiority, but to analyze structural quality attributes at the metamodel level, namely maintainability [34], understandability [35], and extensibility [36].

The comparison is conducted using established software design metrics commonly employed in model-driven engineering studies. These metrics capture fundamental structural properties of a metamodel, including size, complexity, inheritance structure, and coupling characteristics. Specifically, the following basic metrics are used: number of classes (NC), number of attributes (NA), number of relationships (NR), maximum depth of inheritance hierarchy (DITmax), maximum fan-out (Fanoutmax), number of predecessor nodes in the inheritance hierarchy (PRED), number of inherited features (INHF), and total number of features (NTF). To derive high-level quality attributes from the basic structural metrics, three composite measures are employed: maintainability, understandability, and extensibility. These measures are calculated using established formulations from model-driven engineering literature. Maintainability [34] is computed as the average of selected size and complexity metrics, reflecting the metamodel's overall structural complexity. Higher maintainability values indicate increased structural complexity and, consequently, lower ease of maintenance.

$$Maintenability = \frac{NC + NA + NR + DIT_{max} + Fanout_{max}}{5} \quad (1)$$

Understandability [35] is defined based on the average number of predecessor nodes in the inheritance hierarchy, capturing the conceptual clarity of the model structure. Higher values of understandability indicate a clearer, more comprehensible inheritance structure.

$$Understandability = \frac{\sum_{k=1}^{NC}(PRED_k + 1)}{NC} \quad (2)$$

Extensibility [36] measures the metamodel's potential to accommodate future extensions through inheritance and reuse mechanisms. Higher extensibility values indicate greater flexibility for model evolution and extension.

$$Extendibility = \frac{INHF}{NTF} \quad (3)$$

Table 7 summarizes the values of the basic structural metrics and the derived quality measures for the proposed KPIforBlockchain metamodel and the reference metamodels. The results indicate that the proposed metamodel achieves a balance between expressive power and structural complexity. While its maintainability value is higher due to the larger number of classes, attributes, and relationships, this complexity supports a richer representation of blockchain features and KPIs. At the same time, the metamodel exhibits strong understandability and extensibility, reflecting a well-structured inheritance hierarchy and effective reuse of abstract elements.

Table 7. Structural design metrics and derived quality measures for comparing KPIforBlockchain with existing metamodels.

Metric	Our metamodel	MIKADO [31]	MDA-KPI [32]	Customized KPI [33]
Number of Classes (NC)	43	41	43	11
Number of Attributes (NA)	83	53	30	33
Number of Relationships (NR)	42	48	54	12
Maximum Hierarchy Depth (DITmax)	2	4	5	3
Maximum Fanout (Fanoutmax)	7	3	12	4
Predecessor Nodes (PRED)	2	4	4	2
Inherited Features (INHF)	374	53	20	54
Total Number of Features (NTF)	90	59	53	39
Maintainability	35.4	30.2	28.8	12.6
Understandability	2.232	1.78	1.348	1.09
Extensibility	4.155	0.898	0.377	1.384

To further support comparative interpretation, a radar chart visualization is provided to illustrate the relative distribution of key structural quality attributes across the compared metamodels (Figure 9). This visual representation highlights trade-offs among different design dimensions and enables intuitive comparison without relying solely on numerical values. The radar chart shows that the proposed metamodel achieves a well-balanced profile, avoiding extreme values that could negatively affect maintainability or scalability.

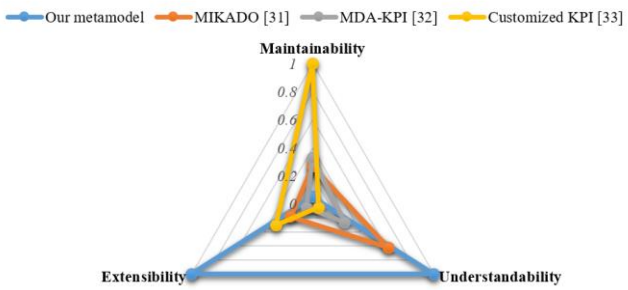

Figure 9. The normalized and direction-consistent structural quality metrics of our metamodel (KPIforBlockchain) and reference metamodels, where higher values indicate better structural quality.

Overall, this structural comparison demonstrates that the proposed KPI for Blockchain metamodel is well-suited for systematic performance modeling and analysis. The results indicate that the metamodel provides a solid foundation for integrating computational indicators, supporting automated model generation, and accommodating future extensions in blockchain and AI-driven evaluation scenarios.

6. Conclusion

This work introduced KPIforBlockchain, a model-driven framework for systematically representing and analyzing key performance indicators in blockchain systems. Instead of emphasizing platform-specific benchmarking or empirical performance measurements, the proposed approach addresses a fundamental gap in existing research by providing a structured, extensible metamodel that captures relationships between blockchain features and multidimensional performance indicators. The study demonstrated how model-driven engineering principles can be applied to blockchain performance evaluation by explicitly separating blockchain feature modeling from KPI interpretation and computation. Through the proposed metamodel, performance indicators are no longer treated as isolated metrics, but as interrelated analytical constructs grounded in measurable system characteristics. This design enables consistent reasoning about performance trade-offs across heterogeneous blockchain configurations, independent of specific platforms or deployment environments. A comprehensive set of KPIs covering performance, cost, scalability, security, decentralization, and auxiliary dimensions was organized within a unified categorization hierarchy. The case–study–based evaluation illustrated how the framework supports comparative and multi-dimensional analysis by computing and visualizing KPI behavior across representative blockchain configurations. Importantly, these results are intended to demonstrate the analytical behavior and expressive power of the framework rather than to serve as empirical benchmarks of real-world blockchain deployments. In addition, the paper explored integrating large language models (LLMs) into the model-driven workflow to assist in generating blockchain models that conform to the proposed metamodel. The evaluation of LLM-assisted model generation showed that example-based prompting significantly improves the structural correctness and completeness of generated models, highlighting the potential of AI-assisted techniques for supporting complex modeling tasks in blockchain and other distributed systems.

At the metamodel level, a structural comparison with existing KPI-oriented metamodels demonstrated that KPIforBlockchain achieves a balanced trade-off between expressive power and structural complexity. While richer features and KPI representations increase model size, the use of abstract categories and inheritance hierarchies improves understandability and extensibility, supporting future evolution and reuse. Overall, this work provides a conceptual and analytical foundation for structured blockchain performance evaluation. By combining metamodeling, formalized KPIs, and AI-assisted model generation, the proposed framework enables systematic analysis, comparative reasoning, and decision support in the design of blockchain systems. Future research directions include extending the metamodel to incorporate emerging blockchain architectures and consensus mechanisms, refining computational indicators with domain-specific constraints, and exploring advanced AI-driven techniques for automated model refinement, consistency checking, and adaptive performance analysis. These extensions would further enhance the applicability of model-driven evaluation frameworks in increasingly complex and interconnected blockchain-enabled environments.

7. Acknowledgements

The authors declare that no individuals other than the listed authors contributed to this research. The authors used ChatGPT, an artificial intelligence–based language assistance tool, to improve the clarity and fluency of the manuscript, as they are non-native English speakers. The scientific ideas, methodological design, implementation, analysis, and final content of the manuscript were entirely conceived, developed, and validated by the authors, and the use of AI tools was fully supervised.

8. Funding

This research received no external funding. The manuscript is derived from the first author's Master's thesis, conducted without financial support from funding agencies in the public, commercial, or non-profit sectors.

9. Author Contributions

Kimiya Karimi Dehkordi contributed to conceptualization, methodology, metamodel design, modeling, implementation, data analysis, investigation, and preparation of the original draft. Leila Samimi-Dehkordi contributed to conceptualization, supervision, methodological design, validation, analysis, and critical review and editing of the manuscript, and provided continuous scientific guidance throughout the research process. Abbas Horri contributed to supervision, conceptual discussion, and scientific guidance on selected aspects of the study. All authors reviewed and approved the final version of the manuscript.

10. References

[1] [1] Zheng, X., Zhu, Y., & Si, X. (2019). A survey on challenges and progresses in blockchain technologies: A performance and security perspective. Applied Sciences, *9*(22), 4731. DOI:10.3390/app9224731

[2] [2] Grida, M. O., Abd Elrahman, S., & Eldrandaly, K. A. (2022). Critical success factors evaluation for blockchain's adoption and implementing. Systems, *11*(1), 2. DOI:10.3390/systems11010002

[3] [3] Brambilla, M., Cabot, J., & Wimmer, M. (2017). Model-driven software engineering in practice (2nd ed.). Morgan & Claypool Publishers.

[4] [4] Ibatova, A. Z., Kuzmenko, V. I., & Glychova, G. (2018). Key performance indicators of management consulting. Management Science Letters, *8*(5), 475–482.

[5] [5] Abdullah, A. S. B., Zulkifli, N. F. B., Zamri, N. F. B., Harun, N. W. B., & Abidin, N. Z. Z. (2022). The benefits of having key performance indicators (KPI) in public sector. International Journal of Academic Research in Accounting, Finance and Management Sciences, *12*(3), 719–726. DOI:10.6007/IJARAFMS/v12-i3/15064

[6] [6] Sendall, S., & Kozaczynski, W. (2003). Model transformation: The heart and soul of model-driven software development. IEEE Software, *20*(5), 42–45. DOI:10.1109/MS.2003.1231150

[7] [7] Mernik, M., Heering, J., & Sloane, A. M. (2005). When and how to develop domain-specific languages. ACM Computing Surveys, *37*(4), 316–344. DOI:10.1145/1118890.1118892

[8] [8] Tonelli, R., Pierro, G. A., Ortu, M., & Destefanis, G. (2023). Smart contracts software metrics: A first study. PLoS ONE, *18*(4), e0281043. DOI:10.1371/journal.pone.0281043

[9] [9] Bovenzi, G., Aceto, G., Persico, V., & Pescapè, A. (2023). Blockchain performance in Industry 4.0: Drivers, use cases, and future directions. Journal of Industrial Information Integration, *36*, 100513. DOI:10.1016/j.jii.2023.100513

[10] [10] Madhwal, Y., Borbon-Galvez, Y., Etemadi, N., Yanovich, Y., & Creazza, A. (2022). Proof of delivery smart contract for performance measurements. IEEE Access, *10*, 69147–69159. DOI:10.1109/ACCESS.2022.3185634

[11] [11] Merrad, Y., Habaebi, M. H., Islam, M. R., Gunawan, T. S., & Mesri, M. (2022). Blockchain: Consensus algorithm key performance indicators, trade-offs, current trends, common drawbacks, and novel solution proposals. Mathematics, *10*(15), 2754. DOI:10.3390/math10152754

[12] [12] Ajienka, N., Vangorp, P., & Capiluppi, A. (2020). An empirical analysis of source code metrics and smart contract resource consumption. Journal of Software: Evolution and Process, *32*(10), e2267. DOI:10.1002/smr.2267

[13] [13] Pinna, A., Ibba, S., Baralla, G., Tonelli, R., & Marchesi, M. (2019). A massive analysis of Ethereum smart contracts: Empirical study and code metrics. IEEE Access, *7*, 78194–78213. DOI:10.1109/ACCESS.2019.2921936

[14] [14] Alaslani, M., Nawab, F., & Shihada, B. (2019). Blockchain in IoT systems: End-to-end delay evaluation. IEEE Internet of Things Journal, *6*(5), 8332–8344. DOI:10.1109/JIOT.2019.2917226

[15] [15] Alrubei, S., Rigelsford, J., Willis, C., & Ball, E. (2019). Ethereum blockchain for securing the Internet of Things: Practical implementation and performance evaluation. In Proceedings of the International Conference on Cyber Security and Protection of Digital Services (CyberSecPODS). IEEE. DOI:10.1109/CyberSecPODS.2019.8885029

[16] [16] Cui, P., Guin, U., Skjellum, A., & Umphress, D. (2019). Blockchain in IoT: Current trends, challenges, and future roadmap. Journal of Hardware and Systems Security, *3*(4), 338–364. DOI:10.1007/s41635-019-00079-5

[17] [17] Huang, J., Kong, L., Chen, G., Wu, M.-Y., Liu, X., & Zeng, P. (2019). Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism. IEEE Transactions on Industrial Informatics, *15*(6), 3680–3689. DOI:10.1109/TII.2019.2903342

[18] [18] Cao, B., Li, Y., Zhang, L., Zhang, L., Mumtaz, S., Zhou, Z., & Peng, M. (2019). When Internet of Things meets blockchain: Challenges in distributed consensus. IEEE Network, *33*(6), 133–139. DOI:10.1109/MNET.2019.1900002

[19] [19] Liu, Y., Wang, K., Lin, Y., & Xu, W. (2019). LightChain: A lightweight blockchain system for industrial Internet of Things. IEEE Transactions on Industrial Informatics, *15*(6), 3571–3581. DOI:10.1109/TII.2019.2904049

[20] [20] Novo, O. (2018). Blockchain meets IoT: An architecture for scalable access management in IoT. IEEE Internet of Things Journal, *5*(2), 1184–1195. DOI:10.1109/JIOT.2018.2812239

[21] [21] Kuhi, K., Kaare, K., & Koppel, O. (2018). Ensuring performance measurement integrity in logistics using blockchain. In Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI). IEEE. DOI:10.1109/SOLI.2018.8476737

[22] [22] Thakkar, P., Nathan, S., & Viswanathan, B. (2018). Performance benchmarking and optimizing Hyperledger Fabric blockchain platform. In Proceedings of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE. DOI:10.1109/MASCOTS.2018.00034

[23] [23] Hao, Y., Li, Y., Dong, X., Fang, L., & Chen, P. (2018). Performance analysis of consensus algorithm in private blockchain. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV). IEEE. DOI:10.1109/IVS.2018.8500557

[24] [24] Jiang, T., Fang, H., & Wang, H. (2018). Blockchain-based Internet of Vehicles: Distributed network architecture and performance analysis. IEEE Internet of Things Journal, *6*(3), 4640–4649. DOI:10.1109/JIOT.2018.2874398

[25] [25] Zheng, P., Zheng, Z., Luo, X., Chen, X., & Liu, X. (2018). A detailed and real-time performance monitoring framework for blockchain systems. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP '18)* (pp. 134–143). ACM. DOI:10.1145/3183519.3183546

[26] [26] Rouhani, S., & Deters, R. (2017). Performance analysis of Ethereum transactions in private blockchain. In Proceedings of the IEEE International Conference on Software Engineering and Service Science (ICSESS). IEEE. DOI:10.1109/ICSESS.2017.8342866

[27] [27] Pongnumkul, S., Siripanpornchana, C., & Thajchayapong, S. (2017). Performance analysis of private blockchain platforms in varying workloads. In Proceedings of the IEEE International Conference on Computer Communication and Networks (ICCCN). IEEE. DOI:10.1109/ICCCN.2017.8038517

[28] [28] Meng, M. H., & Qian, Y. (2018). A blockchain aided metric for predictive delivery performance in supply chain management. In Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI). IEEE. DOI:10.1109/SOLI.2018.8476723

[29] [29] Hayat, M., & Winkler, H. (2022). An analytic hierarchy process for selection of blockchain-based platform for product lifecycle management. Sustainability, *14*(21), 13703. DOI:10.3390/su142113703

[30] [30] Fan, C., Ghaemi, S., Khazaei, H., & Musilek, P. (2020). Performance evaluation of blockchain systems: A systematic survey. IEEE Access, *8*, 126927–126950. DOI:10.1109/ACCESS.2020.3006078

[31] [31] De Sanctis, M., Iovino, L., Rossi, M. T., & Wimmer, M. (2022). MIKADO: A smart city KPIs assessment modeling framework. Software and Systems Modeling, *21*(1), 281–309. DOI:10.1007/s10270-021-00907-9

[32] [32] Letrache, K., El Beggar, O., & Ramdani, M. (2016). Modeling and creating KPIs in MDA approach. In Proceedings of the IEEE International Colloquium on Information Science and Technology (CIST). IEEE. DOI:10.1109/CIST.2016.7805046

[33] [33] André, P., & Goepp, V. (2023). A framework for defining customised KPI in manufacturing systems. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing (SOHOMA 2023)* (pp. 309–320). Springer. DOI:10.1007/978-3-031-53445-4_26

[34] [34] Genero, M., & Piattini, M. (2001). Empirical validation of measures for class diagram structural complexity through controlled experiments. In Proceedings of the ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering. DOI:10.1007/3-540-46102-7_40

[35] [35] Sheldon, F. T., & Chung, H. (2006). Measuring the complexity of class diagrams in reverse engineering. Journal of Software Maintenance and Evolution: Research and Practice, *18*(5), 333–350. DOI:10.1002/smr.336

[36] [36] Arendt, T., Mantz, F., & Taentzer, G. (2009). UML model quality assurance techniques (Technical Report). Philipps-University Marburg, Germany.