

Article

A Requirement-Driven Expert System for Blockchain Consensus Mechanism Selection

Ivica Lukić , Nikola Ramčić, Ivan Ivković  and Miljenko Švarcmajer 

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, Kneza Trpimira 2b, 31000 Osijek, Croatia; nikola.ramcic@student.ferit.hr (N.R.); ivan.ivkovic@ferit.hr (I.I.); miljenko.svarcmajer@ferit.hr (M.Š.)

* Correspondence: ivica.lukic@ferit.hr

Abstract

Blockchain technology has introduced a rapidly expanding range of consensus mechanisms, each designed to satisfy different operational requirements related to security, scalability, decentralization, transaction throughput, and energy efficiency. Selecting an appropriate consensus mechanism has consequently become a complex multi-criteria decision problem, particularly for developers and organizations without extensive expertise in distributed systems and blockchain architectures. Existing tools primarily address protocol benchmarking and static documentation, leaving the decision-support dimension largely unaddressed. This paper presents a web-based expert system designed to support the selection of blockchain consensus mechanisms according to specific user-defined operational requirements. The proposed solution was implemented using the MERN technology stack, consisting of MongoDB, Express.js, React, and Node.js, enabling a modular and scalable architecture suitable for future expansion and maintenance. The recommendation process is based on a two-phase filtering and scoring algorithm. In the first phase, mechanisms incompatible with mandatory user-defined constraints, including network type and key resource type, are systematically eliminated. In the second phase, the remaining mechanisms are ranked using attribute matching across criteria encompassing energy efficiency, scalability, security, decentralization, and transaction speed. The system returns the three most suitable consensus mechanisms for the given operational scenario together with their key characteristics. In addition to recommendation functionality, the application supports user authentication, recommendation history management, and administrative maintenance of the consensus mechanism database, which currently contains 38 distinct blockchain consensus protocols. Experimental evaluation through twelve representative usage scenarios demonstrated that the system consistently produces contextually relevant recommendations aligned with user-specified requirements. A comparative analysis with existing tools confirms that the proposed system occupies a distinct decision-support role currently absent from the available tooling landscape. A comparative analysis against four representative existing tools indicates that the proposed system combines a set of decision-support capabilities not jointly offered by any one of them. The presented approach contributes a transparent and extensible decision-support framework intended to simplify architectural planning and management of blockchain-based distributed systems.



Academic Editor: Mehdi Sookhak

Received: 1 June 2026

Revised: 15 July 2026

Accepted: 20 July 2026

Published: 22 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: blockchain; consensus mechanism; expert system; MERN stack; decision support; distributed systems

1. Introduction

Blockchain technology has evolved from a cryptocurrency-oriented concept into a broad distributed systems paradigm applied across domains such as finance, Internet of Things (IoT), supply chain management, digital identity, and decentralized applications [1,2]. A fundamental component of every blockchain architecture is the consensus mechanism, which enables geographically distributed nodes to agree on the validity of transactions and maintain a consistent state of the shared ledger without relying on a centralized authority [3]. Over time, many consensus mechanisms have emerged, including Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Byzantine Fault Tolerance (BFT)-based approaches, and numerous hybrid variations [4]. Each mechanism introduces different trade-offs regarding scalability, decentralization, energy efficiency, transaction throughput, and security.

The growing diversity of blockchain consensus mechanisms has significantly increased the complexity of selecting an appropriate solution for specific operational and organizational requirements. In practice, the selection process depends on multiple interconnected factors, including network type, resource availability, scalability requirements, security constraints, transaction speed, and long-term system maintainability [5]. An unsuitable consensus mechanism may negatively affect infrastructure costs, network performance, resilience, and the sustainability of distributed systems. Consequently, selecting an appropriate consensus mechanism has become an important architectural and organizational challenge that cannot be adequately addressed through informal review of protocol documentation alone.

Although numerous studies analyse blockchain consensus protocols and compare their characteristics, existing practical decision-support tools remain limited [6]. Current platforms such as Blockbench and Hyperledger Caliper primarily focus on benchmarking blockchain performance under predefined workloads rather than assisting users during the initial decision-making process [7,8]. Similarly, available online consensus explorers provide static protocol descriptions without supporting personalized recommendation capabilities based on user-defined operational constraints [9,10]. As a result, developers and organizations without extensive expertise in distributed systems often face significant difficulties when attempting to identify the most suitable consensus mechanism for a particular use case.

Expert systems represent an established paradigm for encoding domain knowledge into structured decision-support tools applicable to complex multi-criteria selection problems [11]. By combining a structured knowledge base with explicit inference rules, expert systems provide transparent, explainable, and reproducible recommendations that are particularly valuable in architectural planning environments where decisions must be justified and auditable [12]. Despite the suitability of this paradigm for blockchain consensus selection, the application of expert systems in this domain remains comparatively unexplored in the current research literature.

To address this gap, this paper presents a web-based expert system designed to support blockchain consensus mechanism selection according to specific user-defined requirements. The proposed system was implemented using the MERN technology stack and employs a two-phase recommendation algorithm based on hard filtering constraints and attribute-based scoring. The system evaluates consensus mechanisms according to criteria including network type, energy efficiency, scalability, decentralization, security, and transaction speed, returning the three most suitable mechanisms for a given scenario together with their key operational characteristics.

The main contributions of this paper are as follows:

- Development of a web-based expert system for blockchain consensus mechanism selection implementing a two-phase filtering and scoring recommendation model.
- Creation of an extensible consensus mechanism knowledge base containing 38 blockchain consensus protocols with structured attribute representations.
- Development of an administrative interface enabling non-technical maintenance and extension of the knowledge base without source code modification.
- Systematic evaluation through twelve representative recommendation scenarios covering a diverse range of user-defined constraint combinations.
- Comparative analysis of the proposed system against existing blockchain tools, demonstrating the distinct decision-support contribution of the proposed approach.

The remainder of this paper is organized as follows. Section 2 provides background on consensus mechanism characteristics and reviews related work in blockchain analysis, benchmarking, and decision support. Section 3 describes the proposed system architecture and component design. Section 4 presents the implementation details and the recommendation algorithm. Section 5 reports the experimental evaluation results and comparative analysis. Section 6 discusses findings, limitations, and future directions, and Section 7 concludes the paper.

2. Background and Related Work

Consensus mechanisms constitute the core protocol layer of any blockchain architecture, governing how distributed nodes reach agreement on the state of a shared ledger in the absence of a trusted central authority. The earliest and most widely recognized consensus mechanism, Proof of Work (PoW), was introduced by Nakamoto [3] and requires participating nodes to solve computationally intensive cryptographic puzzles to earn the right to append new blocks. While PoW provides strong security guarantees and high decentralization, its substantial energy consumption and limited transaction throughput have motivated the development of alternative approaches [4].

Proof of Stake (PoS) and its derivatives, including Delegated Proof of Stake (DPoS), address the energy inefficiency of PoW by replacing computational work with economic stake as the basis for block validation rights [13]. PoS-based mechanisms generally achieve higher energy efficiency and improved transaction throughput, though they introduce different decentralization and security trade-offs depending on the specific implementation. Byzantine Fault Tolerant (BFT) consensus protocols, including Practical Byzantine Fault Tolerance (PBFT) and its successors, are commonly deployed in permissioned blockchain environments due to their deterministic finality and low latency, albeit with limited scalability for large node counts [14].

More recently, novel consensus paradigms have emerged to address specific operational requirements. Proof of History (PoH), used in the Solana blockchain, introduces a verifiable delay function to establish temporal ordering of transactions, enabling high throughput with low confirmation latency [15]. Directed Acyclic Graph (DAG)-based consensus structures, such as those employed in IOTA and Fantom, depart from the linear chain model entirely, offering potential improvements in scalability and parallelism [16]. Storage-based mechanisms including Proof of Capacity (PoC) and Proof of Space leverage available disk space as the primary consensus resource, providing an energy-efficient alternative to computational PoW [17].

The diversity of available consensus mechanisms means that no single protocol is universally optimal across all deployment contexts. Selection must account for the specific combination of operational requirements present in each use case, making structured decision support a valuable tool for practitioners.

Several studies have examined blockchain consensus mechanisms from the perspectives of scalability, security, decentralization, and performance evaluation. Bamakan et al. [6] presented one of the most comprehensive surveys of blockchain consensus algorithms, proposing a taxonomy based on evaluation criteria including energy efficiency, scalability, fault tolerance, and network structure. Their classification framework provided a structured foundation for the knowledge base design employed in the present work.

Vukolic [18] analysed the architectural differences between BFT-based protocols and Nakamoto-style consensus mechanisms, emphasizing communication overhead, scalability limitations, and distributed agreement complexity. The study highlighted the importance of selecting consensus protocols aligned with operational requirements and network constraints, supporting the motivation for the expert system approach proposed in the present paper.

Cachin and Vukolic [19] further examined the evolution of consensus protocols in permissioned blockchain systems, discussing the scalability limitations of classical BFT approaches and potential improvement directions. Their analysis reinforced the observation that consensus selection requires context-sensitive evaluation rather than universal ranking.

In the area of blockchain benchmarking, Dinh et al. [7] introduced Blockbench, a framework for analysing private blockchain systems through metrics including throughput, latency, and fault tolerance under controlled workloads. Hyperledger Caliper [8] provides quantitative benchmarking of blockchain networks by measuring transaction throughput, latency, CPU utilization, and memory consumption. While these platforms deliver valuable performance evaluation capabilities, they are fundamentally oriented toward post-selection validation rather than initial decision support.

Multi-criteria decision-making (MCDM) methods have been applied to technology selection problems in related domains. Analytical Hierarchy Process (AHP) and Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) have been used to evaluate cloud service providers, IoT platforms, and database management systems based on weighted criteria [20,21]. The application of structured scoring and filtering methods to blockchain consensus selection, as proposed in the present work, extends this established MCDM paradigm to an emerging and practically relevant domain.

Expert systems for technology selection have been explored in several adjacent areas. Turban and Aronson [11] provide a foundational treatment of knowledge-based decision support systems, establishing the conceptual basis for encoding domain expertise as structured rules and scoring models. More recently, web-based expert systems have been applied to domains including medical diagnosis, environmental impact assessment, and engineering component selection [22,23]. Despite the established utility of expert systems in analogous selection problems, their application to blockchain consensus mechanism selection remains limited in the current literature.

Beyond AHP and TOPSIS, other MCDM formulations relevant to consensus mechanism selection have been proposed in the blockchain literature, including a dedicated MCDM-based decision framework for blockchain consensus protocol selection [24] and fuzzy-set extensions such as fuzzy TOPSIS and fuzzy VIKOR that accommodate imprecise or linguistically expressed criterion values [25]. Relative to these MCDM-only formulations, the present work contributes an integrated, deployable system rather than a standalone ranking method. The two-phase architecture couples hard, auditable constraint filtering with attribute-based scoring, and the accompanying administrative interface allows the underlying knowledge base and, in principle, the criterion weights to be maintained and extended by non-technical domain experts without modifying the decision logic itself. This positions the contribution as an applied expert-system realization of MCDM principles,

rather than a proposal of a new MCDM method competing with AHP, or fuzzy TOPSIS on theoretical grounds.

Regarding tools specifically addressing blockchain consensus information, AccountEnd [9] provides a simple consensus mechanism explorer containing descriptive summaries of selected protocols, while CryptoCompare [10] enables filtering and comparison of blockchain-related assets. However, neither platform supports personalized recommendation processes based on structured user-defined operational requirements. The expert system proposed in the present paper addresses this limitation through an interactive decision-support framework combining mandatory constraint filtering with attribute-based scoring, delivering transparent and reproducible recommendations tailored to specific deployment contexts.

3. System Design and Architecture

The proposed system was designed as a web-based expert system implementing a modular client-server architecture using the MERN technology stack [26], consisting of MongoDB, Express.js, React, and Node.js. This technology selection enables separation of frontend presentation logic, backend business logic, and database management into independently maintainable components while providing a consistent JavaScript-based development environment across all layers. The system architecture consists of four primary components: the React-based frontend interface, the Express.js backend API, the recommendation engine, and the MongoDB database layer. Figure 1 illustrates the overall architecture and communication flow between the system components. Communication between the frontend and backend is performed exclusively through RESTful API endpoints, with all data transferred in JSON format using the Axios HTTP client library. This decoupled design enables independent evolution of the frontend and backend components and simplifies potential future migration to alternative frontend frameworks.

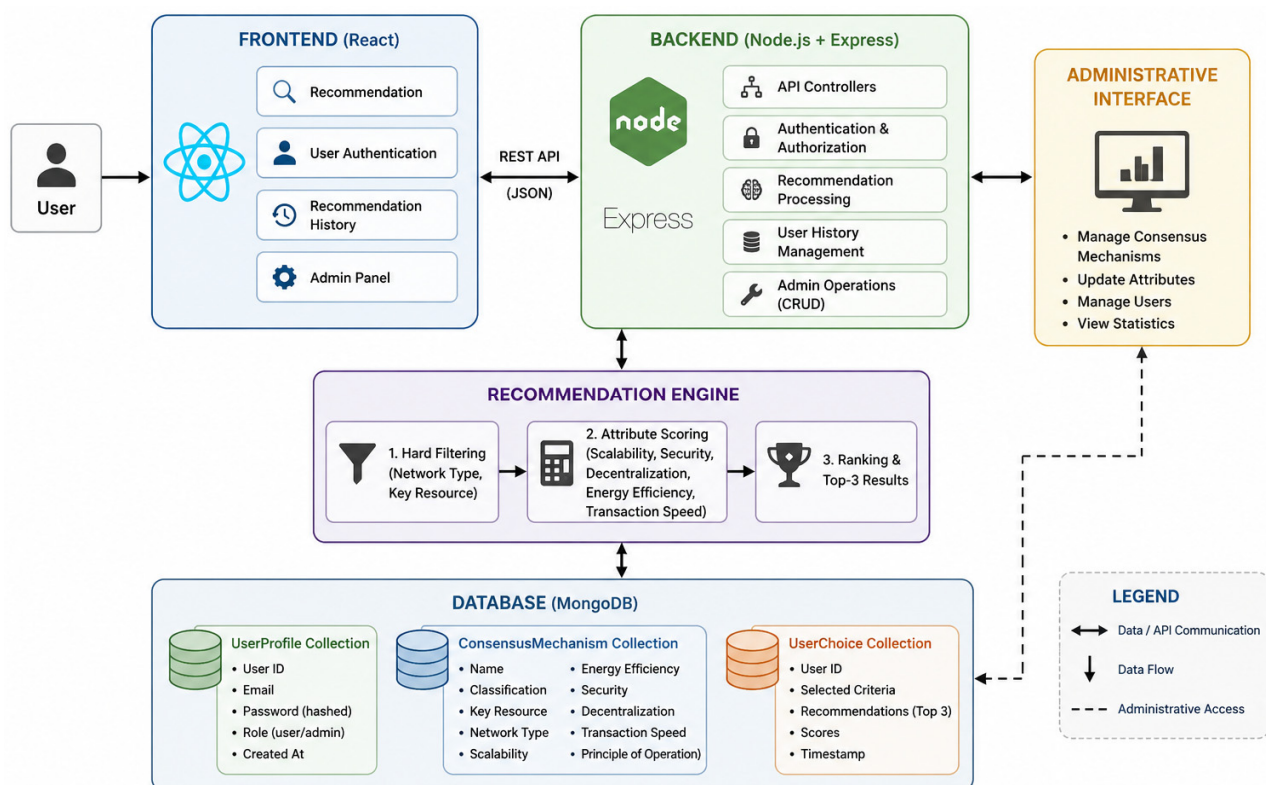


Figure 1. Overall system architecture of the proposed blockchain consensus recommendation expert system.

MongoDB was selected as the database management system due to its native compatibility with JSON-like document structures commonly used in JavaScript-based applications and its flexibility for representing heterogeneous consensus mechanism attributes without requiring a rigid relational schema. Three primary collections are used within the system.

The UserProfile collection stores authentication-related information for each registered user, including username, bcrypt-hashed password, account creation timestamp, and user role designation distinguishing standard users from administrators. The ConsensusMechanism collection constitutes the core knowledge base of the expert system and contains detailed attribute records for each of the 38 consensus mechanisms supported by the system. The UserChoice collection persists historical recommendation sessions, recording both the criteria specified by the user and the recommendation results generated by the system, enabling historical analysis and future personalization features.

Each consensus mechanism record in the ConsensusMechanism collection contains the following attributes used during the recommendation process: mechanism name, classification category, network type compatibility, key resource type, scalability level, security level, decentralization level, energy efficiency rating, transaction speed rating, and a textual description of the operating principle. All ordinal attribute values are represented using a three-level scale of low, medium, and high, enabling consistent comparison across mechanisms with different underlying technical implementations.

The knowledge base comprises 38 blockchain consensus mechanisms derived from scientific literature and publicly available blockchain consensus taxonomies. The mechanisms span all major consensus paradigm families, including computational resource mechanisms (PoW and variants), stake-based mechanisms (PoS, DPoS, PoA, and variants), storage-based mechanisms (PoC, PoSpace, PoST), BFT-based mechanisms (PBFT, Tendermint, HotStuff), DAG-based consensus structures, and identity-based mechanisms. Both public permissionless and private permissioned network environments are represented within the knowledge base. Table 1 presents the primary evaluation attributes used within the recommendation process together with their descriptions and value domains.

Table 1. Consensus Mechanism Evaluation Attributes.

Attribute	Description
Network Type	Public or private blockchain environment
Key Resource	Computational, stake-based, storage-based, or hybrid resource
Energy Efficiency	Estimated energy consumption level
Scalability	Ability to support large transaction volumes
Security	Resistance to attacks and malicious activity
Decentralization	Degree of network distribution
Transaction Speed	Relative transaction throughput capability

The Low /Medium/High labels for each of the five scoring attributes were assigned through a structured literature-review process. For each mechanism, primary sources—the original protocol specification or whitepaper, peer-reviewed surveys and comparative studies covering the corresponding consensus family, and, where available, independently reported benchmarking results—were consulted to determine the attribute level. Where a mechanism lacked a dedicated benchmarking study, its attribute levels were derived by analogy with closely related, better-documented mechanisms of the same consensus family, subject to the cross-checking procedure described below. Where sources reported conflicting or ambiguous values for a given attribute, the discrepancy was resolved through expert judgment by the authors, applying a conservative rule: the label was set to the value supported by the majority of consulted sources, or, in case of an even split, to the more cautious (lower) level. To ensure internal consistency across the 38 mechanisms, classifications

were cross-checked against the paradigm-family groupings (e.g., all BFT-based mechanisms were reviewed together) to confirm that mechanisms sharing a consensus family did not receive contradictory attribute levels without an explicit, documented justification.

The 38 mechanisms included in the knowledge base were selected based on the following criteria: (i) the mechanism is established and has at least one publicly available specification or peer-reviewed description, (ii) sufficient public information exists to reliably populate all five scoring attributes and the key_resource classification, and (iii) the mechanism is either in active production use or has been the subject of substantial academic or industrial evaluation. Newer or less-documented protocols for which reliable attribute values could not be established were excluded from the current version of the knowledge base. The knowledge base is designed to be extensible via the administrator interface described in Section 4, allowing new mechanisms to be incorporated as they mature and as sufficient documentation becomes available.

The recommendation process is implemented through a two-phase filtering and scoring algorithm. In the first phase, mechanisms incompatible with strict user-defined constraints are eliminated. This includes filtering according to network type and key resource requirements. Mechanisms that do not satisfy mandatory conditions are excluded from further evaluation. Figure 2 presents the workflow of the recommendation process implemented within the proposed expert system.

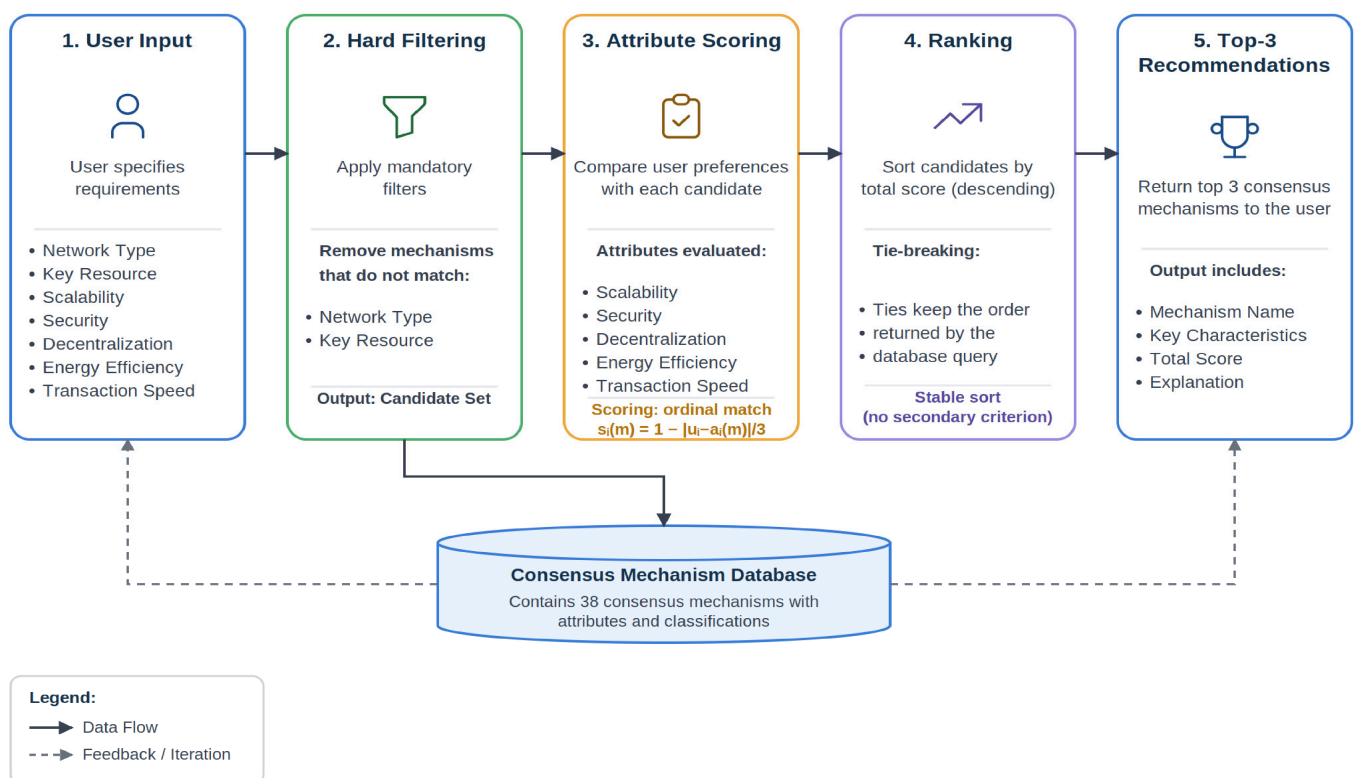


Figure 2. Workflow of the recommendation process.

In the second phase, the remaining candidate mechanisms are evaluated using an attribute-based scoring model. The system compares user-selected preferences against mechanism properties including scalability, security, decentralization, energy efficiency, and transaction speed. Each matching attribute contributes to the final recommendation score proportionally to its assigned weight, as defined in the scoring model presented in Section 4.3. The mechanisms are subsequently ranked according to the calculated score, and the three highest-ranked candidates are returned to the user.

The proposed architecture was intentionally designed to support future extensibility and simplified maintenance. New consensus mechanisms can be added through the administrative interface without modifying the application source code, while additional evaluation criteria and scoring models may be integrated with minimal architectural changes. Such flexibility is particularly important due to the rapidly evolving nature of blockchain consensus research and distributed system infrastructures.

4. Implementation and Recommendation Process

The proposed expert system was implemented as a full-stack web application using the MERN technology stack, consisting of MongoDB, Express.js, React, and Node.js. The implementation was divided into frontend and backend layers, enabling separation of presentation logic, business logic, and data management. Such an architecture improves maintainability and simplifies future extension of the system.

4.1. Backend Implementation

The backend system was implemented using Node.js and Express.js and serves as the central processing layer of the application. Its primary responsibilities include user authentication, database communication, recommendation processing, and administrative management of consensus mechanisms. The backend architecture follows a modular controller-based structure in which individual functionalities are separated into independent components.

Four primary controllers were implemented within the system:

- AuthenticationController, responsible for user registration and login,
- RecommendationsController, responsible for generating and storing recommendations,
- UserController, responsible for recommendation history management,
- AdminController, responsible for administrative CRUD operations over the consensus mechanism database.

Communication between the frontend and backend layers is performed through RESTful API endpoints. The implemented API supports authentication operations, recommendation requests, recommendation history retrieval, and database maintenance functions. User authentication was implemented using hashed password storage through the bcrypt library. The system distinguishes standard users from administrators, where administrative accounts are authorized to modify the consensus mechanism database through dedicated management interfaces.

4.2. Database Structure

The ConsensusMechanism collection was populated with 38 entries derived from scientific literature and publicly available blockchain consensus taxonomies. All ordinal attributes are stored as enumerated string values (low, medium, high) to enable direct equality matching during the filtering and scoring phases. The UserChoice collection records each recommendation session as a single document embedding both the input criteria and the returned results, enabling efficient history retrieval without requiring joins across collections.

4.3. Recommendation Algorithm

The first phase is implemented as a single MongoDB query applying equality conditions on the network type and key resource fields, returning only documents satisfying both mandatory constraints. This database-level filtering avoids loading incompatible mechanisms into application memory, improving response efficiency as the knowledge

base grows. In the second phase, the remaining candidate mechanisms are evaluated using the weighted scoring model defined formally below.

Formally, let $M = \{m_1, m_2, \dots, m_n\}$ be the set of all mechanisms in the knowledge base, and let $C = \{c_{\text{network}}, c_{\text{resource}}\}$ be the set of mandatory user constraints. The filtered candidate set M' is defined as:

$$M' = \{m \in M \mid m.\text{network_type} = c_{\text{network}} \wedge m.\text{key_resource} = c_{\text{resource}}\} \quad (1)$$

Mechanisms that fail mandatory requirements are permanently removed from the candidate set and do not participate in the scoring phase.

The scoring model for the second phase compares user-specified preference values against the corresponding mechanism attributes across five evaluated criteria: scalability, security, decentralization, energy efficiency, and transaction speed. Following the hard filtering phase, the remaining consensus mechanisms $m \in M'$ are evaluated using a weighted scoring model. This model quantifies the suitability of each mechanism based on a predefined set of evaluation criteria. The total score $S(m)$ for a given mechanism M' is calculated as:

$$S(m) = \sum_{i=1}^n w_i \cdot s_i(m) \quad (2)$$

where n denotes the total number of evaluation criteria, w_i represents the weight assigned to the i -th criterion (reflecting its relative importance to the user), and $s_i(m)$ is the normalized score assigned to mechanism m for the i -th criterion. By applying this weighted sum, the system facilitates a multi-criteria decision-making process, ensuring that the selected consensus protocol aligns with the specific requirements and priorities of the target application.

In the evaluated implementation, n denotes the number of criteria for which the user specified a preference in each request, rather than a fixed value of five. Criteria left unspecified by the user are excluded from the scoring computation for that request. The weight w_i is a configurable parameter of the model that can, in principle, be set to reflect a user- or domain-specific priority ordering among criteria. In the configuration evaluated in this paper, w_i is fixed to $1/n$ for every specified criterion, i.e., all specified criteria are weighted equally. This equal-weighting configuration was adopted as a neutral baseline in the absence of empirical data on domain-specific priorities, as discussed in Section 6.1, and is distinct from the general weighted-sum formulation in Equation (2), which supports non-uniform weights for future, domain-calibrated deployments.

The criterion score $s_i(m)$ is computed from the ordinal distance between the user's requested level and the mechanism's attribute level for criterion i . Low, Medium, and High are encoded as the integer values 1, 2, and 3, respectively. Let u_i denote the encoded value of the user's request for criterion i and $a_i(m)$ denote the encoded value of mechanism m for that criterion. The raw match score is defined as $r_i(m) = 3 - |u_i - a_i(m)|$, and the normalized criterion score is $s_i(m) = r_i(m)/3$. This formulation implements ordinal partial matching rather than exact matching only: an exact match yields $s_i(m) = 1$, a one-level mismatch (e.g., a High request matched against a Medium attribute) yields $2/3$, and a two-level mismatch (e.g., a High request matched against a Low attribute) yields $1/3$.

Consider a request specifying High energy efficiency, High scalability, and High decentralization ($n = 3$, $w_i = 1/3$ for each specified criterion), consistent with Scenario A in Table 2. For an illustrative candidate mechanism m with attribute levels High efficiency, High scalability, and Medium decentralization, the criterion scores are $s_1(m) = 1$ (exact match on efficiency), $s_2(m) = 1$ (exact match on scalability), and $s_3(m) = 2/3$ (one-level mismatch on decentralization). The total score is $S(m) = (1/3)(1) + (1/3)(1) + (1/3)(2/3) = 8/9 \approx 0.89$, which is compared against the scores of all other candidates remaining after hard filtering.

The three mechanisms with the highest $S(m)$ are returned. The attribute values used in this example are illustrative, the complete, verified attribute values for all 38 mechanisms are provided in the extended knowledge base table (Supplementary Material, Table S1, sheet KnowledgeBase_S1). Following score computation, all mechanisms in M' are sorted in descending order according to their score $S(m)$. The three highest-ranked mechanisms are returned as recommendations together with their key attribute values and a textual description of their operating principle. In the case of score ties, mechanisms retain the order in which they were returned from the underlying database query, a consequence of the stable sort applied during ranking rather than an explicit secondary ranking criterion. This equal-weight design was chosen to avoid introducing implicit bias in the absence of empirical data on the relative importance of different criteria across diverse deployment contexts. Future iterations may incorporate weighted scoring models calibrated through domain expert elicitation or historical usage data, as discussed in Section 6.

Table 2. Recommendation Scenarios and Results.

Scenario	Network	Resource	Key Preferences	Top-3 Recommendations
A	Public	Stake	High efficiency, High scalability, High decentralization	PoS, PoH, Casper
B	Public	Stake	High security, High decentralization, Low efficiency	Casper, PoAc, NPoS
C	Public	Computational	Medium scalability, High security, Low efficiency	PoW, PoW-NG, Ethash
D	Public	Storage	High efficiency, Medium scalability, High decentralization	PoC, PoSpace, PoST
E	Public	Hybrid	High scalability, High speed, Medium security	DAG, PoA, Hashgraph
F	Private	Stake	High speed, High scalability, Low decentralization	PoA, DPoS-Private, SPoS
G	Private	Computational	High security, Medium scalability, Low decentralization	HBBFT, PBFT, HotStuff
H	Private	Hybrid	High speed, High security, Medium efficiency	Istanbul BFT, QBFT, Clique
I	Public	Stake	High efficiency, High scalability, High speed	PoS, PoH, DPoS
J	Private	Stake	High decentralization, Medium security, Medium speed	PoET, PoA, DPoS-Private
K	Public	Computational	High security, Low efficiency, Low speed	PoW, Ethash, SHA-256
L	Private	Computational	Medium security, High speed, Low decentralization	PBFT, Raft, Paxos-BFT

4.4. Frontend Implementation

The frontend application was implemented using React and provides an interactive user interface for system interaction. The interface enables users to specify desired blockchain characteristics through predefined selection forms and dropdown menus.

The frontend consists of several primary components:

- Recommendation form;
- Recommendation results interface;
- Authentication interface;
- Recommendation history interface;
- Administrative management interface.

React Router was used for client-side navigation, while Axios was used for communication with backend API endpoints. Tailwind CSS utility classes were applied to simplify

interface styling and maintain visual consistency across components. Figure 3 presents the graphical user interface of the recommendation module.

Figure 3. Graphical user interface of the recommendation module.

The recommendation interface allows users to define multiple operational criteria and receive recommendation results in real time. In addition to displaying the recommended consensus mechanisms, the system also presents key mechanism characteristics including network type, scalability, security level, and principle of operation.

4.5. Administrative Management

An administrative interface was implemented to simplify maintenance and future extension of the system knowledge base. Administrators can:

- Add new consensus mechanisms;
- Modify existing mechanism attributes;
- Remove obsolete mechanisms;
- Manage stored recommendation data.

The administrative layer allows the recommendation database to evolve without requiring direct source code modification. This extensibility is particularly important due to the rapid development of blockchain technologies and the continuous emergence of new consensus architectures.

5. Experimental Evaluation

The proposed expert system was evaluated through a structured set of twelve recommendation scenarios designed to systematically assess the behaviour of the two-phase recommendation algorithm across a diverse range of user-defined constraint combinations. The evaluation objectives were threefold: to verify the correctness of the mandatory constraint filtering phase, to assess the consistency and contextual relevance of attribute-based scoring recommendations, and to examine the coverage distribution of the knowledge base under varying input configurations.

Scenarios were constructed to cover combinations across both network type values (public, private), all four key resource types (computational, stake-based, storage-based, hybrid), and varying combinations of attribute preferences across the five scored criteria.

Each scenario was executed through direct interaction with the web application interface. API-level validation of selected scenarios was additionally performed using Postman to verify that backend recommendation logic produced consistent results independent of the frontend rendering layer.

Recommendation correctness was assessed by verifying that all returned mechanisms satisfied the mandatory filtering criteria and that the rank ordering of results was consistent with the computed attribute scores. Coverage analysis was performed by tabulating the frequency with which each of the 38 knowledge base mechanisms appeared in recommendations across the full scenario set, identifying mechanisms that were consistently recommended, contextually recommended, or unreachable under the tested constraint combinations. Table 2 summarizes the twelve evaluation scenarios and their corresponding recommendation outputs. Scenarios are organized by network type and key resource type to facilitate systematic coverage assessment.

In Scenario A, the user selected a public blockchain environment with high energy efficiency, high scalability, and high decentralization preferences. The system returned Proof of Stake (PoS), Proof of History (PoH), and Casper as the three highest-ranked mechanisms. These recommendations are contextually consistent: all three mechanisms avoid computationally intensive mining operations, support high transaction volumes, and maintain distributed validator participation structures characteristic of public decentralized networks. Casper, Ethereum's Proof-of-Stake finality gadget, ranks ahead of other stake-based candidates in this scenario owing to its combination of high decentralization and high energy efficiency.

Scenario C demonstrates the filtering behaviour for computational resource requirements in a public network context. With high security and medium scalability preferences, the system returned PoW-based mechanisms, which are expected given that computational resource selection in public networks is predominantly associated with traditional PoW protocol families offering strong security guarantees through hash-rate-based Sybil resistance.

Several mechanisms in Table 2 are introduced through abbreviations that may not be familiar to all readers. SPoS (Simplified Proof of Stake) is a lightweight, permissioned stake-based variant, and DPoS-Private denotes a permissioned adaptation of Delegated Proof of Stake in which validator delegation is restricted to a known, pre-approved participant set. Paxos-BFT denotes a Byzantine-fault-tolerant extension of the classical Paxos consensus protocol adapted for permissioned blockchain deployment. Casper (Casper FFG) is Ethereum's Proof-of-Stake finality gadget, in which Proof-of-Work block proposal is combined with a Proof-of-Stake-based finalization layer. PoAc (Proof of Activity) is a hybrid consensus mechanism in which a Proof-of-Work phase selects the eligible block proposer and a subsequent Proof-of-Stake phase has randomly selected stakeholders validate and sign the proposed block [27]. It is abbreviated PoAc rather than PoA in this manuscript to avoid confusion with Proof of Authority, which is already denoted PoA in the knowledge base. PoW-NG separates block proposal from block validation to increase throughput, with leader election for proposed blocks still based on Proof of Work while block validation itself requires no additional computational work. HBBFT (Honey-BadgerBFT) is the first practical asynchronous Byzantine-fault-tolerant protocol requiring no timing assumptions, using threshold encryption to defend against adaptive network adversaries. Hashgraph is Hedera's gossip-about-gossip and virtual-voting consensus mechanism, achieving asynchronous Byzantine fault tolerance over a directed-acyclic-graph event structure without block mining. Full descriptions and supporting references for all mechanisms, including these, are provided in the extended knowledge base table (Supplementary Material, Table S1, sheet KnowledgeBase_S1).

It is important to distinguish the descriptive paradigm-family labels used in Section 3 (e.g., BFT-based or storage-based) from the *key_resource* filtering tag actually stored for each mechanism, which takes one of four values (computational, stake-based, storage-based, hybrid) and is used exclusively by the hard-filtering phase in Equation (1). BFT-style mechanisms such as PBFT, HotStuff, and Tendermint do not consume stake or storage resources. They rely on computational message-passing and cryptographic signature verification among a known validator set, and are therefore tagged *key_resource* = computational, which is why they correctly pass the hard filter in Scenario G (Private/Computational) despite being described narratively as a distinct BFT family. Similarly, Proof of Space-Time (PoST) combines a storage-based capacity commitment with a temporal, stake-weighted validation history. It is therefore tagged as compatible with both the storage-based and stake-based *key_resource* categories in the knowledge base, which is why it can legitimately satisfy the hard filter in both Scenario D (Storage) and Scenario I (Stake). This dual-compatibility tagging has been made an explicit field in the knowledge base schema (Section 4.2) rather than left implicit, so that every recommendation returned by the system can be traced to the specific *key_resource* tag(s) that allowed it to pass hard filtering for that scenario.

Scenario G illustrates the behaviour in private network environments with computational resource and high security requirements. The system returned HBBFT, PBFT, and HotStuff as the three highest-ranked mechanisms. PBFT and HotStuff are well-established, leader-based BFT choices for permissioned blockchain environments prioritizing deterministic finality over decentralization. HBBFT (HoneyBadgerBFT) ranks first in this scenario because, unlike its leader-based counterparts, it requires no timing assumptions and tolerates an adaptive network adversary without relying on network synchrony, which the knowledge base records as a higher-security configuration for the security level requested in this scenario.

Scenario D evaluated storage-based resource selection in a public network. The returned mechanisms, PoC, PoSpace, and PoST, represent the primary storage-based consensus approaches within the knowledge base, confirming that the mandatory filtering phase correctly isolated this subset while the scoring phase ranked them according to the specified efficiency and decentralization preferences.

Across all twelve scenarios, the mandatory filtering phase correctly eliminated all mechanisms incompatible with the specified network type and key resource constraints. No instance of a filtered mechanism appearing in the results was observed. The scoring phase produced consistent rank ordering, with tied scores retaining the order in which mechanisms were returned from the underlying database query, a consequence of the stable sort applied during ranking rather than an explicit secondary ranking criterion. The complete per-scenario candidate sets, item-level criterion scores, and final weighted scores underlying every entry in Table 2 are provided in Supplementary Material, Table S1 sheet Full_Candidate_Sets, allowing each reported ranking to be independently reproduced from the knowledge base values in Table S1, sheet KnowledgeBase_S1.

5.1. Knowledge Base Coverage Analysis

A coverage analysis across the twelve evaluation scenarios revealed that 27 of the 38 knowledge base mechanisms (71.1%) appeared in at least one Top-3 recommendation across the tested scenario set. This figure was obtained by taking the union of all mechanisms listed in the Top-3 Recommendations column of Table 2 (27 distinct mechanisms across the 12 scenarios) and dividing by the 38 mechanisms in the knowledge base. Eleven mechanisms received no Top-3 recommendations in the tested scenarios, attributable to their association with highly specific constraint combinations not represented in the current

scenario set rather than algorithmic deficiency. These mechanisms remain accessible in scenarios with the corresponding mandatory constraints and preference profiles.

Mechanisms with the broadest recommendation coverage were those associated with the public stake-based category, which encompasses the largest number of knowledge base entries and corresponds to the most deployed contemporary consensus paradigm. BFT-family mechanisms demonstrated consistent recommendations across private network scenarios, reflecting their well-established role in permissioned blockchain architectures. Storage-based mechanisms formed a consistently returned cluster in storage resource scenarios, indicating that the filtering phase correctly isolated this specialized subset.

5.2. Comparative Analysis Using TOPSIS

The same set of twelve evaluation scenarios presented in Table 2 was used to ensure direct comparability between the proposed system and a TOPSIS-based ranking approach. For each scenario, the identical candidate consensus mechanisms and evaluation criteria were considered: energy efficiency, scalability, security, decentralization, and transaction speed.

Since the knowledge base defines attributes qualitatively, a numerical mapping was applied to enable TOPSIS computation: Low = 1, Medium = 2, High = 3. This mapping encodes the ordinal attribute level of each mechanism and is distinct from the preference direction requested by the user in each scenario. For criteria in which the scenario specifies a High (or Medium) preference, the criterion was treated as a benefit criterion in the TOPSIS ideal-solution and distance calculations (i.e., higher mapped values are preferred). For criteria in which the scenario specifies a Low preference (e.g., Low efficiency in Scenario B, Low decentralization in Scenario F, Low efficiency and Low speed in Scenario K, Low decentralization in Scenario L), the criterion was instead treated as a cost criterion (i.e., lower mapped values are preferred), so that the positive and negative ideal solutions correctly reflect the direction of the user's stated preference rather than a fixed "higher is always better" rule. In alignment with the current implementation of the proposed system, equal weights were assigned to all criteria ($w = 0.2$ each), ensuring a fair baseline comparison.

For each scenario, the candidate set was first filtered according to the mandatory network type and key resource constraints, replicating the first phase of the proposed system. TOPSIS was then applied to the filtered set using standard vector normalization, weighted scoring, and distance-based ranking. The resulting Top-3 rankings were compared with those produced by the proposed expert system.

Table 3 presents the comparative results across all twelve evaluation scenarios. The System Top-3 column lists the recommendations produced by the proposed expert system, while the TOPSIS Top-3 column lists the top-ranked mechanisms produced by the TOPSIS method. The Match column reports the number of mechanisms appearing in both Top-3 sets (regardless of order), and the Agreement column indicates full agreement ($\checkmark$), partial agreement ($-$), or disagreement ($\times$).

The results indicate a high degree of alignment between the two approaches. In 10 out of 12 scenarios (83.3%), at least two of the three mechanisms recommended by the proposed system were also present in the TOPSIS Top-3 set. In 4 scenarios (33.3%), all three mechanisms fully matched the TOPSIS rankings, with minor differences in ordering attributable to methodological differences. Full agreement (3/3 matches) was observed in Scenarios A, D, F, and J. Partial agreement (2/3 matches) was observed in Scenarios C, E, G, I, K, and L. Disagreement (0–1/3 matches) occurred in Scenarios B and H.

To complement the match-count comparison above with a quantitative statistical measure, Spearman's rho and Kendall's tau were computed between the System Top-3 and TOPSIS Top-3 rank orderings reported in Table 3, restricted to the subset of mechanisms

common to both lists in each scenario (rank correlation is undefined for scenarios with fewer than two mechanisms in common). Under this restriction, correlations could be computed for 10 of the 12 scenarios (A, C, D, E, F, G, I, J, K, L). Scenarios B and H had at most one mechanism in common between the two lists and were excluded. Across these 10 scenarios, the mean Spearman's rho was 0.40 and the mean Kendall's tau was 0.40. Seven scenarios (A, D, F, I, J, K, and L) showed perfect rank agreement ($\rho = \tau = 1.0$) on their common mechanisms, while three scenarios (C, E, and G) showed a complete rank reversal ($\rho = \tau = -1.0$) on the two mechanisms they had in common, indicating that where the two methods agree on set membership, they frequently disagree on relative order within that shared subset. We note that this restricted correlation is computed over Top-3 lists rather than full candidate rankings. The complete post-filtering candidate sets for every scenario are provided in Supplementary Material, Table S1 sheet Full_Candidate_Sets, and a full-list rank correlation using these complete candidate rankings is identified as a more rigorous alternative for future evaluation.

Table 3. Comparative evaluation of the proposed expert system and TOPSIS across twelve recommendation scenarios.

Sc.	Network/Resource	Key Preferences	System Top-3	TOPSIS Top-3	Match	Agr.
A	Public/Stake	High efficiency, scalability, decentralization	PoS, PoH, Casper	PoS, PoH, Casper	3/3	✓
B	Public/Stake	High security, decentralization, low efficiency	Casper, PoAc, NPoS	PoS, PoH, Casper	1/3	✗
C	Public/Computational	Medium scalability, high security, low efficiency	PoW, PoW-NG, Ethash	PoW-NG, RBFT, PoW	2/3	–
D	Public/Storage	High efficiency, decentralization, medium scalability	PoC, PoSpace, PoST	PoC, PoSpace, PoST	3/3	✓
E	Public/Hybrid	High scalability, speed, medium security	DAG, PoA, Hashgraph	Hashgraph, PoP, DAG	2/3	–
F	Private/Stake	High speed, scalability, low decentralization	PoA, DPoS-Private, SPoS	PoA, DPoS-Private, SPoS	3/3	✓
G	Private/Computational	High security, medium scalability	HBBFT, PBFT, HotStuff	PBFT, Raft, HBBFT	2/3	–
H	Private/Hybrid	High speed, security, medium efficiency	Istanbul BFT, QBFT, Clique	Hashgraph, Istanbul BFT, PoA	1/3	✗
I	Public/Stake	High efficiency, scalability, speed	PoS, PoH, DPoS	PoS, PoH, Casper	2/3	–
J	Private/Stake	High decentralization, medium security, speed	PoET, PoA, DPoS-Private	PoET, PoA, DPoS-Private	3/3	✓
K	Public/Computational	High security, low efficiency, speed	PoW, Ethash, SHA-256	PoW, PoW-NG, SHA-256	2/3	–
L	Private/Computational	Medium security, high speed, low decentralization	PBFT, Raft, Paxos-BFT	PBFT, Raft, HBBFT	2/3	–

To assess sensitivity to the equal-weighting assumption, we recomputed the weighted score $S(m)$ for the mechanisms named in each scenario's Top-3 (Table 2) under two alternative, non-equal weighting schemes: a security-prioritized scheme and an energy-efficiency-prioritized scheme, in each case assigning the prioritized criterion a substantially larger share of the total weight (tested up to $20\times$ the weight of the remaining criteria).

Because both prioritized criteria are drawn from the same pool of five scoring attributes, a reweighting scheme could only be exercised in each scenario when the prioritized criterion (security or energy efficiency) was among the three criteria specified by that scenario. This held for 11 of the 12 scenarios. Scenario F was the sole exception: its specified criteria (transaction speed, scalability, decentralization) include neither security nor energy efficiency, so the three-way tie among PoA, DPoS-Private, and SPoS reported for that scenario was not exercised by either reweighting scheme or its robustness remains unverified by this analysis. Among the 11 testable scenarios, the top-ranked mechanism (or, where tied under equal weights, the tied group as a whole) remained unchanged up to $20\times$ reweighting in 10 scenarios. The exception was Scenario J: under equal weights, PoET is the unique top-ranked mechanism with no tie, but once the security weight reaches approximately $5\times$ that of the other two criteria, PoET is overtaken and the top position shifts to a three-way tie among PoA, DPoS-Private, and SPoS. We note that the stability observed in the other 10 scenarios partly reflects configurations in which the top-ranked mechanism already attains the maximum possible criterion score ($s_i(m) = 1$) across all specified criteria simultaneously, a configuration that no reweighting scheme can displace. In most other scenarios, the margin between the top mechanism and its runners-up was large enough to persist under substantial reweighting, though Scenario J demonstrates that this is not universally the case, particularly where the top-ranked mechanism's advantage rests on a single criterion outside the one being reweighted. This suggests the Top-1 recommendation is robust to reasonable departures from equal weighting in most, but not all, of the tested scenarios, and we do not generalize this finding beyond the 12 evaluated scenarios.

In Scenario A (public, stake-based, high efficiency and decentralization), both methods produced identical Top-3 rankings: PoS, PoH, and Casper. In Scenario G (private, computational, high security), the proposed system ranked HBBFT first based on direct attribute matching against the requested security level, followed by PBFT and HotStuff, while TOPSIS ranked PBFT and Raft above HBBFT, since TOPSIS evaluates overall closeness to the ideal solution across all five criteria rather than matching the single criterion emphasized by the scenario. In Scenario B, divergence arose because TOPSIS ranked mechanisms by overall closeness to the ideal solution across all criteria simultaneously, assigning higher scores to PoS and PoH whose strong efficiency and decentralization attributes dominated, whereas the proposed system weighted the security and decentralization constraints specified by the user more heavily, by design, rather than balancing them against efficiency. This reflects a difference in ranking philosophy between the two methods rather than evidence that one ranking is objectively correct, since both rely on the same author-defined knowledge base.

The observed alignment between the proposed system and TOPSIS rankings demonstrates internal consistency between the two ranking strategies rather than establishing the objective correctness of either: since both methods draw on the same author-defined knowledge base and, in this comparison, largely overlapping criteria, agreement between them indicates that the additive attribute-matching approach behaves coherently relative to an established MCDM technique, not that its recommendations are externally validated. Despite the simplicity of the additive attribute-matching approach, the expert system produces results that are broadly consistent with a well-established MCDM technique across a diverse set of scenarios. Importantly, the proposed system differs from TOPSIS in its two-phase architecture, where mandatory constraint filtering precedes scoring. This ensures that only contextually valid mechanisms are evaluated, reducing the decision space and improving interpretability. In contrast, TOPSIS operates on the provided alternative set without inherently enforcing strict constraint exclusion prior to scoring.

Discrepancies in Scenarios B, G, and H can be attributed to a fundamental methodological difference. TOPSIS computes geometric distance from the ideal solution across all criteria simultaneously, which means that strong performance on unconstrained criteria can elevate a mechanism's ranking even when it does not align with the user's stated priorities. The proposed system, by contrast, applies direct matching against user-specified preferences, giving explicit weight to the stated importance of each criterion. This distinction makes the proposed system more transparent and directly interpretable in practical deployment contexts.

The comparative results demonstrate that the proposed expert system achieves performance broadly comparable to a standard MCDM approach while maintaining a simpler and more interpretable decision process. The alignment with TOPSIS provides indirect validation of the knowledge base structure and attribute assignments, confirming that the qualitative classifications are sufficiently consistent to support meaningful decision-making across a diverse range of operational scenarios.

This comparative analysis assumes equal weighting of criteria in both approaches. Future work should extend this evaluation by incorporating weighted TOPSIS models calibrated through domain expert elicitation or derived from historical user interaction data. Additionally, applying other MCDM methods such as AHP could provide further validation of the system's robustness across different decision-making frameworks.

5.3. Comparative Analysis Using Existing Tools

Blockbench, Hyperledger Caliper, AccountEnd, and CryptoCompare were selected as comparison points because they are among the most widely referenced, publicly documented tools addressing, respectively, blockchain performance benchmarking (Blockbench, Hyperledger Caliper), consensus-related market and protocol information aggregation (CryptoCompare), and account- and protocol-level blockchain data access (AccountEnd), and together span the range of publicly available tooling most closely adjacent to consensus mechanism selection support. The comparison dimensions in Table 4 were defined operationally as follows: personalized recommendation denotes the ability to return a ranked, user-specific set of suggested mechanisms. Mandatory constraint filtering denotes hard exclusion of mechanisms based on binding user requirements, attribute-based scoring denotes a documented, multi-criteria ranking procedure, decision-support focus denotes an explicit orientation toward pre-deployment selection rather than post-deployment monitoring and the remaining dimensions denote the corresponding technical or interface capability. Each Yes/No/Partial judgment reflects publicly available documentation and product descriptions for each tool at the time of writing. Table 4 presents a structured comparison of the proposed expert system against representative existing tools across key functional dimensions relevant to consensus mechanism selection support.

Table 4. Comparative Analysis of the Proposed System and Existing Tools.

Feature	Proposed System	Blockbench	Hyperledger Caliper	AccountEnd	CryptoCompare
Personalized recommendation	Yes	No	No	No	Partial
Mandatory constraint filtering	Yes	No	No	No	No
Attribute-based scoring	Yes	No	No	No	No
Decision-support focus	Yes	No	No	Yes	Partial
Performance benchmarking	No	Yes	Yes	No	No
Knowledge base management	Yes	No	No	No	No
User history tracking	Yes	No	No	No	No
Web-based interface	Yes	No	No	Yes	Yes
Open-source extensibility	Yes	Yes	Yes	No	No

Within the four tools compared, the analysis indicates that the proposed system addresses a functional combination that is not covered by any single one of them. Blockbench and Hyperledger Caliper deliver comprehensive performance benchmarking capabilities but provide no decision-support functionality for initial mechanism selection. AccountEnd and CryptoCompare offer accessible browsing and partial filtering of consensus-related information but do not implement structured recommendation logic based on user-defined operational requirements. Among this compared set, the proposed expert system is the only tool that combines mandatory constraint filtering, attribute-based scoring, recommendation history management, and extensible knowledge base maintenance within a unified web-based interface. We do not claim this combination is unique across the entire tool landscape, since tools outside this comparison were not evaluated.

5.4. System Deployment, Input Handling, and Access Control

The system was implemented and evaluated in a local development environment consisting of Node.js (v22.19) running the Express.js (v5.1.0) backend server on port 3001, a React (v18) frontend served via Vite on port 5173, and MongoDB (v8.0) as the persistence layer, connected through Mongoose. Cross-origin communication between the frontend and backend is explicitly configured via CORS middleware restricted to the frontend origin. Robustness of the recommendation endpoint was verified with respect to two specific edge cases required for practical deployment. First, when the mandatory filtering constraints (network type and key resource) exclude every mechanism in the knowledge base, the system does not return a default or fallback recommendation. Instead, the /recommendations endpoint explicitly returns an HTTP 404 response with the message “No mechanisms match the specified criteria”, allowing the client to distinguish an empty result from a server error. Second, all controller functions (authentication, recommendation, and administrative CRUD operations) wrap their logic in try-catch blocks, malformed or incomplete requests—for example, a registration request missing the required password field are caught before reaching the database layer and result in a descriptive HTTP error response (400 or 500 with an explanatory message) rather than an unhandled server exception.

Administrative functionality (adding, updating, and deleting consensus mechanisms) is currently gated at the presentation layer: the isAdmin flag returned at login determines whether administrative navigation links and forms are rendered in the React frontend, and client-side routing (RequireAuthorization) prevents unauthenticated navigation to protected pages. At present, the corresponding backend routes (/add-mechanism, /update-mechanism/:id, /delete-mechanism/:id) do not independently enforce this restriction at the API level. Because these endpoints can add, modify, or delete entries in the knowledge base that directly drives every subsequent recommendation, server-side authentication and authorization middleware on these routes should be regarded as a prerequisite for, rather than an optional enhancement to, any production deployment, and is identified as a planned hardening step prior to such deployment. As an interactive decision-support tool used in a single-session, single-user advisory context rather than a real-time or high-throughput production system, formal concurrency and load testing was not considered proportionate to the system’s intended use case. Basic functional verification confirmed that the recommendation endpoint returns a response in approximately 17–50 ms for a single request under the current knowledge base size (38 mechanisms), consistent with interactive use. Systematic load and concurrency testing is identified as relevant primarily for future deployments targeting multi-tenant or high-concurrency scenarios, which fall outside the current single-user advisory scope.

6. Discussion

The proposed expert system demonstrates several strengths that contribute to its practical utility as a decision-support tool for blockchain consensus mechanism selection. The two-phase recommendation architecture provides a transparent and interpretable recommendation process: users can directly understand why specific mechanisms were recommended by examining the attribute matching results, distinguishing it from black-box recommendation approaches that provide recommendations without accessible justification.

The administrative knowledge base management interface represents a significant practical advantage. The blockchain consensus landscape evolves rapidly, with new protocols proposed and existing protocols updated on a continuous basis. The ability to extend and modify the knowledge base through a web interface without requiring source code modification ensures that the system can remain current without imposing a technical barrier on domain experts responsible for maintaining the knowledge base.

The MERN stack implementation provides a production-relevant technology foundation with well-established ecosystem support, facilitating potential deployment in organizational contexts. The RESTful API architecture enables potential integration with other organizational tools and workflow systems, extending the utility of the recommendation functionality beyond the standalone web interface.

The comparative evaluation against the TOPSIS method provides quantitative validation for the proposed expert system. The high level of alignment observed across diverse scenarios confirms that the system's attribute-based scoring model effectively captures the relative suitability of consensus mechanisms. Furthermore, the identified discrepancies in specific scenarios highlight the distinct advantage of the system's two-phase architecture: by prioritizing strict constraint filtering, the system ensures that recommendations remain contextually valid, preventing the mathematical dilution of user-specified requirements that can occur in purely distance-based MCDM models.

6.1. Limitations

Several limitations of the current implementation warrant acknowledgment. The scoring model assigns equal weight to all five evaluated preference criteria, which may not accurately reflect real-world organizational priorities in which certain characteristics, such as security in financial applications or energy efficiency in sustainability-focused deployments, carry substantially greater importance than others. The equal-weight assumption represents a practical simplification rather than an empirically grounded design choice.

The knowledge base attribute values are manually assigned based on qualitative classification from literature review and available documentation, rather than derived from quantitative benchmarking measurements. This introduces a degree of subjectivity in attribute assignments, particularly for ordinal classifications such as medium scalability or high security, where the precise threshold definitions may vary across sources. The absence of quantitative grounding limits the precision of recommendations in contexts where fine-grained performance differentiation is required.

The current evaluation was performed through structured scenario testing rather than formal user studies with domain practitioners. It is important to acknowledge explicitly that the twelve evaluation scenarios, the knowledge base, and the expected recommendation behaviour used to judge correctness were all defined by the authors. The evaluation therefore demonstrates the internal consistency of the recommendation algorithm rather than externally validated effectiveness in practical deployment. While scenario testing provides systematic coverage of algorithmic behaviour, it does not capture the usability characteristics of the interface or the perceived utility of recommendations from the perspective of target users, including blockchain developers, system architects, and organizational

decision-makers. Expert-panel evaluation and studies involving blockchain practitioners on representative real-world use cases, together with a formal usability evaluation using an instrument such as the System Usability Scale (SUS), are identified as necessary next steps before claims of practical effectiveness can be made.

The knowledge base, while comprehensive at 38 mechanisms, does not capture all consensus protocols described in the research literature, and newly emerging protocols are absent until manually added. The coverage of hybrid and novel consensus paradigms may be incomplete relative to the current state of the art.

6.2. Future Work

Several directions for future development are identified based on the current limitations and evaluation findings. The introduction of weighted multi-criteria scoring models represents the most impactful near-term improvement. Weights could be elicited through structured domain expert surveys using established MCDM methodologies such as AHP or through analysis of historical user selection patterns once sufficient usage data has been accumulated. Weighted scoring would enable the system to more accurately reflect the relative importance of different criteria across diverse deployment contexts.

Integration with quantitative benchmarking data obtained from frameworks such as Hyperledger Caliper would enable attribute values in the knowledge base to be grounded in measured performance metrics rather than qualitative classifications. This would improve recommendation precision and enable finer-grained differentiation between mechanisms with similar qualitative profiles.

A formal usability evaluation involving practitioners with blockchain development and architectural planning experience would provide empirical validation of the system utility and identify interface improvements. Usability assessment using standardized instruments such as the System Usability Scale (SUS) would complement the algorithmic evaluation presented in this paper.

Machine learning-assisted recommendation techniques, potentially incorporating collaborative filtering based on historical user selections and outcomes, could extend the system beyond static knowledge base matching toward adaptive recommendations informed by accumulated usage experience. Token-based authentication mechanisms and enhanced session management would be required to support production-scale deployment. Expansion of the knowledge base to incorporate recently proposed consensus protocols and improved coverage of hybrid and emerging paradigm families represents a continuous maintenance requirement.

7. Conclusions

This paper presented a MERN-stack web application functioning as an expert system for blockchain consensus mechanism selection. The proposed system accepts up to seven user-defined criteria, applies a two-phase filtering and scoring algorithm over a database containing 38 consensus mechanisms, and returns the three most suitable candidates together with their key characteristics. The implemented administrative interface enables extensibility of the mechanism database without requiring direct source code modification, thereby improving system maintainability and adaptability to the rapidly evolving blockchain ecosystem.

Experimental evaluation through twelve representative recommendation scenarios demonstrated that the two-phase algorithm consistently produces contextually relevant consensus mechanism recommendations. The mandatory filtering phase correctly eliminated incompatible mechanisms across all tested constraint combinations, and the attribute-based scoring phase provided consistent and interpretable rank ordering of candidate mech-

anisms. Coverage analysis confirmed that 71.1% of knowledge base mechanisms were represented in Top-3 recommendations across the tested scenario set. Comparative analysis against four representative existing tools indicated that the proposed system combines a set of decision-support capabilities not jointly offered by any one of them, within the compared set. Additionally, a comparative evaluation against the TOPSIS method demonstrated that the system produces results consistent with established MCDM frameworks, while providing superior interpretability through its constraint-first recommendation process.

The presented approach contributes a transparent, extensible, and practically deployable decision-support framework addressing a gap in existing blockchain tooling. While current limitations in scoring model weighting and qualitative attribute grounding represent areas for future improvement, the system provides a functional and accessible mechanism-selection resource capable of supporting architectural and organizational decision-making in distributed systems environments.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/network6030056/s1>, Dataset Table S1: Extended consensus mechanism knowledge base and per-scenario score reconstruction. Table S1 is provided as a single spreadsheet workbook comprising three sheets: KnowledgeBase_S1, listing the classification family, key_resource filtering tag, network type, five attribute values, confidence label and descriptive summary for all 38 consensus mechanisms in the knowledge base; Table2_Scenarios, listing the twelve evaluation scenarios with their mandatory constraints, stated criterion preferences and resulting Top-3 recommendations as reported in Table 2; and Full_Candidate_Sets, listing the complete post-filtering candidate set for each scenario with item-level criterion scores, final weighted scores $S(m)$ and resulting ranks.

Author Contributions: Conceptualization, N.R.; methodology, M.Š. and I.L.; software, N.R.; validation, N.R., I.L. and M.Š.; formal analysis, N.R. and I.L.; writing—original draft preparation, I.L., M.Š. and N.R.; writing—review and editing, I.L. and M.Š.; supervision, I.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded and is a part of research for the activities to be carried out for the project “Researching advanced algorithms and innovative business intelligence solutions in the cloud-NPOO.C3.2.R3-I1.04.0128”.

Data Availability Statement: All data supporting the findings of this study are contained within the article and its Supplementary Materials. The complete knowledge base, the twelve evaluation scenarios and the full per-scenario candidate sets with item-level and final weighted scores are provided in Table S1. No new data were created beyond those reported.

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

Abbreviations

The following abbreviations are used in this manuscript:

AHP	Analytical Hierarchy Process
API	Application Programming Interface
BFT	Byzantine Fault Tolerance
CRUD	Create, Read, Update, Delete
DAG	Directed Acyclic Graph
DPoS	Delegated Proof of Stake
IoT	Internet of Things
JSON	JavaScript Object Notation
MCDM	Multi-Criteria Decision Making
MERN	MongoDB, Express.js, React, Node.js

NPoS	Nominated Proof of Stake
PBFT	Practical Byzantine Fault Tolerance
PoA	Proof of Authority
PoAc	Proof of Activity
PoC	Proof of Capacity
PoET	Proof of Elapsed Time
PoH	Proof of History
PoS	Proof of Stake
PoSpace	Proof of Space
PoST	Proof of Space and Time
PoW	Proof of Work
REST	Representational State Transfer
SUS	System Usability Scale
TOPSIS	Technique for Order of Preference by Similarity to Ideal Solution
UI	User Interface

References

1. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In Proceedings of the 2017 IEEE International Congress on Big Data, Boston, MA, USA, 11–14 December 2017; pp. 557–564.
2. Tapscott, D.; Tapscott, A. *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World*; Portfolio/Penguin: New York, NY, USA, 2016.
3. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3440802 (accessed on 21 April 2026).
4. Svarcmajer, M.; Kohler, M.; Krpic, Z.; Lukic, I. Consensus-Driven Framework for Data-Driven Optimization of Distributed Systems Through Blockchain Consensus Mechanism Selection. *Big Data Cogn. Comput.* **2026**, *10*, 154. [CrossRef]
5. Xu, X.; Weber, I.; Staples, M.; Zhu, L.; Bosch, J.; Bass, L.; Pautasso, C.; Rimba, P. A Taxonomy of Blockchain-Based Systems for Architecture Design. In Proceedings of the 2017 IEEE International Conference on Software Architecture, Gothenburg, Sweden, 3–7 April 2017; pp. 243–252.
6. Bamakan, S.M.H.; Motavali, A.; Babaei Bondarti, A. A Survey of Blockchain Consensus Algorithms Performance Evaluation Criteria. *Expert Syst. Appl.* **2020**, *154*, 113385. [CrossRef]
7. Dinh, T.T.A.; Wang, J.; Chen, G.; Liu, R.; Ooi, B.C.; Tan, K.-L. BLOCKBENCH: A Framework for Analyzing Private Blockchains. In Proceedings of the 2017 ACM International Conference on Management of Data; ACM: New York, NY, USA, 2017; pp. 1085–1100.
8. Kaushal, R.K.; Kumar, N. Exploring Hyperledger Caliper Benchmarking Tool to Measure the Performance of Blockchain Based Solutions. In Proceedings of the 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO), Noida, India, 14–15 March 2024; pp. 1–6.
9. Blockchain Consensus Mechanism Explorer—Accountend. Available online: <https://accountend.com/blockchain-consensus-mechanism-explorer/> (accessed on 10 April 2026).
10. CryptoCompare. Available online: <https://www.cryptocompare.com/> (accessed on 21 April 2026).
11. Turban, E.; Aronson, J.E. *Decision Support Systems and Intelligent Systems*, 6th ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2001.
12. Jackson, P. *Introduction to Expert Systems*, 3rd ed.; Addison Wesley: Harlow, UK, 1999.
13. King, S.; Nadal, S. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. 2012. Available online: <https://decred.org/research/king2012.pdf> (accessed on 21 April 2026).
14. Castro, M.; Liskov, B. Practical Byzantine Fault Tolerance. In Proceedings of the 3rd Symposium on Operating Systems Design and Implementation, New Orleans, LA, USA, 22–25 February 1999; pp. 173–186.
15. Yakovenko, A. Solana: A New Architecture for a High Performance Blockchain v0.8.13. 2018. Available online: <https://coincode-live.github.io/static/whitepaper/source001/10608577.pdf> (accessed on 21 April 2026).
16. Popov, S. *The Tangle*; IOTA Foundation: Berlin, Germany, 2018.
17. Cohen, B.; Pietrzak, K. *The Chia Network Blockchain*; Chia Network: South San Francisco, CA, USA, 2019.
18. Vukolic, M. The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication. In *Open Problems in Network Security*; Springer: Cham, Switzerland, 2016; pp. 112–125.
19. Cachin, C.; Vukolic, M. Blockchain Consensus Protocols in the Wild. *arXiv* **2017**, arXiv:1707.01873.
20. Saaty, T.L. *The Analytic Hierarchy Process*; McGraw-Hill: New York, NY, USA, 1980.
21. Hwang, C.L.; Yoon, K. *Multiple Attribute Decision Making: Methods and Applications*; Springer: Berlin, Germany, 1981.
22. Shortliffe, E.H.; Buchanan, B.G. A Model of Inexact Reasoning in Medicine. *Math. Biosci.* **1975**, *23*, 351–379. [CrossRef]

23. Giarratano, J.C.; Riley, G.D. *Expert Systems: Principles and Programming*, 4th ed.; Course Technology: Boston, MA, USA, 2005.
24. Filatovas, E.; Marcozzi, M.; Mostarda, L.; Paulavičius, R. A MCDM-based framework for blockchain consensus protocol selection. *Expert Syst. Appl.* **2022**, *204*, 117609. [[CrossRef](#)]
25. Ospanov, R.; Tashatov, N.; Satybaldina, D.; Seitzkulov, Y.; Yergaliyeva, B.; Utebayev, K. Application of MCDM Methods to Optimal Consensus Protocol Selection for Blockchain-Based IoT Networks. *Appl. Sci.* **2025**, *15*, 6898. [[CrossRef](#)]
26. MERN Stack Explained. Available online: <https://www.mongodb.com/resources/languages/mern-stack> (accessed on 21 April 2026).
27. Bentov, I.; Lee, C.; Mizrahi, A.; Rosenfeld, M. Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake. *ACM SIGMETRICS Perform. Eval. Rev.* **2014**, *42*, 34–37. [[CrossRef](#)]

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