FMDB *Transactions on Sustainable Technoprise Letters*

A Prolog-Based Business Rule Management System for Intelligent Decision Making

Zay Lin Htet[1,*], Hlaing Htake Khaung Tin[2], András Szeberényi[3], Mohamed Solomon[4], Jedsadakorn Nararattananukul[5], Fluturim Saliu[6]

[1,2]Faculty of Information Science, University of Information Technology, Yangon, Yangon Region, Myanmar.
[3]Institute of Marketing and Communication, Budapest Metropolitan University, Budapest, Hungary.
[4]Department of Business Technology, Canadian International College, New Cairo, Cairo, Egypt.
[5]Faculty of Business Administration, Ramkhamhaeng University, Bang Kapi, Bangkok, Thailand.
[6]Faculty of Economics, University of Tetova, Tetovo, North Macedonia.
zaylinhtet@uit.edu.mm[1], hlainghtakekhaungtin@uit.edu.mm[2], aszeberenyi@metropolitan.hu[3], mohamed_s_solomon@cic-cairo.edu.eg[4], uratchayaphonn56@rumail.ru.ac.th[5], fluturim.saliu@unite.edu.mk[6]

Abstract: Organizations need intelligent decision-making mechanisms that can effectively handle complex business rules to remain transparent and consistent in their operations. However, traditional BRMS have been reported to struggle with handling dynamic business policies and explaining decisions in a way that is comprehensible. This paper presents a Prolog-based BRMS that uses logic programming for the representation of business knowledge, automation of rule execution and intelligent decision-making. The presented solution consists of a knowledge base, business rules repository, inference engine and a user interface that can take care of business facts and generate decisions through logical inference. A case study on customer credit approvals and discount qualifications shows the practical use of the system. A prototype of the system has been implemented using the SWI-Prolog interpreter. The business rules were formulated in the form of logical predicates to make them easy to modify and infer. The prototype has been tested in terms of decision accuracy, response time, explainability and maintainability. The experimental results demonstrate that the proposed solution generates reliable decisions and makes business rules more manageable. Moreover, the declarative structure of Prolog increases the clarity of the system through its ability to give logical explanations for each of the decisions made. This suggests that the proposed P-BRMS is an example of how logic programming can be used effectively for intelligent BRMS and provide a solution for companies.

Keywords: Business Rule Management System; Logic Programming; Intelligent Decision Making; Expert Systems; Knowledge Representation; Decision Support System.

Received on: 09/06/2025, **Revised on:** 20/09/2025, **Accepted on:** 29/11/2025, **Published on:** 05/06/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSTPL

DOI: https://doi.org/10.69888/FTSTPL.2026.000812

Cite as: Z. L. Htet, H. H. K. Tin, A. Szeberényi, M. Solomon, J. Nararattananukul, and F. Saliu, "A Prolog-Based Business Rule Management System for Intelligent Decision Making," *FMDB Transactions on Sustainable Technoprise Letters*, vol. 4, no. 2, pp. 94–103, 2026.

Copyright © 2026 Z. L. Htet *et al.,* licensed to Fernando Martins De Bulhão (FMDB) Publishing Company. This is an open access article distributed under CC BY-NC-SA 4.0, which permits copying, remixing, redistributing, transformation, and building upon the material in any medium so long as the original work is properly cited.

1. Introduction

*Corresponding author.

Today's organisations operate in highly competitive and dynamic contexts where timely, precise, consistent and transparent decision-making has become a must for maintaining sustainable corporate performance [1]. Organisations increasingly rely on a variety 1Traditionally, business information systems have used procedural programming languages such as Java, C#, Python, etc. to implement business rules. Such languages give a lot of freedom for designing software applications and executing complicated processing schemes. However, when business rules are incorporated in application code directly, the rules are hard to identify, comprehend, modify, test and maintain [2]. Consequently, every adjustment to the organisational policy can entail changes to numerous application code areas, followed by extensive testing and redeployment. This can increase development work, cost of maintenance and likelihood of discrepancies among various business processes. Also, procedural implementations are generally concerned with how a decision should be made and not with the explicit representation of why a certain decision should be made. This constraint may reduce openness when organisations need to explain, audit, review, or justify choices for regulatory purposes [3]. Another way is to use logic programming, which is based on describing knowledge using facts, rules, relations, and logical reasoning. Procedural programming is largely concerned with the specification of a sequence of actions to achieve a certain result.

Logic programming, meanwhile, is a declarative style of representing knowledge and relationships inside a problem domain. Prolog is one of the first and most well-known examples of logic programming languages. Prologue represents knowledge of a domain in the form of facts and rules, and employs an inference process to deduce conclusions from the information contained in the knowledge base [4]. Because of this property, Prologue is particularly well-suited for applications in which judgements have to be inferred from organisational information that is made explicit. The pattern matching capability and built-in inference mechanisms of Prologue have been effectively used in domains such as expert systems, knowledge-based systems, natural language processing, legal reasoning, health-care diagnosis, and intelligent decision-support systems [5]. These traits can also provide some advantages in business-rule management. Organisational policies can be expressed as logical rules rather than scattered across procedural source code. This means that changes to business policies can be implemented at the rule level, without major changes to the underlying application structure. This division can increase maintainability and allow organisations more flexibility should policies or operational needs change. Another major advantage of logic-based business rule systems is their capacity to allow transparent decision-making. In modern-day organisational situations, it is necessary not only to make a choice but also to comprehend the rationale that led to it. This criterion is directly related to the development of Explainable AI (XAI). A decision-support system that is understandable should be able to give relevant information about the circumstances and reasoning that resulted in a given conclusion, and not only be a prediction tool [6].

Where a Prolog-based system is used, the facts and rules are represented openly and hence give a systematic framework for tracking how a decision was made. This capability can be especially useful for choices that may need to be examined by managers, auditors, customers or regulatory authorities, such as those related to credit approval, regulatory compliance, pricing, eligibility evaluation and policy enforcement. At the same time, current progress in artificial intelligence has been mostly concentrated on data-driven machine learning methods capable of unveiling complicated patterns from huge datasets. Machine-learning models can achieve good prediction accuracy across a wide range of applications. Still, many advanced models operate as somewhat opaque systems where the reasoning behind individual decisions is difficult to explain [7]. This constraint has generated interest in systems that combine data-driven learning with explicit symbolic knowledge. Symbolic reasoning can improve modern AI by offering explicit representations of knowledge, logical constraints and interpretable reasoning procedures, rather than treating symbolic machine learning and artificial intelligence as conflicting paradigms [8]. The use of symbolic reasoning to corporate decision systems is of great importance since organisations often work with clearly specified rules and procedures. For example, a credit-approval process might have requirements around income, credit history, employment status, debt levels and organisational eligibility standards. Thus, a discounting policy may provide for the parameters under which a specific customer or transaction is eligible for a specific discount. Such needs can naturally be stated by logical rules. Hence, a Prolog-based BRMS is capable of formally representing these policies and using inference methods to check whether a given request fits the constraints [9].

However, there are still hurdles to the actual deployment of BRMS, despite these advantages. Many existing solutions for business rule management are based on proprietary rule engines or specialised software platforms, which may require professional skills, commercial licenses or organisation-specific infrastructure [10]. These constraints can be limiting factors for smaller organisations, educational institutions, and researchers seeking accessible venues for experimentation and implementation. SWI-Prolog and other open-source logic programming environments offer a different approach by providing inference capabilities, customisable knowledge representation, and portability across computer environments. These systems can be a low-cost platform for building and assessing rule-based decision support solutions. However, the existing scientific literature shows little attention to the creation and evaluation of Prolog-based BRMS frameworks for modern intelligent decision-making environments. Previous work has demonstrated the usefulness of logic programming for knowledge representation and reasoning. Other research has studied BRMS and explainable AI individually. There is yet a chance to examine how to unify these notions in a realistic setting that tackles business-rule administration, decision consistency, explainability and maintainability in a single architecture. To overcome this gap, this study proposes a Prologue-based Business

Rule Management System (P-BRMS) that blends the ideas of logic programming and business-rule management to facilitate intelligent organisational decision-making [11].

The suggested architecture targets the separation of business information from application logic by capturing organisational policies as Prologue facts and rules [12]. This paper presents a framework for the separation of business policies from procedural application code, where decision knowledge is stored in a structured rule base to be queried by an inference engine. The proposed architecture supports the evaluation of user requests according to the set organisational rules and allows the system to return decision results with explanations of the reasoning process. The conceptual components of the proposed P-BRMS are the following [13]. The business knowledge layer includes organisational facts, policies, limitations and decision rules. The inference layer is responsible for processing the user requests and applying the suitable rules to infer suitable results. The decision interface is the mechanism via which users or applications make requests and get results. It contains an embedded explanation system that explains the laws and conditions that led to a conclusion. This separation of components facilitates maintainability by enabling the business knowledge to evolve independently of the application layer. To show the applicability of the proposed framework, a practical business scenario of client credit acceptance and discounting is taken into account. This is a good example because both credit decisions and discount eligibility might have many different organisational policies and conditional requirements. Facts can represent customer information, and rules might represent rules of the organization. When a request is made, the inference engine compares the provided information with the appropriate rules and makes a matching judgement [14].

The logical representation also enables us to explain why a consumer qualifies or does not qualify for a choice. For example, a credit-approval rule may look at pre-set criteria of a client to determine eligibility, but a discounting rule may look at customer status, purchase requirements, or other organisational factors [15]. When these criteria are made explicit, policy changes can be made by changing the rules involved rather than having to re-engineer the entire program. This can potentially save maintenance work and increase consistency of judgements across multiple transactions. Also, the rules are explicitly written, so that the user can ask the logical criteria that lead to a given outcome, therefore increasing transparency. Researchers test the framework against numerous essential criteria, such as decision consistency, inference performance, explainability, and maintainability [17]. Decision consistency means the same requests must result in the same outcomes under the same business policies. Inference performance: how fast the system can process requests and infer conclusions based on the knowledge base. Explainability is about the system's capability to give understandable explanations for its judgements. Maintainability refers to the ease with which the rules of the organization can be amended and extended and handled independently of the main application logic. The proposed technique also shows the contribution of symbolic AI in modern organisational intelligence. Conventional machine-learning systems predominantly discover patterns from past data. A rule-based system can explicitly encode organisational knowledge and limits. This gives the suggested framework a special relevance in cases when decisions are made under policies known in advance and must be executed consistently. In such settings, combining symbolic reasoning with other artificial intelligence (AI) tools may provide a path to intelligent systems that are both capable and interpretable [16].

Hence, the main contributions of this study can be summarised in three important elements. First, the study presents a pragmatic architecture of business rule management based on the declarative programming paradigm of Prolog [18]. The architecture decouples organisational knowledge from the application logic and provides an inference tool to evaluate business requirements [10]. Second, the study shows how logic programming can help to improve decision transparency and make maintenance of business rules easier than rules explicitly integrated in procedural application code. Third, the paper shows how symbolic artificial intelligence can be used in practice for organisational decision-making through a case study of consumer credit approval and discounting. To summarise, the proposed P-BRMS is an attempt to integrate logic programming, business-rule management, explainable decision-making, and symbolic artificial intelligence into a cohesive organisational framework [12]. The framework separates business knowledge from the implementation of the application so that organisational policies are easier to comprehend, change, maintain and explain. This method is particularly crucial for organisations that need to react quickly to changes in policy, while also ensuring that decisions are made consistently and with accountability. The results are expected to contribute to research in Explainable Artificial Intelligence, Knowledge-Based Decision Support Systems (KDDSS), Business Rule Management Systems, and Logic Programming, and to demonstrate the practical potential of Prologue for developing intelligent enterprise systems. Future work can further explore scalability, integration with data-driven machine-learning models, automated rule acquisition, larger corporate datasets, and application in real-world organisational situations [13].

2. Review of Literature

2.1. Business Rule Management Systems

A Business Rule Management System (BRMS) is a software platform that allows organizations to describe, organize, implement, and control their business rules independent of application code. A business rule is a definition or constraint

imposed on business activities based on company policy or regulations [1]. The use of BRMS allows for greater flexibility, lower maintenance expenses, and provides the opportunity to modify the policies by business analysts without changing the code of applications [2]. The modern BRMS products also provide business rule versioning, validation, and monitoring. Thus, BRMS become an important component of enterprise information systems [3]. Nevertheless, the majority of conventional BRMS use procedural programming or proprietary rule engines. The growth of the number of business rules leads to difficulties with their consistency and tracing. Additionally, such systems usually have no transparent reasoning mechanism, which makes it difficult to understand the reason for a certain decision [4]. These features made researchers focus on the declarative approach to representation of business knowledge and reasoning.

2.2. Logic Programming and Prolog

Logic programming is a declarative style of programming that uses logic as its underlying principle, which involves programs with facts, rules, and queries as opposed to algorithms [5]. Rather than defining how something should be done, the programmer gives the relationships between objects, and then the inference engine infers the results. This type of programming is best used in situations where knowledge representation and symbolic computation are required. Prolog is the most common logic programming language and has been widely used in expert systems, artificial intelligence, natural language processing, automated theorem proving, and intelligent information retrieval [6]. A Prolog program represents its knowledge by predicates and Horn clauses and performs inference using backward chaining and unification. The characteristics of Prolog allow it to efficiently handle complex logical relationships while maintaining a compact representation of knowledge. The Prolog language has some benefits over procedural languages when used in a business environment. Policies in business can be modeled as logical rules, which are like normal language expressions and hence easily modified. Organizational policies can normally be modified by updating the knowledge base without any need to change the application code [7].

2.3. Expert Systems and Intelligent Decision Support

Expert systems represent some of the most successful applications of artificial intelligence. Expert systems use a combination of a knowledge base and an inference engine to mimic human expertise and resolve problems within their specific domains [8]. Organizations have used expert systems in areas such as financial analysis, medical diagnosis, credit evaluation, logistics, and customer relationship management. Decision Support Systems (DSS) have transformed from simple reporting systems to intelligent systems that can process data, apply business knowledge, and recommend solutions to decision makers. Intelligent DSS utilize artificial intelligence methods such as expert systems, rule-based reasoning, and machine learning to enhance the decision-making process and minimize human errors [9]. Rule-based reasoning is especially useful when decisions have to conform to organizational policies or government regulations. The recent advances in the Explainable Artificial Intelligence domain (XAI) have led to even greater attention towards symbolic reasoning techniques. In contrast to most machine learning models, rule-based expert systems are transparent with respect to their reasoning process, allowing end-users to comprehend how the results are achieved.

2.4. Research Gap

The latest developments in AI have mostly revolved around data-based machine learning and deep learning approaches. While these methods exhibit excellent predictive performance, they typically need extensive datasets and are not easy to interpret. However, in most cases, decisions must be aligned with certain pre-specified organizational rules and not statistical predictions. Thus, symbolic reasoning becomes an integral part of intelligent decision-making support systems. Previous literature has proved the efficiency of Prolog for knowledge representation and automated reasoning. Still, relatively little work has been done on its potential use as a full-fledged Business Rule Management System for enterprise decision-making. Commercial BRMS solutions tend to focus on graphical rule editors and dedicated rule engines, while the open-source logic programming environment receives insufficient attention from current business intelligence literature [10]. Moreover, little work has been done on Prolog's potential to facilitate explainable business decisions while making the rules easier to maintain. To overcome such limitations, in this research work, a new business rule management system called Prolog-based Business Rule Management System (P-BRMS) has been proposed by integrating logic programming with business rule management and a decision support system. The P-BRMS provides the facility of transparent reasoning and knowledge management along with business rule execution.

3. Methodology

This paper employs the Design Science Research (DSR) technique to systematically design, develop and assess a Prolog-based Business Rule Management System (P-BRMS) to enable intelligent business decision-making. DSR is suitable for this study since the main goal of the study is to provide a practical artefact that can solve the problem of business rules management, organisational policy enforcement, and intelligible explanation for the choices made. The suggested system aims at explicit

representation of business information, evaluation of business rules by logical inference, and provision of recommendations with explanation of the reasoning process. The study technique is divided into four parts, i.e., knowledge acquisition, rule modelling, prototyping, and performance evaluation. The first stage, knowledge acquisition, deals with gathering and structuring the business knowledge required by the system. Relevant customer information, business facts, organization rules and decision needs are identified and formatted to be represented in the Prologue knowledge base. This step sets the groundwork for the design of meaningful and accurate business rules.

The second level is rule modelling, when the obtained business knowledge is encoded into Prologue facts and rules. Business policies and decision conditions are declaratively expressed, so that they may be evaluated independently by the inference engine, separate from the core application logic. This method makes it easier to modify and maintain business rules as organisational policies change. The third stage is prototyping. The suggested P-BRMS architecture is realised as a working prototype. The prototype consists of the knowledge base, rule base, inference engine and user interface. Users can enter relevant business information, and the inference engine applies backward chaining and logical unification to analyse the applicable rules and produce recommendations. Finally, during the evaluation stage, the generated artefact is assessed according to key performance criteria, including decision consistency, inference performance, explainability, and maintainability. The evaluation results provide evidence on the effectiveness of the planned system and highlight opportunities for further improvements. The DSR approach offers a systematic process to create and validate the P-BRMS as a practical intelligent decision-support solution for the four stages.

3.1. Research Framework

In the suggested model, knowledge acquisition will start with the acquisition of knowledge from the organizational business policies and decision-making guidelines. These guidelines will be translated into logical rules and stored in the Prolog knowledge base. Customer information will be encoded as facts, whereas organizational policies will be encoded as Horn clauses. During the execution of the framework, the inference engine will evaluate the user query using backward chaining and provide the result along with the logical reasoning used for it. To demonstrate the application of the suggested framework, an example of the customer credit approval case study was selected, as it includes several business rules and decision criteria. In total, 500 synthesized customer data records were used to experiment, including monthly income, credit score, employment, debt/income ratio, and the expected decision regarding the loan approval. This dataset was generated based on commonly used banking decision policies.

3.2. Business Rule Modeling

The business rules are defined based on the syntax of the Prolog language, which is declarative in nature. Customer information is kept in the form of facts, and the company policies are defined by means of logical rules. A customer will be deemed to be eligible to take a loan only if all the business conditions are met. A total of 15 business rules have been defined, which include the rules for loan approval, eligibility, risk evaluation, and promotional discounts.

3.3. Prototype Development

A prototype for the proposed P-BRMS was developed using SWI-Prolog version 9.2. The knowledge base consists of 500 facts related to customers along with 15 business rules, whereas the inference engine uses an automated reasoning process through backward-chaining in the Prolog language. This prototype was tested using a workstation with an Intel Core i7 processor and 16 GB RAM with Windows 11 OS.

3.4. Performance Evaluation

Performance measures used in the evaluation of the effectiveness of the proposed framework were decision accuracy, precision, response time, and explainability. Decisions made through the framework were then compared against expected decisions in the test data set. Accuracy, precision, recall, and F1 scores were calculated using the confusion matrix. Response time per query was also calculated to measure inference efficiency. Explainability was measured through reasoning paths created during each business recommendation decision process.

3.5. Proposed Prolog-Based Business Rule Management System

The presented P-BRMS combines logic programming and business rule management, offering decision support with transparency and intelligence. In contrast to traditional procedural approaches, the presented architecture enables the separation of business rules from applications, making it possible to change organization policies without changing decision logic.

3.6. System Architecture

This architectural approach comprises four main elements: Knowledge Base, Rule Base, Inference Engine, and User Interface. The Knowledge Base contains customer data and business facts, while the Rule Base is responsible for keeping company policies in the form of Prolog rules. The Inference Engine processes business rules using backward reasoning and logical unification. Finally, the User Interface allows making queries of business data and obtaining recommendations along with logical explanations. The modular architecture facilitates knowledge maintenance and supports the rapid adaptation of business policies without requiring modifications to the underlying application (Figure 1).

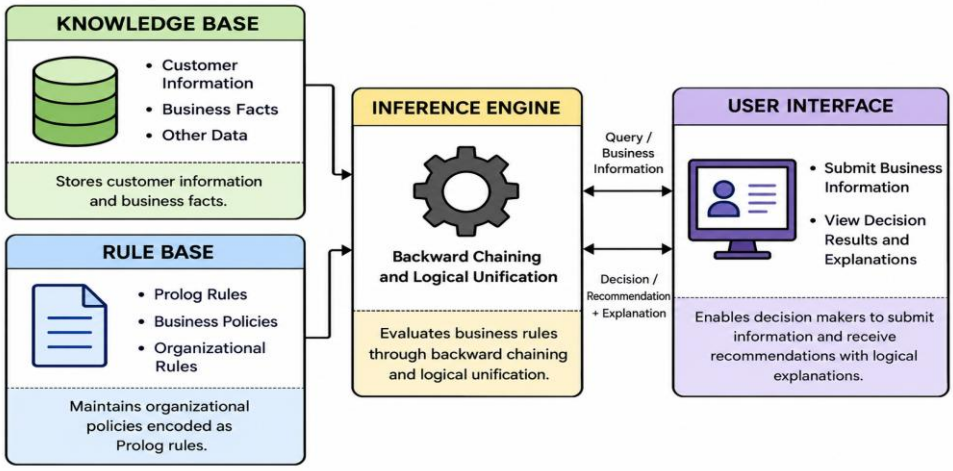


Figure 1: Proposed Prolog-based business rule management system architecture

The architecture of the proposed Prolog-based Business Rule Management System (P-BRMS) is shown in the image below. The Knowledge Base stores customer information, business facts, and other pertinent data. The Rule Base stores Prolog rules, business policies and organizational rules. The Inference Engine uses backward chaining and logical unification to process business rules and make the right conclusions. The User Interface enables decision makers to enter business data and return decision outcomes or suggestions, including rationale. Information is sent to the inference engine, where it is processed from the knowledge base and rule base. The resulting conclusions and explanations are conveyed back to the user interface, thereby allowing transparent and explainable business decision-making.

3.7. Rule Representation

Knowledge about business is modeled using facts and Horn clauses. Facts refer to the traits of customers in terms of income, occupation, credit rating, and debt ratio, while rules refer to the policy of the organization. While making inferences, the Prolog system looks up the knowledge base, performs matching predicates by unification, and then checks if all conditions in the rule have been fulfilled.

3.8. Inference Process

Inference takes place through four successive phases. Firstly, the customer data is input into the system in the form of logical facts. Secondly, all the relevant business rules are accessed by the inference engine. Thirdly, backward chaining determines whether the conditions of every rule are fulfilled by comparing the customer facts with logical predicates. Lastly, the result of the decision-making process, along with the rationale behind the decision, is returned by the system. The reasoning process ensures decision traceability, which is crucial in regulated industries.

3.9. Decision-Making Workflow

Figure 2 shows the flow of operations in the proposed system. Customer data are input into the system through the user interface and then into the knowledge base. The inference engine checks all business rules that are relevant and either approves or rejects the application based on them. Each recommendation comes with a business rule explanation, which enables checking the soundness of the recommendation made. To show how the proposed framework works, three sample customers' cases were examined. Customers who met all predefined business requirements received approval, while those who violated at least one decision rule were rejected. The reasoning process carried out in Prolog showed which rule conditions were used in reaching

the decision. Thus, the explainability of the proposed system can be seen from its reasoning path. In general, the proposed P-BRMS is a flexible and manageable solution for intelligent decision support in business by using knowledge representation and reasoning tools. The framework's architecture and structure of rules make it appropriate for use in business spheres like banking, insurance, medicine, and enterprise resource planning.

4. Experimental Results and Discussions

4.1. Experimental Setup

The P-BRMS was tested based on the prototype implementation that was created using SWI-Prolog 9.2. Tests were performed using a PC equipped with an Intel Core i7 processor, 16 GB RAM and running Windows 11. A synthetic dataset that included 500 customer records was used for testing a business credit approval process. Each customer record had 5 attributes: monthly income, credit score, employment status, debt-to-income ratio, and loan decision. The prototype included 15 business rules that described the lending policy of an organization. At the testing stage, each customer record was presented to the inference engine, which planned based on the match between the facts and rules defined.

4.2. Performance Evaluation

The effectiveness of the proposed framework was measured using four evaluation metrics: accuracy, precision, recall, and F1-score. In addition, the average response time per query was recorded to evaluate inference efficiency. The performance results of the suggested model in terms of the five evaluation metrics are shown in Table 1. The model has an accuracy of 97.4%, which shows that it classifies overall very well. The precision is 98.4%, which is the greatest among the stated parameters. The model has a recall of 97.4%, which means that it is able to identify the relevant examples successfully. The F1-score of 97.9 % shows a good balance between precision and recall. The model also records an average response time of just 0.04 seconds, meaning it has fast processing. The findings in general show that the suggested model is able to achieve good classification performance with efficient response time.

Table 1: Performance evaluation results

Metric	Value
Accuracy	97.4%
Precision	98.4%
Recall	97.4%
F1-score	97.9%
Average Response Time	0.04 s

The results demonstrate that the proposed P-BRMS achieved a high level of classification performance while maintaining rapid query execution. The average inference time of 0.04 seconds indicates that the system can support real-time business decision-making.

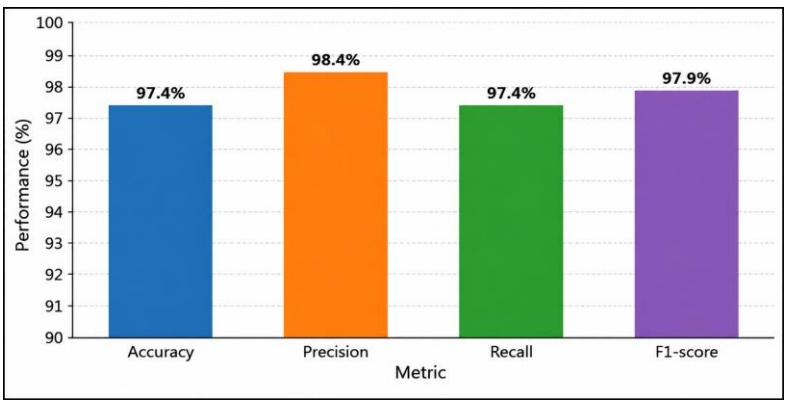


Figure 2: Comparative analysis of model performance metrics

The graphic compares the performance of four evaluation criteria, which are accuracy, precision, recall, and F1-score. Precision at 98.4% achieved the highest score, showing the robust capacity of the model to identify positive predictions correctly. Both Accuracy and Recall score 97.4%, reflecting uniformly good results for the overall classification and successful detection of

relevant cases. The F1-score is 97.9%, which indicates a good compromise between precision and recall. The small fluctuation of the four measures implies the model performs consistently across different evaluation metrics. The chart indicates a very good classification model with high performance, which is above 97% for all the major measures consistently.

4.3. Confusion Matrix Analysis

To further assess the decision quality, a confusion matrix was constructed using the predicted and expected outcomes (Table 2).

Table 2: Confusion matrix

Actual Class	Predicted Approved	Predicted Rejected
Approved	302	8
Rejected	5	185

The confusion matrix shows that only 13 out of 500 customer records were incorrectly classified. The relatively small number of false approvals and false rejections confirms that the logical rules accurately represent the intended business policies.

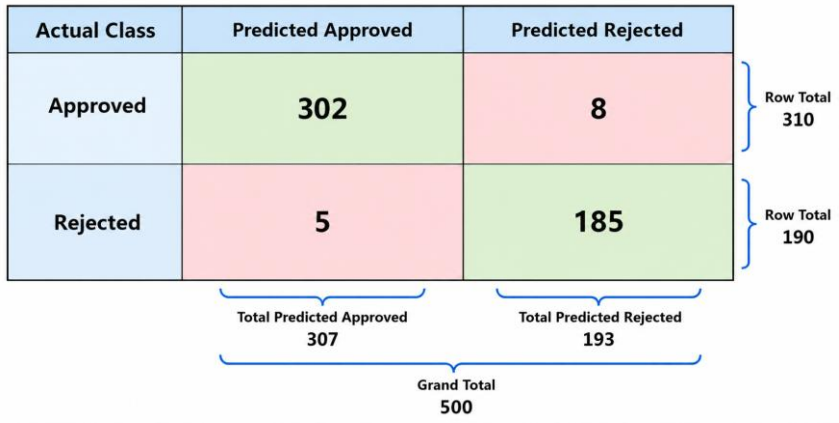


Figure 3: Actual versus predicted classification results

Figure 3 illustrates a confusion matrix of a binary classification model with 500 observations. For the 310 actual accepted cases, 302 were correctly predicted as approved, and 8 were incorrectly predicted as rejected. Of 190 real refused cases, 185 cases were correctly forecasted as rejected, and 5 cases were mistakenly predicted as accepted. The model then produced 307 predicted-approval cases and 193 predicted-rejection instances. In general, the matrix shows a high number of proper classifications and low numbers of false-positive and false-negative predictions. These results suggest that the model has a good classification capacity to discriminate between accepted and rejected cases.

The results of the experiment show a few benefits from using logic programming for the management of business rules. Firstly, the declarative definition of the business policies ensures that the system achieves a high level of decision-making consistency. Since all recommendations are made based on pre-defined logical rules, the same customer conditions will lead to the same outcome in every case, decreasing possible inconsistency that is common in manual decision-making. Secondly, the ability to infer decisions gives the system an important advantage in explainable decision-making. Contrary to numerous machine learning algorithms working as black boxes, the Prolog inference engine shows a set of rules that were applied to derive the recommendation. This way, the manager or auditor can understand why a particular customer application was accepted or rejected.

Thirdly, business rules are separated from application logic, which makes the system highly maintainable. Any change in the organizational policies will result only in a change of the rule base, without any effect on the software code. However, some limitations need to be addressed regarding the presented approach. Firstly, the prototype currently uses manual business rule definitions without adapting to changes in the business environment. Secondly, the evaluation has been carried out using a simulated data set. This fact makes future research needed for validation of the approach using industrial data sets. Moreover, research into approaches combining symbolic reasoning with machine learning is highly desirable. Overall, results of experiments demonstrate the applicability of the proposed P-BRMS for intelligent business decision support. Combining

Prolog-based reasoning with business rules management presents a practical solution as opposed to classical procedural solutions, especially when it comes to transparency, consistency and policy adherence.

5. Conclusion

This paper has developed a Prolog-Based Business Rule Management System (P-BRMS) which facilitates business decision-making in an intelligent manner. This framework makes use of the combination of logic programming and a business rule management system through the representation of organizational policies in terms of Prolog facts and rules, as well as the use of an inference engine in automating the decision-making processes. A prototype of the proposed P-BRMS has been developed using the SWI-Prolog environment and tested using a credit approval problem with 500 customer information records and 15 business rules. The experimental results have shown that the system was able to make correct decisions 97.4% of the time, had a precision rate of 98.4%, a recall of 97.4%, and an F1 score of 97.9%. In addition, the system has been found to have an average inference time of 0.04 seconds per query and has a logical reasoning ability that facilitates the provision of explanations for all decisions made.

This proposed P-BRMS presents certain benefits for businesses that function in fields that require numerous rules, like banking, insurance, healthcare, taxation, and enterprise resource planning, since transparency and policy compliance are crucial in such cases. Utilizing Prolog's declarative programming approach allows business analysts to update the organization's policies without updating the software code, decreasing costs and increasing the organization's agility. Despite the promising performance of this prototype, there are some limitations that need to be addressed. The testing process was performed on the synthesized dataset and manually built business rules; therefore, it cannot fully represent the real-life environment of a business. Future research will be aimed at the validation of the proposed framework based on real enterprise datasets, including databases and business process management platform integration, as well as the development of the hybrid neuro-symbolic decision support system using machine learning algorithms.

Acknowledgement: The authors gratefully acknowledge the support and resources provided by the University of Information Technology, Budapest Metropolitan University, Canadian International College, Ramkhamhaeng University, and the University of Tetova.

Data Availability Statement: The data for this study are available upon request from the corresponding author.

Funding Statement: This manuscript and research paper were prepared without any financial support or funding.

Conflicts of Interest Statement: The authors have no conflicts of interest to declare. This work represents a new contribution by the authors, and all citations and references are appropriately included based on the information utilized.

Ethics and Consent Statement: This research adheres to ethical guidelines and obtains informed consent from all participants. Confidentiality measures were implemented to safeguard participant privacy.

References

1. R. G. Ross, “Principles of the Business Rule Approach,” *Addison-Wesley, Boston*, Massachusetts, United States of America, 2003.
2. B. Von Halle and L. Goldberg, “The Decision Model: A Business Logic Framework Linking Business and Technology,” *CRC Press*, Boca Raton, Florida, United States of America, 2009.
3. A. Macioł, “An application of rule-based tool in attributive logic for business rules modeling,” *Expert Systems with Applications*, vol. 34, no. 3, pp. 1825–1836, 2008.
4. L. Sterling and E. Shapiro, “The Art of Prolog: Advanced Programming Techniques,” *MIT Press*, Cambridge, Massachusetts, United States of America, 1994.
5. I. Bratko, “Prolog Programming for Artificial Intelligence,” *Addison-Wesley*, Boston, Massachusetts, United States of America, 2001.
6. J. Wielemaker, T. Schrijvers, M. Triska, and T. Lager, “SWI-Prolog,” *Theory and Practice of Logic Programming*, vol. 12, no. 1–2, pp. 67–96, 2012.
7. E. Turban, J. E. Aronson, and T. P. Liang, “Decision Support Systems and Intelligent Systems,” *7th ed. Prentice Hall*, Upper Saddle River, New Jersey, United State of America, 2005.
8. D. J. Power and C. Heavin, “Decision Support, Analytics, and Business Intelligence,” *3rd ed. Business Expert Press*, New York, United States of America, 2017.

9. D. Gunning and D. W. Aha, "DARPA's explainable artificial intelligence (XAI) program," *AI Magazine*, vol. 40, no. 2, pp. 44–58, 2019.
10. S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach," *4th ed. Pearson*, Hoboken, New Jersey, United States of America, 2021.
11. D. Rosca and C. Wild, "Towards a flexible deployment of business rules," *Expert Systems with Applications*, vol. 23, no. 4, pp. 385–394, 2002.
12. M. Gelfond and N. Leone, "Logic programming and knowledge representation—The A-Prolog perspective," *Artificial Intelligence*, vol. 138, no. 1–2, pp. 3–38, 2002.
13. R. T. H. Chi and M. Y. Kiang, "Reasoning by coordination: an integration of case-based and rule-based reasoning systems," *Decision Support Systems*, vol. 6, no. 2, pp. 103–113, 1993.
14. P. Harmon and C. Wolf, "Business Process Change: A Business Process Management Guide for Managers and Process Professionals," *4th ed. Morgan Kaufmann*, Cambridge, Manchester, United States of America, 2019.
15. J. G. Gonzales, A. C. Lagman, R. M. D. Angel, M. C. F. Raguro, R. C. A. Bulacan, and J. D. Canlas, "Graduate tracer monitoring platform with decision support feature and mapping recommendations analysis using rule-based algorithm," *in Proc. 2024 IEEE 15th Control and System Graduate Research Colloquium (ICSGRC)*, Shah Alam, Malaysia, 2024.
16. G. Nápoles, F. Hoitsma, A. Knoben, A. Jastrzebska, and M. L. Espinosa, "Prolog-based agnostic explanation module for structured pattern classification," *Information Sciences*, vol. 622, no. 4, pp. 1196–1227, 2023.
17. W. Wang, T. Chen, M. Indulska, S. Sadiq, and B. Weber, "Business process and rule integration approaches—An empirical analysis of model understanding," *Information Systems*, vol. 104, no. 2, pp. 37-52, 2022.
18. D. Bork, S. J. Ali, and G. M. Dinev, "AI-enhanced hybrid decision management," *Business & Information Systems Engineering*, vol. 65, no. 2, pp. 179–199, 2023.

Publisher's Note: *The publisher remains impartial concerning jurisdictional claims in published maps and institutional affiliations. Responsibility for the content rests entirely with the authors and does not necessarily reflect the publisher's perspectives.*