IBA-VNS: A Logic-Based Machine Learning Algorithm and Its Application in Surgery

Nevena Ćolić 1, Pavle Milošević 1,*, Ivana Dragović 1 and Miljan S. Ćeranić 2

1 Faculty of Organizational Sciences, University of Belgrade, Jove Ilića 154, 11000 Belgrade, Serbia; nv20165029@student.fon.bg.ac.rs (N. Ć.); ivana.dragovic@fon.bg.ac.rs (I.D.)
2 Clinic for Emergency Surgery, University Clinical Center of Serbia, Faculty of Medicine, University of Belgrade, 11000 Belgrade, Serbia; miljan.ćeranic@med.bg.ac.rs
* Correspondence: pavle.milosevic@fon.bg.ac.rs; Tel.: +381-11-3950-852

Abstract: The interpretability and explainability of machine learning (ML) approaches play a key role in the trustworthiness of ML models in various applications. The objective of this paper is to incorporate a logic-based reasoning in the ML model that is not only accurate but also interpretable and easily applied. More precisely, we propose a hybrid IBA-VNS approach based on interpolative Boolean algebra (IBA) and variable neighborhood search (VNS). IBA is chosen over traditional multi-valued and/or fuzzy logic techniques due to its consistency in preserving all Boolean axioms. The VNS heuristic is used for model training, i.e., determining the optimal logical aggregation function within the IBA framework for solving observed prediction problems. Obtained logic aggregation functions are easy to understand and may provide additional insight to the decision-maker. The proposed approach does not require any domain knowledge and is applicable in various domains. IBA-VNS is evaluated on several standard datasets. Further, IBA-VNS is applied to the real-world problem of predicting hospital length of stay (LOS), showing exceptional results in terms of interpretability and accuracy. In fact, the dataset is collected from the LabSerb program regarding colorectal surgeries in the period 2015–2023. The proposed approach extracted knowledge regarding the problem, i.e., the causal relations between the patient’s health condition and LOS, along with achieving an MAE of 1.144 days.

Keywords: interpolative Boolean algebra; variable neighborhood search; interpretable machine learning; logic-based explainability; hospital length of stay prediction

MSC: 68T27

1. Introduction

The potential of machine learning (ML) algorithms is vast due to their ability to perform complex input–output mapping without an exact mathematical model. This feature has led to their widespread application across diverse domains, including computer graphics, visualization, speech recognition, data mining, information retrieval, and beyond. However, as ML models increase in complexity, the human ability to understand and interpret decisions has not been able to keep up. Due to their opaque reasoning mechanisms, most machine learning models are considered to be black boxes [1]. Recognizing this challenge, ML users increasingly stress the significance of the model explainability and interpretability. Particularly, transparency, interpretability, and explainability are paramount for fostering trust and ensuring the ethical and responsible use of ML [2]. Consequently, it becomes imperative to strike a suitable balance between interpretability and model performance when designing an ML algorithm [3]. Ensuring that a model is not only accurate but also interpretable fosters user trust, facilitating better understanding and acceptance of machine learning solutions in diverse applications.
Academic researchers also recognize the importance of more transparent and explainable systems for better understanding and analysis of machine learning models. In pursuit of this goal, some papers have explored the integration of machine learning with multi-valued fuzzy logic as an explainable yet precise soft computing technique for data presentation [4]. Zou et al. [5] focused on making the machine learning process explainable by providing real-time internal status updates. This involved selecting and revealing meaningful internal states of machine learning algorithms during the model inference. The paper employed various visualization techniques to allow users to interactively observe how the final results are obtained in machine learning. Additionally, Wood [6] enhanced the transparency of machine learning algorithms by clearly disclosing intermediate calculations and the weightings applied to independent variables in deriving predictions. The substantial progress in enhancing the explainability of the ML model was achieved by incorporating logical relations into the machine learning algorithm [7]. Integrating logical relations into machine learning involves incorporating formal logical structures into the learning process. Unlike traditional ML methods that rely solely on statistical patterns, logic-based approaches leverage explicit rules and logical reasoning to enhance the interpretability and explainability of the models [8]. By introducing logical connections, these models can capture complex relationships, handle uncertainty, and provide insights into the decision-making process.

One possible direction for enhancing the interpretability of ML involves the incorporation of interpolative Boolean algebra (IBA) in ML settings. IBA is a consistent [0, 1]-realization of Boolean algebra in the sense of preserving all the laws on which Boolean algebra relies [2]. In contrast to many multi-valued logics, including traditional fuzzy logic, IBA relies on the principle of structural functionality, making it applicable within the Boolean framework [9]. IBA enables transparency in the inference process, making it an effective choice for a wide range of domains, e.g., from ranking to prediction problems. Particularly, IBA-DE and IBA-GA approaches paved the way for application of IBA in ML settings [10,11]. Still, there is substantial space for different IBA-based ML approaches and models. For instance, genetic algorithms (GAs) and differential evolution (DE) can sometimes struggle with premature convergence problem due to improper parameter settings. This may lead to premature convergence problem as well as a time-consuming optimization process. Therefore, the hybridization of IBA with a simple, non-parametric and yet powerful optimization heuristic may be seen as a promising direction to explore.

In this paper, we aim to formalize, inspect, and extend the idea presented in [12]. In fact, we employ the variable neighborhood search (VNS) heuristic in order to determine the best logical aggregation function within the IBA framework for dealing with an observed prediction problem. The proposed IBA-VNS ML approach utilizes the full potential of IBA-based logical aggregation functions to understand and analyze a model and further prediction, while VNS is used in the simple form, in order to avoid parameter settings and minimize algorithm complexity. The IBA-VNS approach consists of three phases: data preparation, execution of the IBA-VNS algorithm, and interpretation of the results. This approach offers the advantage of providing a verbal model during the training phase, followed by enhancing the interpretability of the model by revealing both the significance of the factors and the logical relations. Therefore, we can assert that the primary contribution of the IBA-VNS approach lies in meaningful insight in the inference process, as well as interpretable results. Finally, verbal models and interpretable recommendations are paramount in real-world settings to enhance confidence in decision-making. The approach is validated on several well-known test datasets, showing consistent performance.

Finally, IBA-VNS is applied for solving the hospital length of stay prediction after laparoscopic surgeries. In the medical domain, a transparent logic-based ML holds a particularly significant place due to the critical nature of healthcare decisions. The experiment was conducted on the real-world dataset, collected on the territory of the Republic of Serbia in the period from January 2015 to December 2023.
The rest of the paper is organized as follows. The theoretical background of the two main methodologies used in the proposed hybrid model are given in Section 2. In Section 3, we present the proposed IBA-VNS approach and the main steps of the algorithm. Section 4 contains validation of the proposed model, experimental setup, performance measures, and results in comparison to other machine learning algorithms. Implementation of the IBA-VNS approach in the medical domain, more precisely on the hospital length of stay, is presented in Section 5. Section 6 summarizes the conclusion and some future work.

2. Theoretical Background

In this section, we will briefly explain the theoretical background of the proposed IBA-VNS approach, encompassing machine learning, metaheuristic and interpolative Boolean algebra. Machine learning is one of the most important subsets of artificial intelligence in terms of practical application. A commonly used definition of machine learning was given by Mitchell [13]. It follows that ML independently solves certain tasks based on data and improves results automatically, relying on experience. ML algorithms often involve metaheuristics within the algorithm learning phase, while fuzzy logic-based techniques may be used to enhance algorithm interpretability.

2.1. Metaheuristics in Machine Learning

Metaheuristic optimization techniques are a class of optimization algorithms used in situations where traditional approaches may struggle. For this reason, such techniques remain unaffected by the drawbacks of traditional mathematical programming search techniques [14]. Genetic algorithms, particle swarm optimization (PSO), tabu search (TB), and VNS are some of the more popular algorithms applied for parameter estimation purposes. The metaheuristics are not designed specifically for a particular problem but are considered as a general approach that can be tuned for specific issues [15].

Over the years, numerous papers that have successfully applied metaheuristics to optimize a wide variety of machine learning algorithms have been published. For instance, Nandi et al. [16] proposed a promising path for integration of metaheuristics and ML algorithms. They used a combination of process modeling and optimization strategies, integrating artificial neural networks, support vector regression, and GAs. Further, researchers have effectively employed GAs to optimize the parameters of neural networks, significantly enhancing their performance and adaptability [17]. Also, PSO [18] and TB [19] demonstrated success in optimizing feature selection for support vector machines, leading to notable improvements in model efficiency. Many additional successful applications of metaheuristics in optimizing various machine learning algorithms are detailed and discussed in [20]. The referenced studies illustrate how the hybridization of machine learning and metaheuristics accelerates the overall model and enhances the development of more efficient and promising solutions.

2.2. Variable Neighborhood Search

Variable neighborhood search, introduced by Mladenović and Hansen [21], represents a flexible framework for building heuristics that solve discrete and continuous nonlinear optimization problems. This simple metaheuristic algorithm systematically changes the size and type of the neighborhood during the search process in order to escape from local optima [22]. This helps the VNS to explore neighborhoods that are distant from the current solution and to move towards an improved solution. VNS relies upon the following observation [23]:

- A local minimum with respect to one neighborhood structure is not necessarily a local minimum for another neighborhood structure;
- A global minimum is a local minimum with respect to all possible neighborhood structures;
- For many problems, local minima with respect to one or many neighborhoods are relatively close to each other.
These observations can be treated deterministically, stochastically, or combined. Depending on the way these observations are used, different modifications of the VNS method are obtained: VN descent [24], reduced VNS [25], general VNS [26], skewed VNS [25], and VN decomposition search [27]. In this paper, basic VNS, as a combination of deterministic and stochastic method, is used for solving combinatorial optimization problems.

The basic VNS heuristic consists of three stages: the shaking procedure, improvement procedure, and neighborhood change step [28]. Let \( N_k (k = 1, \ldots, k_{\text{max}}) \) be a set of predefined neighborhood structures, where \( k_{\text{max}} \) is the total number of neighborhood structures to be used in the search. A simple shaking procedure is based on selecting a random solution \( x' \) from the \( k^{th} \) neighborhood structure in order to escape from local optima (diversification), i.e., to resolve local minima traps. Further, in the improvement phase, the local search heuristic method is applied repeatedly in order to seek an improvement in the neighborhood of the current solution based on predefined neighborhood structures (intensification), until a local optimum is found. The purpose of a third stage, the neighborhood change step is to guide the variable neighborhood search heuristic while exploring the solution space. In this step, a decision is made about the next neighborhood to be searched as well as whether some solution will be accepted as a new incumbent solution or not [29]. The widely used neighborhood change procedures are as follows: the sequential neighborhood change step [30], cyclic neighborhood change step [31], pipe neighborhood change step [32], and skewed neighborhood change step [33]. The improvement phase and the shaking procedure, together with the neighborhood change step, are executed alternately until a predefined stopping condition is fulfilled. The stopping condition can be a limit on CPU time, a limit on the number of iterations, or a certain number of iterations without improvement. Algorithm 1 describes the main stages of the basic version of the VNS algorithm with only one parameter [25].

Algorithm 1 Steps of the basic VNS [25]

1. **Initialization**: Select the set of neighborhood structures \( N_k \), for \( k = 1, \ldots, k_{\text{max}} \), that will be used in the search; find an initial solution \( x \); choose a stopping condition;
2. **Repeat** the following sequence until the stopping condition is met:
   1. Set \( k \leftarrow 1 \);
   2. **Repeat** the following steps until \( k = k_{\text{max}} \):
      a. **Shaking**: Generate a point \( x' \) at random from the \( k^{th} \) neighborhood of \( x \) \((x' \in N_k(x))\);  
      b. **Local search**: Apply some local search method with \( x' \) as an initial solution; denote with \( x'' \) the obtained local optimum;
      c. **Move or not**: If the local optimum \( x'' \) is better than the incumbent \( x \), move there \((x \leftarrow x'')\), and continue the search with \( N_1 \) \((k \leftarrow 1)\); otherwise, set \( k \leftarrow k + 1 \).

Unlike many other metaheuristics, the basic schemes of VNS and its extensions are simple and require few, and sometimes no, parameters. Consequently, in addition to providing very good solutions, often in simpler ways than other methods, VNS gives insight into the reasons for such a performance, which can lead to more efficient and sophisticated implementations [25]. These advantages have been validated in various studies, where the VNS method has consistently demonstrated effectiveness in optimizing ML models across diverse applications. Han and Wang [34] introduce a hybrid model, combining the random forest method with VNS for enhanced performance. Araujo et al. [35] demonstrate the efficacy of VNS in improving convolutional neural networks. In [36], the first SVM-VNS algorithm for classification is presented and successfully applied in predicting responses of osteosarcoma patients. Alba et al. [37] optimize parameters for a hardware-adapted SVM version. Sidaoui and Sadouni [38] propose a VNS for multiclass clustering using the carrier vector method, enhancing multiclass classifier performance for high-dimensional problems. Vahdani et al. [39] and Yazdani et al. [40] apply VNS for optimizing SVM in nonlinear regression problems, utilizing the general method of variable environments. Also, a novel
2.3. Interpolative Boolean Algebra

Interpolative Boolean algebra is a consistent real-valued [0, 1] realization of classical Boolean algebra proposed by Radojević [42]. Unlike any other multi-valued logic, IBA is considered to be consistent in the sense that all Boolean axioms and theorems are preserved in the [0, 1]-valued case. IBA is based on the structure functionality principle, which states that the structure of any IBA element can be directly calculated on the basis of the structure of its components. The structure functionality principle implies the necessity to treat the structure of the logical function independently of their valued realization [42]. Therefore, IBA has two levels: symbolic (or qualitative) and valued (or quantitative).

The main idea behind the IBA framework is to present a logical expression \( \varphi(a_1, \ldots, a_n) \) of attributes \( a_1, \ldots, a_n \) as a scalar product of structural and atomic vectors on the symbolic level, while the values are introduced on the valued level. An atomic vector consists of atomic elements \( a_i(a_1, \ldots, a_n) \), the simplest element of Boolean algebra in the sense that it does not contain any other Boolean algebra element, except for the trivial constant 0 [42]. The number of atomic elements is \( 2^n \), i.e., it is determined by the number of attributes \( n \). The structure of a logical expression determines which atomic elements are included and which are not included in it [9]. In other words, any logical expression on the IBA symbolic level is determined by a structure vector \( \sigma_\varphi \) — a binary vector as long as the number of atoms.

\[
\varphi(a_1, \ldots, a_n) = \sigma_\varphi \left[ \begin{array}{c} \alpha_1(a_1, \ldots, a_n) \\ \vdots \\ \alpha_2^n(a_1, \ldots, a_n) \end{array} \right]
\]

In other words, the atomic vector represents the intensities of variables and their relations, while the structural vector defines the inclusion of atoms in the final expression.

Since the structural vector is binary, operations of conjunctions, disjunction, and negation in the IBA framework is a direct extension of operations in the classical binary logic. In this way, all the laws of Boolean algebra are preserved at the structural level of the IBA. From the technical standpoint, operations of conjunctions, disjunction, and negation in the IBA framework may be modeled by a generalized Boolean polynomial (GBP) using the following rules [2]:

\[
\begin{align*}
\varphi(a_1, \ldots, a_n) \land \Omega(a_1, \ldots, a_n) & = \varphi \otimes(a_1, \ldots, a_n) \otimes \Omega \otimes(a_1, \ldots, a_n), \\
\varphi(a_1, \ldots, a_n) \lor \Omega(a_1, \ldots, a_n) & = \varphi \oplus(a_1, \ldots, a_n) + \Omega \oplus(a_1, \ldots, a_n) - \varphi \otimes(a_1, \ldots, a_n) \otimes \Omega \otimes(a_1, \ldots, a_n), \\
\neg \varphi(a_1, \ldots, a_n) & = \big(\neg \varphi(a_1, \ldots, a_n)\big) \otimes = 1 - \varphi \otimes(a_1, \ldots, a_n)
\end{align*}
\]

After the transformation process to a GBP has been completed, the value of each element is calculated at the value level. The value of any IBA element represents the intensity of the specific attribute and has [0, 1]-value realization. The operations that can be performed within the Boolean polynomial are as follows: generalized product (\( \otimes \)), standard plus (+), and minus (−). The generalized product has priority in expressions over other operations. At the value level, each element of the IBA can be uniquely represented by the corresponding GBP. In order to calculate the value of the IBA element, it is necessary to choose an appropriate realization of the generalized product operator. The standard \( t \)-norms that are the most usual choice [43] are the minimum function, ordinary product, and Lukasiewicz operator. The decision of which \( t \)-norm to use depends on the nature of the attributes and/or correlation between them. Attributes of the same or similar nature should be aggregated using the minimum, the Lukasiewicz \( t \)-norm should be used for negatively correlated attributes, and the ordinary product for attributes that are not correlated, i.e., different by the nature [44,45].
IBA has a wide range of applications from data clustering \cite{45,46} to ranking and decision-making \cite{47,48}. The most prominent application of IBA involves data aggregation for tasks of ranking, selection, or prediction. Logical aggregation (LA) is an aggregation procedure in the IBA frame that involves combining multiple logical expressions to derive a single result that captures the combined uncertainty or vagueness present in the original expressions \cite{9}. The simplest form of LA function involves a single logical function, and as a consequence, a single structural vector, used for data aggregation. There is a growing interest for the application of LA in the context of machine learning, i.e., to extract optimal LA function based on the data.

2.4. Interpolative Boolean Algebra in Machine Learning

Recently, IBA has been successfully applied within ML algorithms for classification and prediction problems \cite{43,49}. However, these approaches are mainly based on IBA similarity, and they do not use the full potential of IBA LA functions. LA functions have multiple advantages such as transparency and interpretability, but they demand a high level of expert knowledge for proper implementation. So, the natural idea of optimizing LA functions by finding the optimal structural vector for the observed problem using metaheuristic optimization methods came into focus.

In \cite{11}, an author presented the idea of optimizing the structural vector of the LA function using GAs. The model was used to choose a strategy for trading on the stock market. Due to the difference in the nature of the information, the product was used as the generalized product operator. As a result of the research, a hybrid IBA-GA classification model was obtained that makes a decision on buying or selling a stock or currency.

A second model that integrates IBA into ML settings is presented in \cite{10}. The authors proposed a hybrid approach for sovereign credit rating prediction based on IBA and differential evolution (DE) algorithm. The main idea of this paper was to use ML and computational intelligence to increase the transparency of credit ratings modeling as well as to improve forecasting performance. The proposed model used IBA for processing macroeconomic inputs and translating them into atomic vectors which were further used as inputs for the optimization with the DE algorithm. The final IBA model was used to determine the impact of each input and improve the model by excluding those factors with the smallest impact on the final output.

The IBA-GA and IBA-DE models demonstrate several limitations. Firstly, both approaches require a substantial number of function evaluations during optimization. Secondly, they demand considerable optimization time, often necessitating extended periods to converge to optimal solutions. Furthermore, these models are susceptible to the curse of dimensionality, particularly when handling datasets with numerous attributes. These shortcomings underscore the importance of exploring alternative optimization models and conducting further research to address these challenges effectively.

3. IBA-VNS Approach

The proposed IBA-VNS approach is presented in detail in this section.

3.1. The Basics of IBA-VNS Approach

The motivation behind this novel approach is the necessity for developing a logic-based ML algorithm that is both interpretable and applicable across a wide range of domains. On one hand, the interpretability of the model is enabled by implementing IBA, renowned for its transparency and ease of understanding. We have specifically chosen IBA over fuzzy logic due to its consistency in preserving all Boolean axioms and laws, including the laws of excluded middle and contradiction. On the other hand, unlike many other optimization algorithms, basic VNS stands out as a simple non-parametric heuristic that enables fast convergence to the best available solution. It worked particularly well for solving the combinatorial optimization problem \cite{50}. This characteristic renders VNS an excellent fit for our approach.
As it is explained in Section 2.3, any logical expression in the IBA framework is an LA function that can be presented as a scalar product of the atomic and structural vector. Values of atomic vector elements are constant for each instance. Bearing in mind that the structural vector determines logical expression and that it is binary by its nature, finding the adequate structure of expression comes down to the combinatorial (binary) optimization problem. Therefore, basic VNS seems to be the natural choice as an optimization heuristic. In other words, the VNS algorithm’s objective is to effectively identify the optimal structure vector for a given problem in the proposed approach. Compared to the other proposed IBA-based methods in ML settings, e.g., IBA-GA and IBA-DE, IBA-VNS is computationally less expensive and time-consuming. More precisely, unlike the IBA-VNS approach, the basis of IBA-DE and IBA-GA are population-based optimization heuristics, i.e., the optimization process implies a significant number of function evaluations (FEs) within a single generation. This may lead to a time-consuming optimization process, since the final number of FEs in GAs and DE is equal to the product of the population size and the number of generations utilized in optimization. In fact, the experimental results show that the VNS-based approach is efficient, reducing significantly the computation time of several well-known problems from the literature [51].

Also, genetic algorithms and differential evolution can sometimes struggle with premature convergence problem, i.e., due to the GA and DE parameter settings, the algorithms may settle on suboptimal solutions. In contrast, VNS incorporates a dynamic neighborhood search strategy that enables systematic exploration of the diverse solution space. Finally, the IBA-VNS approach is a straightforward method that does not require any control parameters, unlike IBA-GA and IBA-DE.

The final outcome of the IBA-VNS approach is an optimal structure of the LA function for the observed problem. The LA function facilitates clear-cut analysis and interpretation of the output, providing transparency and additional knowledge extracted from data. Compared to well-known ML add-ons that provide some explainability to a black-box model, such as SHAP [52], IBA-VNS provides interpretability on a higher level. In fact, the IBA-VNS approach enables a deeper insight into the model and decision-making process since the obtained LA function can easily be translated into a verbal model. Thus, the logical connections between the input attributes are extracted from the data and clearly given to the end-user, along with the importance of individual atoms in the final results. On the other hand, SHAP provides only the influence of individual inputs in the final decision.

3.2. Phases of IBA-VNS Approach

The IBA-VNS approach is organized into three phases: the data preparation, the execution of the IBA-VNS algorithm, and the interpretation of results, as shown in Figure 1.

![Figure 1. The main phases of the IBA-VNS approach.](image)

**Data preparation** is an initial phase in the proposed IBA-VNS approach. This phase involves several key steps aimed at ensuring the quality and suitability of the input data for subsequent analysis and model development. The first step in the data preparation is thorough analysis and understanding of the input attributes. This involves examining the distribution of data points, identifying potential outliers, and understanding their relevance.
Especially, the presence of the outliers can significantly affect the performance of machine learning models and must be addressed appropriately, e.g., they can be omitted or taken into account in the data normalization step. The second step involves data normalization, i.e., ensuring that all input values fall within the range [0, 1], as mandated by the modeling process within the IBA framework. To achieve this, we have employed the standard min–max normalization technique, which scales the values accordingly. However, in scenarios where outliers are present, alternative normalization methods, such as utilizing piecewise linear normalization with fixed interval endpoints or employing nonlinear function, e.g., sigmoid, could be appropriate. Furthermore, correlation analysis is performed to identify relationships between different input attributes. Correlation analysis is a critical step for effective feature selection and/or dimensionality reduction, as well as for choosing an appropriate GP operator in the IBA framework. Highly correlated attributes may introduce redundancy into the dataset and adversely impact model performance. In such cases, it is recommended to reduce the dimensionality of the dataset by selecting a subset of relevant features. This can be accomplished by choosing a single attribute from a group of highly correlated attributes or by performing attribute aggregating using suitable logical functions, e.g., minimum. This phase is summarized in Figure 2.

![Figure 2. Data preparation phase of the IBA-VNS approach.](image)

Once the data preparation phase is complete, the **training of the IBA-VNS model** can be conducted. First, the algorithms should be set for the training process. The initial step in this process involves the automatic generation of atoms based on the number of input attributes. As explained in Section 2.3., the number of atoms is equal to $2^n$, with $n$ representing the number of attributes in the dataset. Furthermore, it is necessary to choose the GP operator among input attributes according to the correlation analysis. If
the issue of highly correlated attributes is resolved, the product can be selected as the operator. Otherwise, recommendations given in Section 2.3 should be followed. Since one criterion optimization is employed, selecting a single metric, i.e., one objective function, is necessary. Depending on the nature of the problem, objective functions for regression may involve minimizing mean squared error (MSE) or mean absolute error (MAE), while for classification, they may entail maximizing accuracy, precision, or F1 score. Finally, defining the stopping condition, such as setting the maximum number of iterations and implementing early stopping, is important for effective algorithm execution. In case of IBA-VNS approach, the early stopping criterion is in conflict with the VNS paradigm, so it is omitted. In this version of the approach, reaching the time limitation and the number of iterations is solely implemented as the stopping criterion, bearing in mind that the number of iterations and function evaluations are equal in this approach. The desired value of the objective function can be also utilized as the algorithm stopping criterion. The setup of the IBA-VNS algorithm is represented in Figure 3.

![Figure 3. The setup of the IBA-VNS algorithm.](image)

Further, VNS-based optimization of the LA structure vector in the IBA frame is conducted. The initial solution, i.e., the initial structural vector is selected randomly. The LA value is computed, followed by the derivation of the objective function. After this, the shake phase is initiated within the VNS method, wherein the structural vector for one or more atoms is altered. The selection of which atom to modify is random. Subsequently, the LA of the newly formed structural vector is calculated, followed by the computation of the objective function. If the solution is better than the current best solution, it is promoted as the new best solution. Otherwise, the process returns to the shake phase to alter additional atom of the structural vector. The optimization procedure is summarized in Figure 4.

![Figure 4. The optimization procedure of the IBA-VNS algorithm.](image)
Further, VNS-based optimization of the LA structure vector in the IBA frame is conducted. The initial solution, i.e., the initial structural vector is selected randomly. The LA value is computed, followed by the derivation of the objective function. After this, the shake phase is initiated within the VNS method, wherein the structural vector for one or more atoms is altered. The selection of which atom to modify is random. Subsequently, the LA of the newly formed structural vector is calculated, followed by the computation of the objective function. If the solution is better than the current best solution, it is promoted as the new best solution. Otherwise, the process returns to the shake phase to alter additional atom of the structural vector. The optimization procedure is summarized in Figure 4.

Figure 4. The optimization of a structure vector using the IBA-VNS algorithm.

The third and final phase involves the interpretation of the model and the obtained results. The IBA-VNS algorithm results in an optimized structure vector for an observed training dataset. The structure vector can be easily mapped into a logical expression, i.e., an explainable model for the prediction of target values. Consequently, the model provides a clear understanding and meaningful insights into the outcomes. Logical expressions are easy to understand for both experts in the field, as well as decision-makers without deep knowledge in model inference and training. Further, the model is directly assessed based on the input attributes of instances in the test part of the dataset.

A pseudo-code of the IBA-VNS approach is presented in detail in Algorithm 2.
Algorithm 2 Essential steps of IBA-VNS approach

Data preparation:
1. Perform data normalization;
2. Define correlation threshold, e.g., \( a = 0.6 \); Determine GP operator based on correlation analysis:
   - (2.1) If \( \rho \geq a \) than GP is standard product;
   - (2.2) If \( -a < \rho < a \) than GP is minimum;
   - (2.3) If \( \rho \leq -a \) than GP is Lukasiewicz operator;
3. Perform chosen data reduction based on attribute selection or attribute aggregation method if necessary;

IBA-VNS:
4. Create an atomic vector \( \alpha \); Calculate values of atomic vectors \( \alpha^v \) for the dataset;
5. Select the set of neighborhood structures \( N_k \), where \( k = 1, \ldots, 2^n \);
6. Randomly initialize structural vector \( \sigma \);
7. Calculate initial solution \( x \), as a product of structural vector \( \sigma \) and values of atomic vectors \( \alpha^v \);
8. Repeat the following until the stopping condition is met:
   - (8.1) Set \( k \leftarrow 1 \);
   - (8.2) Repeat the following steps until \( k = k_{\text{max}} \):
     - (8.2.a) Shaking: Generate a structural vector \( \sigma' \) at random from \( k^{\text{th}} \) neighborhood of \( \sigma \) \( (\sigma' \in N_k(\sigma)) \);
     - (8.2.b) Expression calculation: Calculate solution \( x' \), as a product of structural vector \( \sigma' \) and values of atomic vectors \( \alpha^v \);
     - (8.2.c) Move or not: If the solution \( x' \) is better than the solution \( x \), or if some acceptance criterion is met, move there \( (x \leftarrow x') \) and \( (\sigma \leftarrow \sigma') \), and continue the search with \( N_1 (k \leftarrow 1) \); Otherwise, set \( k \leftarrow k + 1 \);
9. Interpretation of the outcome:
   - Calculate the final LA \( \phi \) as a product of the atomic vector \( \alpha \) and the best structural vector \( \sigma \);
   - Calculate the minimal form of the LA \( \phi \) in order to ease the interpretation.

3.3. The Main Advantages and Limitations of IBA-VNS Approach

While many ML algorithms function as black boxes, the IBA-VNS algorithm provides transparent outputs, enhancing users’ confidence in the model’s decisions. In other words, the IBA-VNS approach offers distinct advantages over many traditional machine learning methods, particularly in terms of interpretability and transparency. The final structural vector can be easily mapped into a LA function, i.e., logical expression, that is easy to understand and verbalize. On one hand, the obtained LA function may verify the decision-maker’s point of view on a certain problem, while on the other hand, it may provide additional insight to the decision-maker, as well as offer a different perspective on the observed problem. Therefore, this approach performs a knowledge extraction and summarization valuable to the decision-maker. Finally, IBA-VNS is a general approach that is not restricted by the application domain.

When compared to the mentioned IBA machine learning methods, IBA-GA and IBA-DE, the IBA-VNS algorithm demonstrates several advantages. As it was previously pointed out, in the IBA-VNS algorithm, the number of iterations is equal to the number of function evaluations, as opposed to IBA-GA and IBA-DE. Thus, the same number of iterations in IBA-VNS is significantly computationally less expensive and time-consuming. Moreover, unlike IBA-GA and IBA-DE, which require setting control parameters (e.g., crossover rate, mutation rate, and population size) manually, the proposed IBA-VNS algorithm operates in a non-parametric manner, streamlining the optimization process. The IBA-VNS algorithm...
employs a local search strategy by altering the structure vector atom by atom. This fine-grained search approach facilitates a targeted exploration of the solution space, resulting in the discovery of high-quality solutions within a shorter timeframe.

Compared with existing ML approaches, in the case of IBA-VNS, we sacrifice some of the predictive performance for the sake of the white-box approach. In other words, IBA-VNS is limited in detecting linear combinations of logic-based patterns in the data, unlike neural networks or IBA-DE.

The primary challenge encountered by the IBA-VNS approach is the curse of dimensionality. In other words, the atomic vector size exhibits exponential growth as the number of inputs increases, as stated in Section 2. This presents a significant computational challenge, resulting in increased computation time and resource requirements. Therefore, the second phase of IBA-VNS is applicable only in cases when the number of input attributes is limited. However, dimensionality reduction techniques such as feature selection (e.g., filter-based methods) or data aggregation can be employed to transform a larger dataset (in terms of number of inputs) in the form that is suitable for IBA-VNS, without affecting the model interpretability.

4. Validation of the IBA-VNS Approach

In this section, the proposed IBA-VNS approach is evaluated using four benchmark datasets compared to other well-known ML algorithms. The chosen datasets are heavily explored, and their usage is a common standard practice when introducing a novel algorithm. Furthermore, it can be affirmed that all attributes in the benchmark datasets contribute significantly to the prediction of the output, as their individual effects on the outcome have been well-established through comprehensive testing and analysis.

In order to validate the proposed IBA-VNS model across a wide array of applications, four datasets, encompassing diverse domains and featuring various attribute types, have been carefully selected. The datasets are as follows: Combined Cycle Power Plant [53], Yacht Hydrodynamics [54], Real estate valuation [55], Insurance [56]. Table 1 provides basic information about these datasets.

Table 1. The main information about datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Attributes</th>
<th>Number of Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combined Cycle Power Plant</td>
<td>4</td>
<td>9568</td>
</tr>
<tr>
<td>Yacht Hydrodynamics</td>
<td>6</td>
<td>308</td>
</tr>
<tr>
<td>Real estate valuation</td>
<td>5</td>
<td>414</td>
</tr>
<tr>
<td>Insurance</td>
<td>6</td>
<td>347</td>
</tr>
</tbody>
</table>

Based on a review of ML applications [57], the most commonly utilized algorithms with highest performance in this range are neural networks (NNs), support vector machine (SVM), and k-nearest neighbors (kNN) algorithm. These models are well-established and widely employed in various domains, and benchmarking against these standard methods provides a meaningful baseline for evaluating the efficacy of the proposed model. Neural networks are known as non-linear universal approximators and one of the most important black box ML models. On the other hand, kNN is a much simpler lazy learning method that stands out by its transparency. The SVM algorithm offers a compromise in terms of accuracy/transparency, i.e., it is more complex than kNN, yet quite more interpretable than NN. Comparing the IBA-VNS model’s performance against these different complexity levels helps in understanding the trade-off between model complexity and predictive accuracy. Further, two well-known interpretable ML models, generalized additive models (GAM) and decision trees (DT), are included in comparison. GAMs are considered interpretable as they facilitate the visualization and comprehension of the relationship between predictions and attribute variables. Similarly, the simplicity of DTs and their ability to provide transparent, rule-based explanations for predictions make them highly interpretable.
The proposed IBA-VNS method and other ML algorithms are implemented in the MATLAB 2021b environment. To ensure a fair comparison among the models, all datasets were divided as follows: 80% of the data were used for model training, while the remaining 20% represented the test data. In order to eliminate the effect of features that have different scales, a min–max normalization with fixed endpoints based on the train data is applied.

During the training process, parameter optimization was conducted individually for each dataset across all machine learning algorithms utilized for comparison. Specifically, for neural networks, optimization was performed for the number of layers and the number of neurons per layer. For the \( k \)NN algorithm, optimization focused on the number of neighbors, while for SVM, the kernel type was optimized. Across all algorithms, a 10-fold cross-validation was applied. To ensure statistical significance, the experiments were repeated 10 times. The performance was assessed using Mean Squared Error (MSE). The comparison results of the proposed IBA-VNS model with NN, SVM and \( k \)NN algorithm are presented in Table 2.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>MSE Train</th>
<th>Variance Train</th>
<th>MAE Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combined Cycle Power Plant</td>
<td>IBA-VNS</td>
<td>0.0041</td>
<td>0.0000</td>
<td>0.0048</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>0.0024</td>
<td>0.0218</td>
<td>0.0027</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.0037</td>
<td>0.0398</td>
<td>0.0037</td>
</tr>
<tr>
<td></td>
<td>( k )NN</td>
<td>0.0053</td>
<td>0.1370</td>
<td>0.0069</td>
</tr>
<tr>
<td></td>
<td>GAM</td>
<td>0.0045</td>
<td>0.0125</td>
<td>0.0056</td>
</tr>
<tr>
<td></td>
<td>DT</td>
<td>0.0032</td>
<td>0.0673</td>
<td>0.0033</td>
</tr>
<tr>
<td>Yacht Hydrodynamics</td>
<td>IBA-VNS</td>
<td>0.0258</td>
<td>0.0000</td>
<td>0.0298</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>0.0017</td>
<td>0.0514</td>
<td>0.0095</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.0299</td>
<td>0.1537</td>
<td>0.0297</td>
</tr>
<tr>
<td></td>
<td>( k )NN</td>
<td>0.0564</td>
<td>0.0936</td>
<td>0.0680</td>
</tr>
<tr>
<td></td>
<td>GAM</td>
<td>0.0359</td>
<td>0.0241</td>
<td>0.0378</td>
</tr>
<tr>
<td></td>
<td>DT</td>
<td>0.0472</td>
<td>0.0417</td>
<td>0.0521</td>
</tr>
<tr>
<td>Real estate valuation</td>
<td>IBA-VNS</td>
<td>0.0065</td>
<td>0.0000</td>
<td>0.0062</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>0.0045</td>
<td>0.0201</td>
<td>0.0067</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.0065</td>
<td>0.0817</td>
<td>0.0070</td>
</tr>
<tr>
<td></td>
<td>( k )NN</td>
<td>0.0097</td>
<td>0.0729</td>
<td>0.0112</td>
</tr>
<tr>
<td></td>
<td>GAM</td>
<td>0.0081</td>
<td>0.0392</td>
<td>0.0087</td>
</tr>
<tr>
<td></td>
<td>DT</td>
<td>0.0102</td>
<td>0.0381</td>
<td>0.0124</td>
</tr>
<tr>
<td>Insurance</td>
<td>IBA-VNS</td>
<td>0.0144</td>
<td>0.0000</td>
<td>0.0224</td>
</tr>
<tr>
<td></td>
<td>NN</td>
<td>0.0099</td>
<td>0.0247</td>
<td>0.0247</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.0187</td>
<td>0.1049</td>
<td>0.0192</td>
</tr>
<tr>
<td></td>
<td>( k )NN</td>
<td>0.0252</td>
<td>0.0839</td>
<td>0.0284</td>
</tr>
<tr>
<td></td>
<td>GAM</td>
<td>0.0139</td>
<td>0.0092</td>
<td>0.0217</td>
</tr>
<tr>
<td></td>
<td>DT</td>
<td>0.0193</td>
<td>0.0172</td>
<td>0.0197</td>
</tr>
</tbody>
</table>

The results indicate that the proposed model is comparable to NN and SVM in terms of accuracy and outperformed \( k \)NN algorithms. The IBA-VNS approach consistently demonstrated competitive results across all datasets, showcasing its ability and potential for a wide range of applications. Bearing in mind that \( k \)NN is the only fully transparent algorithm used in the comparison, these results are considered promising. Furthermore, IBA-VNS shows the best performance on Real estate valuation dataset, beating NN and SVM. The best results for each problem are given in bold.

The results show that one of the advantages of the IBA-VNS approach lies in its ability to maintain low variance during the training phase. This stability is important in real-world applications where model performance must be reliable across different scenarios, data variations and handle ambiguity in data. The robustness of the IBA-VNS approach positions it as a promising tool for tasks that require consistent and robust machine learning performance. In addition to transparency and interpretability, which is the main advantage
of the proposed approach, IBA-VNS gives good results for all benchmark datasets in terms of accuracy. Further, in the case of the Real estate valuation dataset, IBA-VNS is considered the best one. Given that Table 2 demonstrates the absence of a predominant method, transparency and interpretability set the IBA-VNS approach apart from others.

In summary, the proposed IBA-VNS approach stands out as a promising ML model, offering a balanced combination of accuracy, stability, and interpretability across diverse datasets. This advantage positions IBA-VNS as a valuable tool, particularly in domains where understanding and justifying predictions are essential.

5. IBA-VNS for Hospital Length of Stay Prediction

The length of stay (LOS) in hospital represents the interval between patient admission and his discharge from the hospital, serving as a crucial indicator of quality, efficiency, and hospital performance [58]. Predicting the average LOS has become increasingly significant in recent years, particularly for capacity management, resource planning, and effective admission scheduling. Anticipating LOS enables hospitals to predict discharge dates, it enables a better allocation of hospital beds and reduces the variability in their occupancy. This helps to minimizes patient waiting times and alleviates the workload of healthcare professionals. The primary objective of predicting LOS is to optimize the use of resources and improve patient satisfaction.

5.1. Problem Setup

Machine learning’s ability to handle complex relationships and changes in the data and provide mostly reliable predictions makes it especially suitable for predicting LOS in hospitals. The modeling of LOS in an emergency department presents unique challenges and considerations that distinguish it from the modeling approach employed in a scheduled department. Factors such as the unpredictable nature of patient arrivals, varied acuity levels, and the urgency of care required contribute to the distinctive dynamics of LOS models in emergency settings [59]. In contrast, scheduled departments often operate with a more structured patient flow; therefore, more research is devoted to predicting LOS for patients scheduled for surgery, as the structured and planned nature of scheduled procedures provides a more conducive environment for predictive modeling. Some authors opted to address this issue through classification, predicting whether the hospital stay is categorized as long-term or short-term [60,61]. Others, who employed a regression-based approach, emphasized the significance of attribute selection as crucial to enhancing model accuracy. The accuracy of the prediction depends on the selection of relevant data related to the patient. For instance, in Reference [62], the prediction of LOS in the cardiac surgical intensive care unit is based on factors such as the patient’s age, gender, hematocrit, and measurements of serum and creatinine. Most studies demonstrate a consistent trend of utilizing age, gender, and BMI as input attributes [63]. Conversely, certain authors chose to include personal information like marital status, occupation, and place of residence as additional input data in their study. However, it was discovered that these attributes did not significantly contribute to the output. For this reason, certain authors opt to incorporate techniques that identify and prioritize the most relevant and impactful variables to be used in the machine learning process [64]. The integration of expert knowledge into the attribute selection process has proven to be exceptionally successful [65]. This success can be attributed to the profound understanding that experts possess regarding the domain.

As far as we know, this will be the first attempt to predict LOS for patients undergoing abdominal surgery utilizing a logic-based machine learning algorithm.

5.2. Data and Specific Problem

In this subsection, a real-world problem in the medical domain is presented. The LapSerb program, known as the Serbian National Training Program for minimally invasive colorectal surgery, was introduced in Serbia to facilitate the adoption of laparoscopic colorectal surgery. This program aimed to expedite the training of experienced colorectal
surgeons through a competency-based approach. It involved assessing knowledge, conducting workshops, live surgeries, and evaluating competency through unedited video assessments. The purpose of this study is to present the outcomes of laparoscopic colorectal resections performed by surgeons certified by LapSerb.

One of the activities of the LapSerb program was collecting the data regarding laparoscopic colorectal resections performed in Serbia, from January 2015 to December 2023. The data collected encompassed patient demographics, surgical indications, perioperative information, and 30-day postoperative outcomes. These attributes are analyzed to understand how each of them or a combination of them can affect the length of postoperative hospital stay.

The dataset consists of 1811 instances and includes five attributes: age, gender, ASA physical status classification system category, BMI, and previous surgery in the abdomen. Gender and previous surgery are binary attributes, while the ASA category ranges from values 1 to 4. BMI is a continuous variable with a minimum of 14.000 and a maximum of 51.200, with an average of 25.370. The age attribute comprises a set of integers, with a minimum age of 18, a maximum age of 93, and an average age of 65.835.

The output variable in this research is the number of days spent in the hospital following laparoscopic colorectal resections. The range of this variable extends from a minimum of 2 days to a maximum of 146 days. In [66], LOS of 5 or more days usually refers to the patients who are hospitalized for a long period of time due to more complicated health problems. According to medical experts, these situations are the consequence of associated diseases or some unexpected events that occur during the surgery. In other situations, they may be the result of various social factors, e.g., they do not have proper care at home. In other words, the LOS for more than 10 days cannot be brought into direct connection with obtained inputs. Therefore, instances where the number of days exceeded 10 were considered as outliers and omitted from the further analysis. In the final version of the dataset, the mean number of days spent in the hospital is 6.366, with a median value of 6 days.

5.3. Results

In accordance with Section 3.2, the initial phase of the IBA-VNS approach involves data preparation. A correlation analysis was conducted, revealing no strong correlations within the attributes. The highest correlation identified was between ASA and age, with a coefficient of 0.330. Consequently, this correlation indicates that the product is a suitable choice for the GP operator in this scenario. Subsequently, min–max normalization was applied to non-binary input attributes within the interval [0.1, 0.9]. This range was chosen instead of [0, 1] to prevent instances where a value of 1 for the ASA attribute could potentially skew the function towards 0. Notably, the values for min and max in normalization were fixed to prevent data leakage, with values outside of predefined ranges being set to the corresponding interval limits.

In the second phase of applying the IBA-VNS algorithm, the output criterion was established at 500 iterations, where each iteration corresponded to one function evaluation. An early stopping criterion was not applied due to potential conflicts with the principle of progressive environment change in VNS optimization, especially when satisfactory results were not achieved. Throughout the training process, a 5-fold cross-validation approach was utilized to ensure robustness and accuracy. Consistent results were observed across four out of five folds, resulting in one LA function, while a different LA function was obtained for the remaining fold. Those LA functions are presented in Table 3.
After careful analysis, it was determined that gender had no discernible impact on the results, as consistent outcomes were observed regardless of gender. Consequently, gender was excluded from further consideration in the analysis.

The results from the test dataset demonstrate that the normalized MSE is calculated to be 0.034. A lower value indicates better performance, which is further supported by the normalized MAE value of 0.143. Furthermore, the MAE in days, which specifically measures the average absolute difference in days between the predicted and actual outcomes, is determined to be 1.144. This metric provides a tangible understanding of the model’s performance in predicting the LOS.

Based on these performance metrics, we can conclude that the IBA-VNS approach is well-suited to the observed problem of length of stay. It offers reasonably accurate predictions along with causal relations between the patient’s health condition and LOS in the hospital, which are understandable to both physicians and patients.

5.4. Explainability

The presented IBA-VNS model offers a significant advantage in terms of result comprehensibility. It provides clear and understandable ML model outcomes, allowing conclusions to be drawn even by individuals who are not experts in the observed field.

To better understand the results, we differentiate between patients who have and have not undergone previous surgeries. In the first case, for the patients who did not have previous surgeries, LA function can be written as follows:

\[
LA_1 = (AGE \land ASA \land BMI) \lor (AGE \land \neg ASA \land \neg BMI) \lor (\neg AGE \land (ASA \lor BMI)) \tag{3}
\]

For these patients, a clear pattern emerges: elderly individuals with high ASA and BMI values are considered to be at high risk for longer LOS. This risk remains consistent for elderly patients, regardless of lower ASA and BMI scores. Conversely, younger patients are categorized as potentially having a longer LOS if they exhibit markedly high values in either ASA or BMI.

On the other hand, for patients with a history of previous surgeries, the optimal LA function is the following:

\[
LA_2 = (AGE \land \neg BMI) \lor (BMI \land \neg AGE) \tag{4}
\]

The verbal interpretation of the model is as follows: the ASA score has no direct impact, only serving to amplify specific atoms within the LA function. Specifically, for obese patients, youth is no guarantee of safety, as high BMI serves as a risk factor replacement. Conversely, elderly patients remain at risk even in the absence of obesity, as age substitutes for a favorable BMI value.

All the conclusions derived from the LA functions are in accordance with medical practice and the experience of medical experts.
Next, we will illustrate the calculation of the LA functions and evaluate the model’s performances by presenting cases of four patients. Table 4 provides an overview of the input attribute values and corresponding output values.

Table 4. The illustrative examples: patients characteristics.

<table>
<thead>
<tr>
<th>Patient</th>
<th>AGE</th>
<th>ASA</th>
<th>BMI</th>
<th>S</th>
<th>LOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>64</td>
<td>3</td>
<td>20</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>87</td>
<td>1</td>
<td>25</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>45</td>
<td>1</td>
<td>24</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>44</td>
<td>2</td>
<td>20</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Since patients 1 and 2 did not have previous surgery, their LOS is estimated using LA model given in Equation (3), while LOS for patients 3 and 4 is calculated using LA in Equation (4). Further, in Table 5, we present the values of atomic elements included in the optimal LA function for patients 1 and 2, alongside their percentage share in the final values.

Table 5. The values of atomic elements included in optimal LA function for patients 1 and 2.

<table>
<thead>
<tr>
<th>Pat.</th>
<th>$AGE \land ASA \land BMI \land (\neg S)$</th>
<th>$\neg AGE \land ASA \land BMI \land (\neg S)$</th>
<th>$\neg AGE \land ASA \land BMI \land (\neg S)$</th>
<th>$\neg AGE \land (\neg ASA) \land BMI \land (\neg S)$</th>
<th>$AGE \land (\neg ASA) \land (\neg BMI) \land (\neg S)$</th>
<th>LA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.086 (15.75%)</td>
<td>0.059 (10.81%)</td>
<td>0.200 (36.63%)</td>
<td>0.034 (6.32%)</td>
<td>0.167 (30.59%)</td>
<td>0.546</td>
</tr>
<tr>
<td>2</td>
<td>0.014 (2.05%)</td>
<td>0.003 (0.44%)</td>
<td>0.014 (2.05%)</td>
<td>0.024 (3.51%)</td>
<td>0.629 (92.09%)</td>
<td>0.683</td>
</tr>
</tbody>
</table>

Upon analysis of the data presented in Table 5, we are able to determine the percentage importance of each individual atomic element to the total output value. This information may be interpreted as the relative importance of logic-based rules that leads to the final outcome, allowing us to determine the dominant pattern that describes the patients condition. In the case of patient 1, it is clear that atomic elements $(\neg AGE) \land ASA \land (\neg BMI) \land (\neg S)$ and $AGE \land (\neg ASA) \land (\neg BMI) \land (\neg S)$ have the highest influence on the final score. Therefore, it can be concluded that the profile of the patient 1 is dominantly described by these two logical functions. This may be interpreted as follows: a patient’s low BMI score leads to shorter LOS independently of the age and ASA score. In the case of the patient 2, it is clear that an atomic element $AGE \land (\neg ASA) \land (\neg BMI) \land (\neg S)$ is the dominant one. Therefore, LOS is primarily determined by the patient’s old age and irrelevant to other factors. After calculating the LA function value, denormalization is necessary to obtain estimated LOS. In our case, the denormalized values are 5.905 and 7.102, respectively. Comparing with the output values from Table 4 (6 and 8), we can deduce that the absolute prediction error for the first case is 0.095 and for the second case is 0.898.

In Table 6, we present the values of atomic elements included in the optimal LA function for patients who had previous surgeries, along with their percentage share in the final values.

Table 6. The values of atomic elements included in optimal LA function for patients 3 and 4.

<table>
<thead>
<tr>
<th>Pat.</th>
<th>$AGE \land ASA \land (\neg BMI) \land S$</th>
<th>$AGE \land (\neg ASA) \land (\neg BMI) \land S$</th>
<th>$\neg AGE \land (\neg ASA) \land BMI \land S$</th>
<th>$\neg AGE \land ASA \land BMI \land S$</th>
<th>LA</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.027 (5.90%)</td>
<td>0.239 (52.18%)</td>
<td>0.173 (37.77%)</td>
<td>0.019 (4.15%)</td>
<td>0.458</td>
</tr>
<tr>
<td>4</td>
<td>0.107 (24.71%)</td>
<td>0.184 (42.49%)</td>
<td>0.090 (20.79%)</td>
<td>0.052 (12.01%)</td>
<td>0.433</td>
</tr>
</tbody>
</table>
In the case of patient 3, predominant patterns are $\text{AGE} \land (\neg \text{ASA}) \land (\neg \text{BMI}) \land S$ and $(\neg \text{AGE}) \land (\neg \text{ASA}) \land \text{BMI} \land S$, while the other atomic elements have a small contribution to final LA value. The low value of ASA score directly affects the rather low value of LOS. Finally, in the case of patient 4, all atomic elements have substantial influence on the final LA value. Still, the atomic element $\text{AGE} \land (\neg \text{ASA}) \land (\neg \text{BMI}) \land S$ stands out as the most important one. This patient is generally in good condition, so their estimated LOS is the lowest. After evaluating the LA function values, we proceed to denormalization, resulting in values of 5.137 and 4.918, respectively. Comparing the estimated values with the corresponding outputs from Table 4 (5 and 4), we observe absolute prediction errors of 0.137 and 0.918 for the first and second cases, respectively.

6. Conclusions

In this study, we proposed a novel IBA-VNS approach, aimed to address the necessity for developing a logic-based ML algorithm that prioritizes interpretability and applicability. By integrating IBA with VNS, we sought to create a model that not only achieves desirable accuracy but also provides transparent and easily interpretable results. We opted for IBA over traditional fuzzy logic techniques due to its consistency in preserving all Boolean axioms, ensuring transparent logical reasoning. Complementarily, the simplicity and effectiveness of VNS as a non-parametric heuristic made it an ideal choice for optimizing the logical aggregation function within the IBA framework.

The approach is validated through the comparison with three well-known ML algorithms across four benchmark datasets. The results indicate that our model demonstrates comparable performance to NN and SVM while outperforming the $k$NN algorithm. However, the obtained IBA-VNS models are easy to understand and provide additional insight to the decision-maker.

Moreover, we implemented the proposed IBA-VNS approach for solving a real-world problem in the medical domain, i.e., to predict hospital length of stay after colorectal surgeries. In terms of model accuracy, our approach yielded an MAE of 1.144 days, which is considered to be significant. The achieved results may be used in resource optimization as well as to assess the surgery quality. However, the main contribution of the approach is extracting additional knowledge from the data. The obtained LA functions are easily interpretable as causal relations between patients’ health and their hospital stays. All conclusions derived from the LA functions are in accordance with medical practice and the experience of medical experts.

In summary, the IBA-VNS model represents a promising ML solution that strikes a balance between accuracy and interpretability. Its ability to provide transparent and comprehensible results positions it as a valuable tool, particularly in domains where understanding and justifying predictions are essential. Since the results are promising, the automation of the whole IBA-VNS reasoning will be the subject of further work.

Author Contributions: Conceptualization, N.Č. and P.M.; Methodology, N.Č.; Software, N.Č. and P.M.; Validation, I.D. and M.S.Č.; Investigation, N.Č.; Data curation, M.S.Č.; Writing—original draft, N.Č. and P.M.; Writing—review & editing, P.M., I.D. and M.S.Č.; Supervision, P.M. and I.D. All authors have read and agreed to the published version of the manuscript.

Funding: This study was partially supported by University of Belgrade—Faculty of Organizational Sciences.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References


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.